

Verknüpfungselemente-
Bibliothek
Advant Controller 31
ABB Procontic CS31

Programmiersystem

907 PC 331
Programmier- und Testsoftware

**ABB Schalt-
und Steuerungstechnik**



Bausteinübersicht, geordnet nach Aufrufnamen

Abhängig von den unterschiedlichen Familien der AC31-/CS31-Zentraleinheiten (Serie 30, 40, 50 und Serie 90) werden zwei Bibliotheken benutzt und im Editor "Projekt-Menü" aufgerufen. Wenn das VE im FUPKOP- oder im AWL-Editor aufgerufen wird, so wird automatisch angezeigt, ob das VE verfügbar ist oder nicht.

VEs für 07 KR 91, 07 KT 92 bis 07 KT 94 (Serie 90)

&	1- 1	AWTB	5-18	EMASm	10- 9	MUL2N	16- 6	VRZ	21-16
*	1- 3	AWTD	6- 1	EMASmVT	11- 1	MUL2ND	16-10	VRZD	21-19
*:	1- 5	BCDBIN	siehe	ESV	11- 3	MULD	siehe *D	VVZ	22- 1
*D	1- 7	BCDDUAL		FCDEL	11- 6	MULDI	siehe *:	WAES	22- 4
+	1- 9	BCDDUAL	6- 3	FCRD	11- 8	MUXR	16-14	WAND	22- 6
+D	1-11	BCDDUALD	6- 7	FCWR	11-11	MUXRD	16-17	WDEC	22- 8
-	1-13	BCDDW	siehe	FDEL	11-14	NEGD	17- 1	WDW	22-11
-D	1-15	BCDDUALD		FEHSU	11-16	NOTBIT	17- 3	Wenn dann	22-13
/	1-17	BEG	6-10	FIFO	11-19	PACK	17- 6	Wenn d.w.	22-14
:	1-19	BEGD	6-12	FKG	11-23	PACKD	17- 8	WOL	22-15
:D	2- 1	BETR	6-14	FRD	11-26	PDM	17-10	WOR	22-17
<	2- 4	BETRD	6-16	FWR	12- 1	PE	17-13	WOS	22-19
<=	2- 6	BINBCD	siehe	HLG	12- 4	PI	17-15	WRB	23- 1
<D	2- 8	DUALBCD		I-	12-10	PIDT1	18- 1	WRDW	23- 4
=	2-10	BITSU	6-18	I+	12-13	PT1	18- 8	WRW	23- 8
=1	2-12	BMELD	7- 1	IDENT	12-15	RDB	18-11	WUMC	23-11
=?	2-14	CALLUP	7- 4	IDL	siehe IDLm	RDDW	18-14	WWDW	23-14
=?D	2-16	COPY	7- 8	IDLB	12-17	RDW	18-17	WXOR	23-15
=D	2-18	COS1	7-10	IDLm	12-20	RS	19- 1	ZUDKW	23-17
=PE	3- 1	COUNTB	7-11	IDS	siehe IDSm	SETB	19- 3	ZUWD	siehe =D
=R	3- 3	COUNTW	7-13	IDSB	13- 1	SFEHSU	19- 5		
=S	3- 5	CS31CO	7-17	IDSm	13- 4	SHIFT	19- 9		
=W	3- 7	CS31QU	7-25	INITS	13- 7	SIN1	19-14		
>	3- 9	CS31TE	7-26	INITV	13-10	SINIT	19-15		
<>	3-11	DIN	8- 1	INTK	13-13	SPBM	20- 1		
><	3-13	DIVD	siehe :D	IOCON	13-17	SQRT	20- 4		
>=	3-15	DMUX	8- 2	IOR	14- 1	SR	20- 6		
>D	3-17	DMUXD	8- 5	IOW	14- 3	SUBD	siehe -D		
5F_ARC	3-19	DOUT	8- 9	LDT	14- 5	TESTB	20- 8		
5F_ARC94	3-19	DRUCK	8-10	LIFO	14- 8	UHR	20-16		
ABORT	4- 1	DT1	9- 1	LIZU	14-12	UNPACK	20-20		
ADAPT	4- 3	DUALBCD	9- 4	LZB	14-15	UNPACKD	20-22		
ADDD	siehe +D	DUALBCDD	9- 8	MAJ	14-17	UNPAD	siehe		
ADRWA	4-10	DWAES	9-11	MASKE	14-19	UNPACKD			
AINIT	4-15	DWAND	9-13	MASKED	15- 1	USM	20-24		
AMELD	4-17	DWBCD	siehe	MAX	15- 4	UST	20-26		
AMELDD	4-20	DUALBCDD		MAXD	15- 6	USTD	21- 1		
ANA14_20	5- 1	DWOL	9-15	MAZ	15- 8	USTR	21- 3		
APOLL	5- 3	DWOR	9-17	MAZD	15-10	USTRD	21- 5		
AREC	5- 4	DWOS	9-19	MIN	15-12	VGL3P	21- 7		
ASEND	5- 7	DWUMC	10- 1	MIND	15-14	VGLEH	21-10		
ASEND+ s. ASEND		DWW	10- 4	MOA	15-16	VGLUH	21-13		
ASV	5-11	DWWW	10- 6	MODBUS *)	15-21	VGLD	siehe =?D		
AWM	5-14	DWXOR	10- 7	MOK	16- 1	VGRD	siehe >D		
AWT	5-16	EMAS s. EMASm		MRK	16- 4	VKLD	siehe <D		

*) VE MODBUS nur für Serie 30, 40, 50

Die Aufstellung der gültigen VEs für Serie 30, 40, 50 befindet sich auf der folgenden Seite.

VEs für Serie 30, 40, 50

&	1- 1	AWTB	5-18	EMASm	10- 9	MUL2N	16- 6	VRZ	21-16
*	1- 3								
*:	1- 5	BCDBIN siehe BCDDUAL		ESV	11- 3	MULDI	siehe *:		
+	1- 9	BCDDUAL	6- 3					WAND	22- 6
-	1-13							WDW	22-11
/	1-17	BEG	6-10			NPULSE	17- 4	Wenn dann	22-13
:	1-19					PACK	17- 6	Wenn d.w.	22-14
		BETR	6-14					WOL	22-15
<	2- 4					PDM	17-10	WOR	22-17
<=	2- 6	BINBCD siehe DUALBCD		I-	12-10	PE	17-13		
=	2-10			I+	12-13	PI	17-15		
=1	2-12	BMELD	7- 1			PIDT1	18- 1		
=?	2-14	CALLUP	7- 4	IDL	siehe IDLm				
		COPY	7- 8	IDLm	12-20			WXOR	23-15
=PE	3- 1			IDS	siehe IDSm	RS	19- 1	ZUDKW	23-17
=R	3- 3								
=S	3- 5	CS31CO	7-17	IDSm	13- 4	SINIT	19-15		
=W	3- 7	CS31QU	7-25			SPBM	20- 1		
>	3- 9	CTU	7-30			SR	20- 6		
<>	3-11	CTUH	7-32			TOF	20-10		
>>	3-13	DI	7-35			TON	20-12		
>=	3-15					TP	20-14		
		DMUXD	8- 5						
		DO	8- 8						
		DRUCK	8-10			UHR	20-16		
		DUALBCD	9- 4	LIZU	14-12	UNPACK	20-20		
				MAX	15- 4				
				MIN	15-12				
				MOA	15-16				
ASV	5-11	DWW	10- 4	MOAT	15-19				
				MODBUS	15-21				
				MOK	16- 1				
AWT	5-16	EMAS s. EMASm		MRK	16- 4				

Hinweise

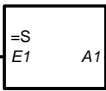
Im vorliegenden Bausteinkatalog erhalten Sie zunächst zwei Übersichten über sämtliche Bausteine

- eine geordnet nach Funktionsgruppen sowie
- eine weitere, alphabetisch geordnet nach den Aufrufnamen in FUPKOP und AWL.

Achtung:

Die Bausteine sind den *Baustein*namen nach alphabetisch sortiert. Sie sind *nicht* nach den Aufrufnamen in FUPKOP und AWL sortiert!

Die Beschreibung jedes Bausteins ist folgendermaßen aufgebaut:

Bausteinname	Kurzbeschreibung									
ZUWEISUNG SETZE SPEICHER	=S	Aufrufname im FUP								
<p>Mit diesem Verknüpfungselement wird eine Binär-Variablen gespeichert gesetzt.</p> <p>Zustand 1 am Eingang setzt den Operanden am Ausgang auf den Zustand 1. Zustand 0 am Eingang hat keinen Einfluß auf den Operanden am Ausgang.</p>	<p>FUP</p>  <p>AWL</p> <pre>! E1 =S A1</pre> <p>KOP</p> <pre>E1 A1] [(S) </pre>	<p>Darstellungen in FUP, KOP und AWL (Ein-/Ausgangsbezeichnungen, die auf dem Bildschirm nicht dargestellt werden, sind in der Beschreibung <i>kursiv</i> dargestellt).</p>								
Parameter		Parameter: Eingänge, Ausgänge / Datentyp / zulässige Operanden / Bedeutung des Parameters								
<table border="1"> <tr> <td>E1</td> <td>BINÄR</td> <td>E, M, A, S, K</td> <td>Setzbedingung</td> </tr> <tr> <td>A1</td> <td>BINÄR</td> <td>M, A</td> <td>Speicher-Variablen</td> </tr> </table>	E1	BINÄR	E, M, A, S, K	Setzbedingung	A1	BINÄR	M, A	Speicher-Variablen		
E1	BINÄR	E, M, A, S, K	Setzbedingung							
A1	BINÄR	M, A	Speicher-Variablen							
VE-Daten	Serie 90	Serie 30, 40, 50	Sonstige Bausteindaten							
<p>Laufzeit:</p> <p>Grundlaufzeit: 4,3 µs</p> <p>Zusätzliche Laufzeit: 2,3 µs pro zusätzlichen Ausgang (A3...An)</p> <p>Aktualisierung der Ausgänge: ja</p> <p>Anzahl der Vergangenheitswerte: keine</p> <p>Verfügbar ab: ABB Procontic CS31 / 907 PC 331</p>		<p>15 µs</p> <p>9 µs</p> <p>ja</p> <p>keine</p> <p>907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51</p>								
Beschreibung			Ausführliche Beschreibung							
<p>Zustand 1 am Eingang setzt den Operanden am Ausgang auf den Zustand 1. Zustand 0 am Eingang hat keinen Einfluß auf den Operanden am Ausgang.</p> <p>ACHTUNG:</p> <p>Dieses VE darf nur als Ausgangs-VE verwendet werden, d. h. es darf im FUP ausgangsseitig nicht durch eine Linie weiter verknüpft werden. Am Ausgang A1 <i>muß</i> ein Operand (M oder A) angegeben werden.</p> <p>Der Ausgang A1 ist doppelbar (A2 ... An). Der Eingang E1 ist invertierbar.</p>										
	3-5	907 PC 331/Stand: 10.95	Seiten-Nummer							

Bausteinname

ZUWEISUNG SETZE SPEICHER

=S

Beispiel

FUP AWL

```
!   E 00,00
=S  A 02,00
```

KOP

VE-FUP-Definition

Formale Definition des VEs

Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	L	J	P	N	0	0
A1	A	L	N	P	N	1	0

VE-AWL-Definition

00000	!	PP 0	E1	Eingang BINÄR
	[1		
00002	=S	PP 1	A1	Ausgang BINÄR (Dopplung möglich)
]	1		

907 PC 331/Stand: 10.95 3-6

Aufrufname im FUP

Beispiel

VE-Definitionen (sie dienen zur Festlegung des Graphiksymbols und als Übersetzungsvorschrift des Grafiksymbols in die AWL; s. a. 907 PC 33 Beschreibung, Allgemeiner Teil, Kap. 15 (Bibliothek/Erstellung der Grafik/Erstellen des Programmablaufes))

Weitere Erläuterungen

VE-Daten

Laufzeit:

Grundlaufzeit: Laufzeit ohne Dopplung
 Zusätzliche Laufzeit: Laufzeit mit Dopplung

Aktualisierung der Ausgänge:

Angabe, ob in jedem Zyklus die Ausgänge erneut zugewiesen werden. Bei "Nein" ist eine direkte Verbindung mit einer Linie zu weiteren VE's nicht möglich. In diesem Fall muß dann ein *nicht mehrfach* verwendeter Merker vergeben werden.

VE-FUP-Definitionen

(S. a. 907 PC 33 Beschreibung, Allgemeiner Teil, Kap. 15 (Bibliothek/Erstellung der Grafik/Erstellen des Programmablaufes))

Param.:

Angabe der Ein- und Ausgänge (symbolischer Name).

Param. Gruppe:

Angabe der Parameter-Gruppe (E=abfragbare Variable, A=zuweisbare Variable, S=Sonder-Variable, K=Konstante, X=Alle).

Param. Typ:

Angabe des Datentyps (L=Bit, B=Byte, W=Wort, D=Doppelwort, F=Floating, A=Analog, T=Textkonstante, Z=Zeitkonstante, S=Sonstige, X=Alle).

Inv.:
Angabe, ob Ein-/Ausgang invertierbar ist.

Belegung:
nur P zugelassen.

Darst. Schirm:
Angabe, ob Parameter-Name im VE auf dem Bildschirm dargestellt wird.

Param. Block:
Angabe, ob Eingang doppelbar ist (Eingang nicht doppelbar: 0; Eingang doppelbar: >0).

Doppl. Art:
Angabe der Art der Dopplung (Einzel=0, Blockweise=1).

Abkürzungen

Verwendete Abkürzungen finden Sie unter Griff 3, S. 0–1 des Ordners "907 PC 33, Allgemeiner Teil".

Hinweis zur komfortablen AWL

In der komfortablen AWL können alle SPS-Funktionen über ein Auswahl-Menü aufgerufen werden. Das Auswahl-Menü ist das gleiche wie beim FUPKOP. Wird eine Funktion aufgerufen, so erscheint auf dem Bildschirm eine Maske, in der alle Eingänge und Ausgänge gleich bezeichnet sind wie im FUPKOP. Der Anwender projiziert dann an den Eingängen und Ausgängen der Maske die von ihm gewünschten Operanden.

Bausteinübersicht, geordnet nach Funktionsgruppen

Funktionsgruppe	Aufrufname in	
Bausteinname	FUPKOP, / AWL 1) komf. AWL /	
Binäre Funktionen		Seite
UND	& 2)	1- 1
ODER	/ 2)	1-17
Exklusiv-ODER	=1 2)	2-12
Majorität	MAJ	14-17
Zuweisung	= 2)	2-10
Anzahl Bits in Wort/Doppelwort testen	COUNTB	7-11
Binärvariable in Wort/Doppelwort setzen	SETB	19- 3
Bit in Wort/Doppelwort testen	TESTB	20- 8
Speicherfunktionen		Seite
Zuweisung Setze Speicher	=S 2)	3- 5
Zuweisung Rücksetze Speicher	=R 2)	3- 3
Speicher, dominierend setzen	RS 2)	19- 1
Speicher, dominierend rücksetzen	SR 2)	20- 6
Arithmetische Funktionen, Wort		Seite
Addition	+ 2)	1- 9
Subtraktion	- 2)	1-13
Multiplikation	* 2)	1- 3
Division	: 2)	1-19
Multiplikation mit Division	*: / MULDI	1- 5
Multiplikation mit 2 hoch N	MUL2N	16- 6
Betragsbildner	BETR	6-14
Zuweisung, Wort	=W 2)	3- 7
Quadratwurzel	SQRT	20- 4
Zuweisung direkte Konstante an Wortvariable	ZUDKW	23-17
Arithmetische Funktionen, Doppelwort		Seite
Addition, Doppelwort	+D / ADDD	1-11
Subtraktion, Doppelwort	-D / SUBD	1-15
Multiplikation, Doppelwort	*D / MULD	1- 7
Division, Doppelwort	:D / DIVD	2- 1
Doppelwort-Multiplikation mit 2 hoch N	MUL2ND	16-10
Negation, Doppelwort	NEGD	17- 1
Betragsbildner, Doppelwort	BETRD	6-16
Zuweisung, Doppelwort	=D / ZUWD	2-18
Quadratwurzel	SQRT	20- 4
Kosinus	COS1	7-10
Sinus	SIN1	19-14

- 1) Existiert für die AWL ein anderer Aufruf als für FUPKOP und komf. AWL, so wird er durch / getrennt zusätzlich notiert.
- 2) Diese Funktion wird in der AWL durch eine Folge von Befehlen und/oder Bausteinen realisiert.

Funktionsgruppe	Aufrufname in	
Bausteinname	FUPKOP, / AWL 1) komf. AWL /	
Vergleichsfunktionen		Seite
Größer als	> 2)	3- 9
Größer gleich	>= 2)	3-15
Gleich	=? 2)	2-14
Ungleich	>< 2)	3-13
Ungleich	<> 2)	3-11
Kleiner gleich	<= 2)	2- 6
Kleiner als	< 2)	2- 4
Vergleicher mit 3-Punkt-Verhalten	VGL3P	21- 7
Vergleicher mit einseitiger Hysterese	VGLEH	21-10
Vergleicher mit unsymmetrischer Hysterese	VGLUH	21-13
Vergleichsfunktionen, Doppelwort		Seite
Größer als, Doppelwort	>D / VGRD	3-17
Kleiner als, Doppelwort	<D / VKLD	2- 8
Gleich, Doppelwort	=?D / VGLD	2-16
Zeitfunktionen		Seite
Einschaltverzögerung	ESV	11- 3
Einschaltverzögerung	TON	20-12
Ausschaltverzögerung	ASV	5-11
Ausschaltverzögerung	TOF	20-10
Variables Verzögerungsglied	VVZ	22- 1
Monostabiles Kippglied "Abbruch"	MOA	15-16
Monostabiles Kippglied "Abbruch"	MOAT	15-19
Monostabiles Kippglied "Konstant"	MOK	16- 1
Monostabiles Kippglied "Konstant"	TP	20-14
Zählfunktionen		Seite
Schneller Zähler	COUNTW	7-13
Vorwärts-/Rückwärtszähler	VRZ	21-16
Vorwärts-/Rückwärtszähler, Doppelwort	VRZD	21-19
Vorwärtszähler	CTU	7-30
Schneller Zähler	CTUH	7-32
Programmsteuerfunktionen		Seite
MODBUS-Master	MODBUS	15-21
Bedingter Sprung auf Marke	SPBM	20- 1
Zielmarke	MRK	16- 4
Programmende	PE	17-13
Bedingtes Programmende	=PE 2)	3- 1
Unterprogrammaufruf für ein Assemblerprogramm	CALLUP	7- 4
Laufzahlblock	LZB	14-15
Programm-Abbruch	ABORT	4- 1

- 1) Existiert für die AWL ein anderer Aufruf als für FUPKOP und komf. AWL, so wird er durch / getrennt zusätzlich notiert.
- 2) Diese Funktion wird in der AWL durch eine Folge von Befehlen und/oder Bausteinen realisiert.

Funktionsgruppe	Aufrufname in	
Bausteinname	FUPKOP, / AWL 1) komf. AWL /	
Programmsteuerfunktionen		Seite
Konfiguration von Ein- und Ausgängen	IOCON	13-17
Direkte Eingänge lesen	DIN	8- 1
Direkte Ausgänge schreiben	DOUT	8- 9
Direkte Eingänge lesen	DI	7-35
Direkte Ausgänge schreiben	DO	8- 8
Formatwandlung		Seite
BCD-nach-DUAL-Wandlung	BCDDUAL / BCDBIN	6- 3
DUAL-nach-BCD-Wandlung	DUALBCD / BINBCD	9- 4
Packen binärer Variablen in Wort	PACK	17- 6
Entpacken eines Wortes in binäre Variablen	UNPACK	20-20
Wort-nach-Doppelwort-Wandlung	WDW	22-11
Wandler 2 Worte nach Doppelwort	WWDW	23-14
Wandler Doppelwort nach 2 Worte	DWWW	10- 6
Doppelwort-nach-Wort-Wandlung	DWW	10- 4
BCD-nach-DUAL-Wandlung, Doppelwort	BCDDUALD / BCDDW	6- 7
DUAL-nach-BCD-Wandlung, Doppelwort	DUALBCDD / DWBCD	9- 8
Packen binärer Variablen in Doppelwort	PACKD	17- 8
Entpacken eines Doppelwortes in binäre Variablen	UNPACKD / UNPAD	20-22
Impuls		Seite
Impuls (positive Flanke)	I+ 2)	12-13
Impuls (negative Flanke)	I- 2)	12-10
Logische Funktionen mit Wortgrößen		Seite
UND-Verknüpfung, Wort	WAND	22- 6
ODER-Verknüpfung, Wort	WOR	22-17
Exklusiv-ODER-Verknüpfung, Wort	WXOR	23-15
Maske	MASKE	14-19
Shift-Baustein	SHIFT	19- 9
Logische Funktionen mit Doppelwortgrößen		Seite
UND-Verknüpfung, Doppelwort	DWAND	9-13
ODER-Verknüpfung, Doppelwort	DWOR	9-17
Exklusiv-ODER-Verknüpfung, Doppelwort	DWXOR	10- 7
Maske, Doppelwort	MASKED	15- 1
Shift-Baustein	SHIFT	19- 9
Zugriff auf EPROM / SmartMedia Card		Seite
Datensegment im Flash-EPROM löschen	FDEL	11-14
Datensatz vom Flash-EPROM lesen	FRD	11-26
Datensatz ins Flash-EPROM schreiben	FWR	12- 1
Datensegment in der SmartMedia Card löschen	FCDEL	11- 6
Datensatz von der SmartMedia Card lesen	FCRD	11- 8
Datensatz in die SmartMedia Card schreiben	FCWR	11-11

1) Existiert für die AWL ein anderer Aufruf als für FUPKOP und komf. AWL, so wird er durch / getrennt zusätzlich notiert.

2) Diese Funktion wird in der AWL durch eine Folge von Befehlen und/oder Bausteinen realisiert.

Funktionsgruppe	Aufrufname in	
Bausteinname	FUPKOP, / AWL 1) komf. AWL /	
Zugriff auf physikalische Adressen		Seite
Wort lesen mit Freigabe Wort schreiben mit Freigabe Wort schreiben bei Wertänderung Kopieren von Speicherbereichen	WOL WOS WAES COPY	22-15 22-19 22- 4 7- 8
Doppelwort-Zugriff auf physikalische Adressen		Seite
Doppelwort lesen mit Freigabe Doppelwort schreiben mit Freigabe Doppelwort schreiben bei Wertänderung	DWOL DWOS DWAES	9-15 9-19 9-11
Zugriff auf physikalische Adressen (I/O Ports)		Seite
Byte-Wert von I/O-Adresse lesen Byte-Wert nach I/O-Adresse schreiben	IOR IOW	14- 1 14- 3
ARCNET		Seite
Initialisierung ARCNET-Controller Datenpakete zum ARCNET-Controller übertragen ARCNET-Datenpakete empfangen ARCNET-Datenpakete senden	AINIT APOLL AREC ASEND	4-15 5- 3 5- 4 5- 7
Ankopplung Visualisierung über ARCNET (für 07 KT 92 und 07 KT 93) Ankopplung Visualisierung über ARCNET (für 07 KT 94)	5F_ARC 5F_ARC94	3-19 3-19
Funktionen höherer Ordnung		Seite
Wortdecodierer Wortumcodierer Doppelwortumcodierer Auswahltor, binär Auswahltor, Wort	WDEC WUMC DWUMC AWTB AWT	22- 8 23-11 10- 1 5-18 5-16
Auswahltor, Doppelwort Umschalttor Umschalttor mit Rücksetzung Umschalttor, Doppelwort Umschalttor mit Rücksetzung, Doppelwort	AWTD UST USTR USTD USTRD	6- 1 20-26 21- 3 21- 1 21- 5
Maximalwertbildner Maximalwertbildner, Doppelwort Minimalwertbildner Minimalwertbildner, Doppelwort Maximalwertbildner über der Zeit	MAX MAXD MIN MIND MAZ	15- 4 15- 6 15-12 15-14 15- 8
Maximalwert über der Zeit, Doppelwort Begrenzer Begrenzer, Doppelwort Hochlaufgeber Funktionsgeber	MAZD BEG BEGD HLG FKG	15-10 6-10 6-12 12- 4 11-23

- 1) Existiert für die AWL ein anderer Aufruf als für FUPKOP und komf. AWL, so wird er durch / getrennt zusätzlich notiert.
- 2) Diese Funktion wird in der AWL durch eine Folge von Befehlen und/oder Bausteinen realisiert.

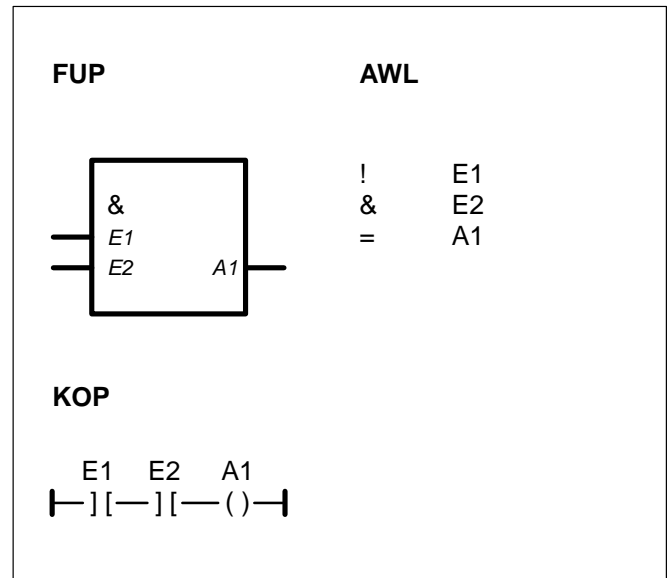
Funktionsgruppe	Aufrufname in	
Bausteinname	FUPKOP, / AWL 1) komf. AWL /	
Funktionen höherer Ordnung		Seite
Stapelspeicher last-in / first-out Stapelspeicher, first-in / first-out Fehlersucher mit automatischem Löschen Fehlersucher mit Speicherung Listenzuordner	LIFO FIFO FEHSU SFEHSU LIZU	14- 8 11-19 11-16 19- 5 14-12
Adressenauswahl Auswahl-Multiplexer Umschalt-Multiplexer Analogwert-Änderungsmelder Analogwert-Änderungsmelder, Doppelwort	ADRWA AWM USM AMELD AMELDD	4-10 5-14 20-24 4-17 4-20
Analogwert 4...20 mA einlesen (07 KT 92) Binärwert-Änderungsmelder Bitsucher Demultiplexer Demultiplexer, Doppelwort	ANAI4_20 BMELD BITSU DMUX DMUXD	5- 1 7- 1 6-18 8- 2 8- 5
Multiplexer mit Rücksetzen Multiplexer mit Rücksetzen, Doppelwort Binärvariable indiziert lesen Binärvariable indiziert schreiben Wortvariable indiziert lesen	MUXR MUXRD IDLB IDSB IDLm / IDL	16-14 16-17 12-17 13- 1 12-20
Wortvariable indiziert schreiben Leuchtdrucktasten-Ansteuerung Uhr Speicherbereich im Operandenspeicher mit Null initialisieren Variablen initialisieren	IDSm / IDS LDT UHR INITS INITV	13- 4 14- 5 20-16 13- 7 13-10

- 1) Existiert für die AWL ein anderer Aufruf als für FUPKOP und komf. AWL, so wird er durch / getrennt zusätzlich notiert.
- 2) Diese Funktion wird in der AWL durch eine Folge von Befehlen und/oder Bausteinen realisiert.

Funktionsgruppe	Aufrufname in	
Bausteinname	FUPKOP, / AWL 1) komf. AWL /	
Regelungstechnische Funktionen		Seite
Differenzierer mit Verzögerung 1. Ordnung Integrierer (komfortabel) Proportional-Integral-Regler PT1-Glied	DT1 INTK PI PT1	9- 1 13-13 17-15 18- 8
PIDT1-Regler Puls-Dauer-Modulator Adaption für adaptive Temperaturregelung Impulsgeber für Schrittmotoren	PIDT1 PDM ADAPT 3) NPULSE	18- 1 17-10 4- 3 17- 4
Kommunikation über serielle Schnittstelle		Seite
Initialisierung und Konfiguration der seriellen Schnittstellen Ausgabe von ASCII-Zeichen und Hex-Werten über eine serielle Schnittstelle Empfang von Zeichen Zusätzliche Vergleichstelegramme für EMASm	SINIT DRUCK EMASm / EMAS EMASmVT 4)	19-15 8-10 10- 9 11- 1
Vergangenheitswerte		Seite
Binärwerte aus Vergangenheitswertespeicher lesen Binärwerte in Vergangenheitswertespeicher schreiben Wortwerte aus Vergangenheitswertespeicher lesen Wortwerte in Vergangenheitswertespeicher schreiben Doppelwortwerte aus Vergangenheitswertespeicher lesen Doppelwortwerte in Vergangenheitswertespeicher schreiben	RDB WRB RDW WRW RDDW WRDW	18-11 23- 1 18-17 23- 8 18-14 23- 4
CS31-Funktionen		Seite
CS31-Module konfigurieren CS31-Fehler quittieren CS31-Vor-Ort-Module testen (nur Master) Identifikation	CS31CO CS31QU CS31TE IDENT	7-17 7-25 7-26 12-15
Sonderfunktionen		Seite
Wenn Dann Wenn Dann, Wort Not Bit	nicht möglich / 2) nicht möglich / 2) nicht möglich / 2)	22-13 22-14 17- 3

- 1) Existiert für die AWL ein anderer Aufruf als für FUPKOP und komf. AWL, so wird er durch / getrennt zusätzlich notiert.
- 2) Diese Funktion wird in der AWL durch eine Folge von Befehlen und/oder Bausteinen realisiert.
- 3) Aufruf nur in AWL möglich.
- 4) Kein Aufruf in AWL möglich.

Das Verknüpfungselement realisiert eine logische UND-Verknüpfung der Operanden an den Eingängen. Das Ergebnis wird dem Operanden am Ausgang zugewiesen.



Parameter

E1	BINÄR	E, M, A, S, K	Operand 1 der UND-Verknüpfung
E2	BINÄR	E, M, A, S, K	Operand 2 der UND-Verknüpfung, doppelbar
A1	BINÄR	M, A, S	Ergebnis der UND-Verknüpfung

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	6,6 µs	20 µs
Zusätzliche Laufzeit:	2,3 µs pro zusätzlichem Eingang	6 µs
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

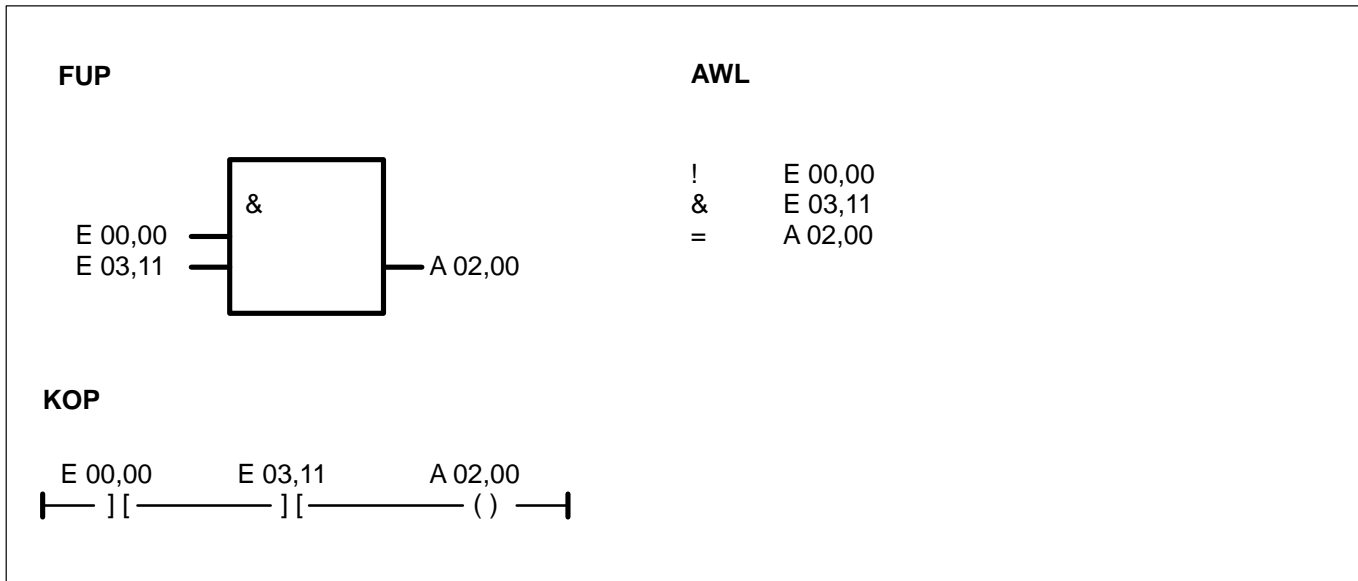
Das Verknüpfungselement realisiert eine logische UND-Verknüpfung der Operanden an den Eingängen. Das Ergebnis wird dem Operanden am Ausgang zugewiesen.

Der Eingang E2 ist doppelbar. Alle Eingänge und der Ausgang sind invertierbar.

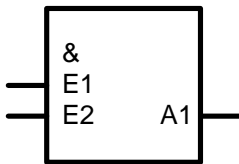
Wahrheitstabelle:

E1	E2	A1
0	0	0
1	0	0
0	1	0
1	1	1

Beispiel



VE-FUP-Definition

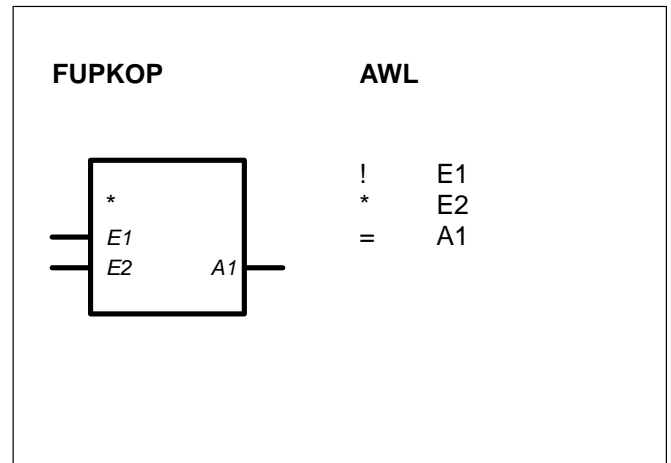


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	L	J	P	N	0	0
E2	E	L	J	P	N	1	0
A1	A	L	J	P	N	0	0

VE-AWL-Definition

00000	!	PP 0	E1	Eingang BINÄR
	[1		
00002	&	PP 1	E2	Eingang BINÄR (Dopplung möglich)
]	1		
00004	=	PP 0	A1	Ausgang BINÄR

Die Werte der Operanden an den Eingängen des Verknüpfungselementes werden miteinander multipliziert und das Ergebnis dem Operanden am Ausgang zugewiesen.



Parameter

E1	WORT	EW, MW, AW, KW	Multiplikand
E2	WORT	EW, MW, AW, KW	Multiplikator; Eingang ist doppelbar
A1	WORT	MW, AW	Ergebnis (Produkt)

VE-Daten

Laufzeit:		
Grundlaufzeit:		< 30 µs
Zusätzliche Laufzeit:		23 µs pro zusätzl. Eingang (E3 ... En)
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Die Werte der Operanden an den Eingängen des Verknüpfungselementes werden miteinander multipliziert und das Ergebnis dem Operanden am Ausgang zugewiesen.

Der Eingang E2 ist doppelbar (E2 ... En). Alle Eingänge und der Ausgang sind negierbar.

Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für die *nicht negierten* Eingänge E1 und E2:

untere Grenze: 8000_H (-32768)

Allgemein gilt:

- untere Grenze: 8001_H -32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (-32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

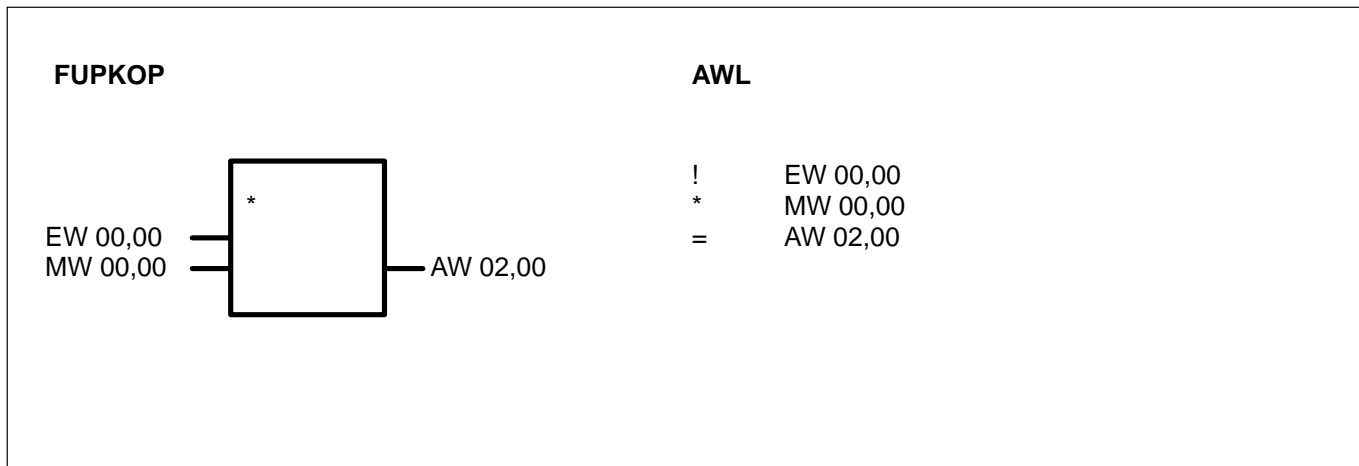
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

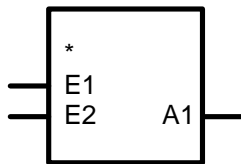
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (-32768) auf den erlaubten Wert 8001_H (-32767) korrigiert.

Beispiel



VE-FUP-Definition

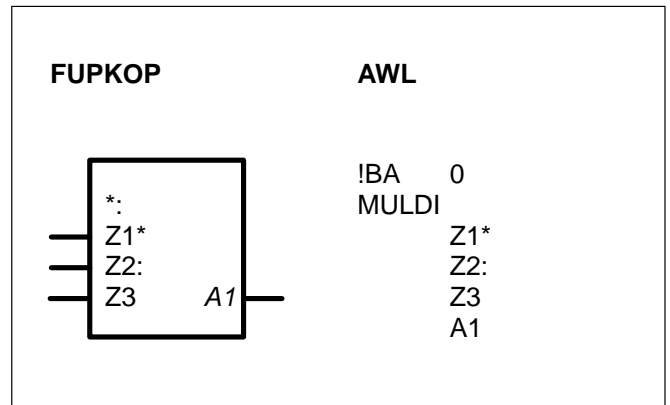


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	J	P	N	0	0
E2	E	W	J	P	N	1	0
A1	A	W	J	P	N	0	0

VE-AWL-Definition

00000	!	PP 0	E1	Eingang WORT
	[1		
00002	*	PP 1	E2	Eingang WORT (Dopplung möglich)
]	1		
00005	=	PP 0	A1	Ausgang WORT

Der Wert des Operanden am Eingang Z1 wird mit dem Wert des Operanden am Eingang Z2 multipliziert, das Zwischenergebnis wird durch den Wert des Operanden am Eingang Z3 dividiert und dann das Ergebnis dem Operanden am Ausgang A1 zugewiesen.



Parameter

Z1*	WORT	EW, MW, AW, KW	Multiplikand
Z2:	WORT	EW, MW, AW, KW	Multiplikator
Z3	WORT	EW, MW, AW, KW	Divisor
A1	WORT	MW,AW	Ergebnis

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	84 µs	1430 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Wert des Operanden am Eingang Z1 wird mit dem Wert des Operanden am Eingang Z2 multipliziert, das Zwischenergebnis wird durch den Wert des Operanden am Eingang Z3 dividiert und dann das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Intern arbeitet der Funktionsbaustein bei der Multiplikation und Division mit Doppelwortgenauigkeit (32 Bit). Erst bei der Zuweisung des Ergebnisses an den Ausgang A1 erfolgt die Begrenzung auf Wortgenauigkeit (16 Bit). Ist der Rest der Division $\geq 0,5$ wird das Ergebnis aufgerundet. Entsteht bei der Division ein Zahlenüberlauf (z. B. Division durch Null), so wird dem Ausgang A1 der vorzeichenrichtige Grenzwert des Zahlenbereiches zugewiesen.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar.

Zahlenbereich

Integer Wort (16 Bit)

- untere Grenze: 8001_H -32767
- obere Grenze: $7FFF_H$ $+32767$
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (-32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

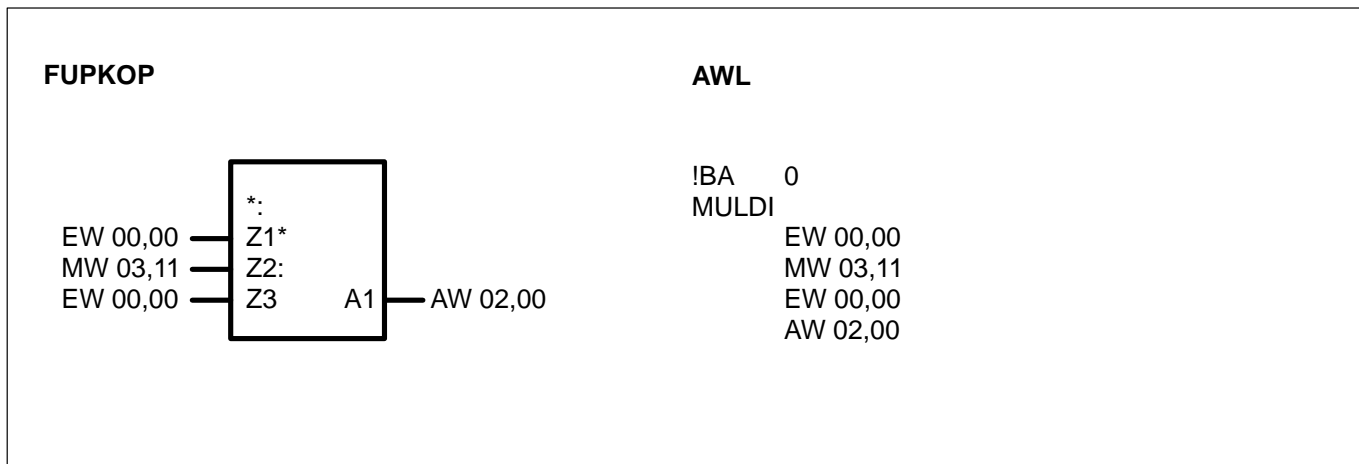
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

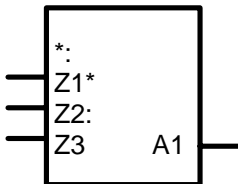
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (-32768) auf den erlaubten Wert 8001_H (-32767) korrigiert.

Beispiel



VE-FUP-Definition



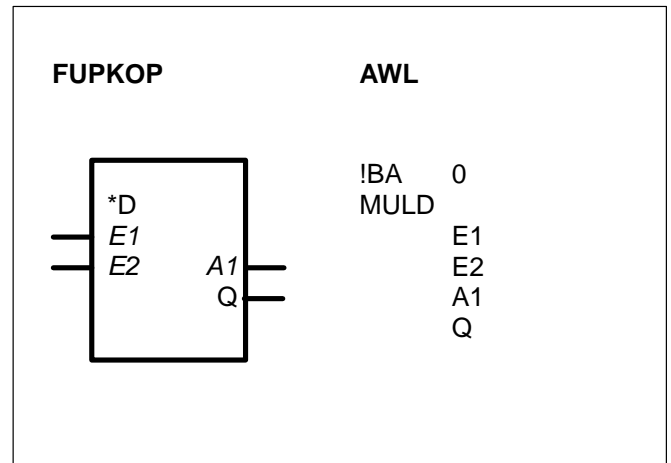
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
Z1*	E	W	N	P	J	0	0
Z2:	E	W	N	P	J	0	0
Z3	E	W	N	P	J	0	0
A1	A	W	N	P	N	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MULDI			
00002		PP 0	Z1*	Eingang WORT
00003		PP 0	Z2:	Eingang WORT
00004		PP 0	Z3	Eingang WORT
00005		PP 0	A1	Ausgang WORT

Der Wert des Operanden am Eingang E1 wird mit dem Wert des Operanden am Eingang E2 multipliziert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Das Ergebnis wird auf den maximalen bzw. minimalen Wert des Zahlenbereichs begrenzt. Fand eine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 1-Signal zugewiesen. Fand keine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 0-Signal zugewiesen.



Parameter

E1	DOPPELWORT MD, KD	Multiplikand
E2	DOPPELWORT MD, KD	Multiplikator
A1	DOPPELWORT MD	Ergebnis (Produkt)
Q	BINÄR A,M	Ergebnis begrenzt

VE-Daten

Laufzeit:	
Grundlaufzeit:	117 ... 120 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang E1 wird mit dem Wert des Operanden am Eingang E2 multipliziert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Das Ergebnis wird auf den maximalen bzw. minimalen Wert des Zahlenbereichs begrenzt. Fand eine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 1-Signal zugewiesen. Fand keine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 0-Signal zugewiesen.

Die Ein- und Ausgänge sind weder doppelbar noch negierbar.

Zahlenbereich

Integer Doppelwort (32 Bit)

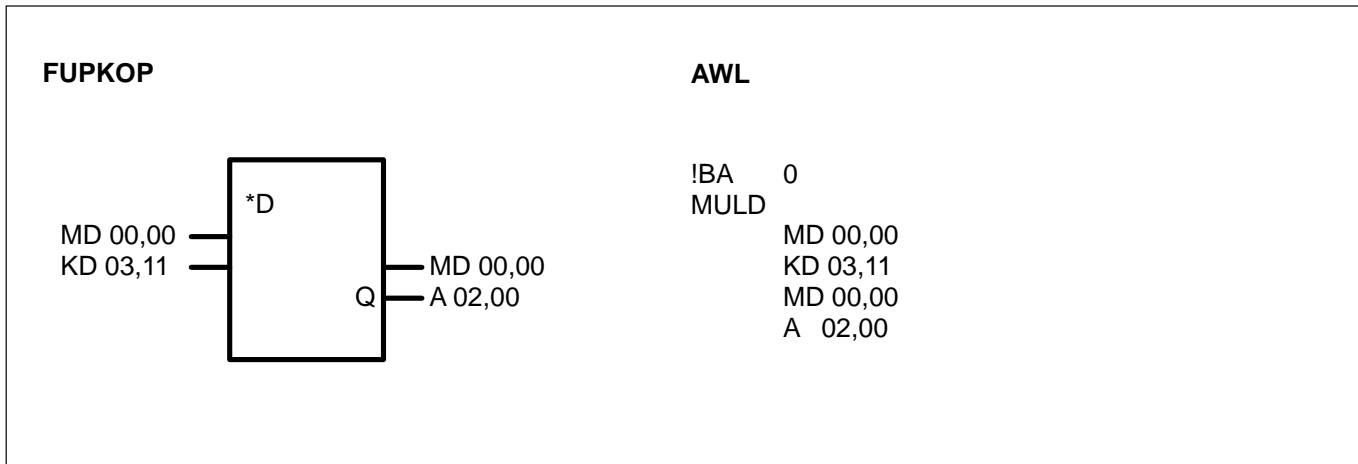
Speziell gilt hier für die Eingänge E1 und E2:

- untere Grenze: 8000 0000_H -2 147 483 648

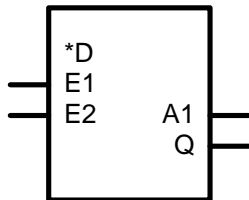
Allgemein gilt:

- untere Grenze: 8000 0001_H -2 147 483 647
- obere Grenze: 7FFF FFFF_H +2 147 483 647
- unzulässiger Wert: 8000 0000_H —

Beispiel



VE-FUP-Definition

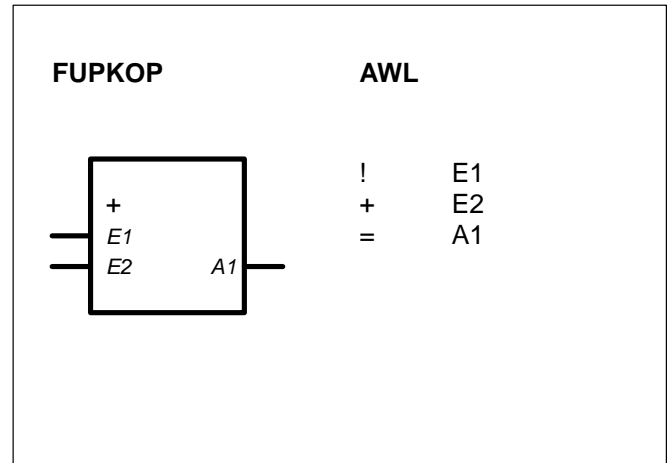


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
E2	E	D	N	P	N	0	0
A1	A	D	N	P	N	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MULD			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	E2	Eingang DOPPELWORT
00004		PP 0	A1	Ausgang DOPPELWORT
00005		PP 0	Q	Ausgang BINÄR

Die Werte der Operanden an den Eingängen des Verknüpfungselementes werden addiert und das Ergebnis dem Operanden am Ausgang zugewiesen.



Parameter

E1	WORT	EW, MW, AW, KW	Summand 1
E2	WORT	EW, MW, AW, KW	Summand 2; Eingang ist doppelbar
A1	WORT	MW, AW	Summe

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

Zusätzliche Laufzeit:

Aktualisierung der Ausgänge:

Anzahl der Vergangenheitswerte:

Verfügbar ab:

< 12 µs

5 µs pro zusätzl. Eingang (E3...En)

ja

keine

ABB Procontic CS31 / 907 PC 331

< 46 µs

5 µs pro zusätzl. Eingang (E3...En)

ja

keine

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Die Werte der Operanden an den Eingängen des Verknüpfungselementes werden addiert und das Ergebnis dem Operanden am Ausgang zugewiesen.

Der Eingang E2 ist doppelbar (E2 ... En). Der Eingang E1 und der Ausgang A1 sind negierbar.

Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für den *nicht negierten* Eingang E1:

Liegt an E1 der unzulässige Wert 8000_H (−32768), so wird er automatisch vor seiner Verarbeitung auf den zulässigen Wert 8001_H (−32767) korrigiert.

Speziell gilt hier für den *nicht negierten* Eingang E2:
untere Grenze: 8000_H (−32768)

Allgemein gilt:

- untere Grenze: 8001_H −32767
- obere Grenze: $7FFF_H$ +32767
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (−32768) liegt bei der Zweierkomplement–Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

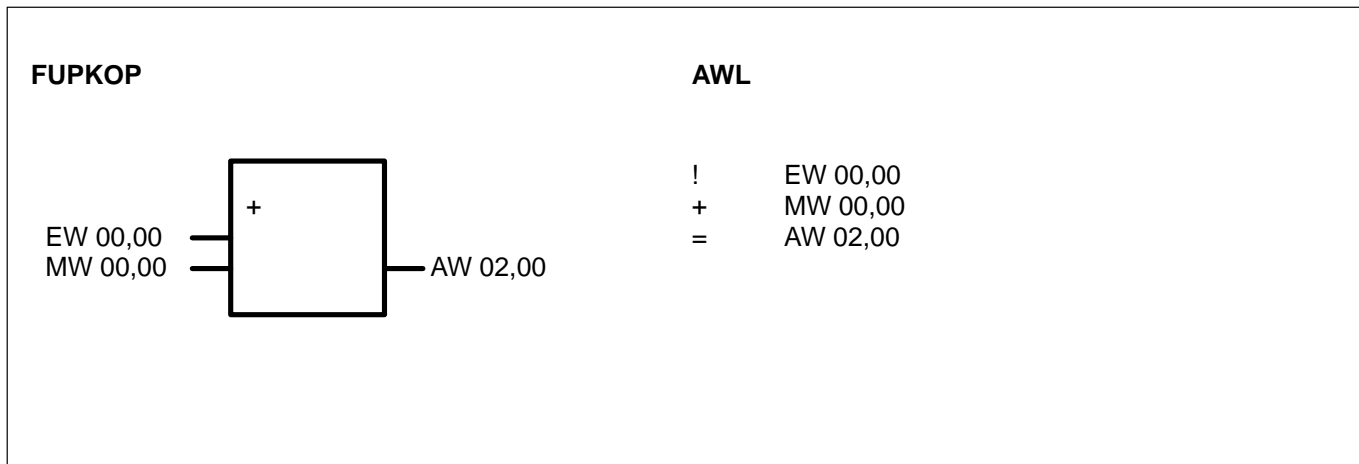
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

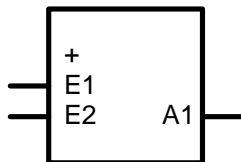
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (−32768) auf den erlaubten Wert 8001_H (−32767) korrigiert.

Beispiel



VE-FUP-Definition



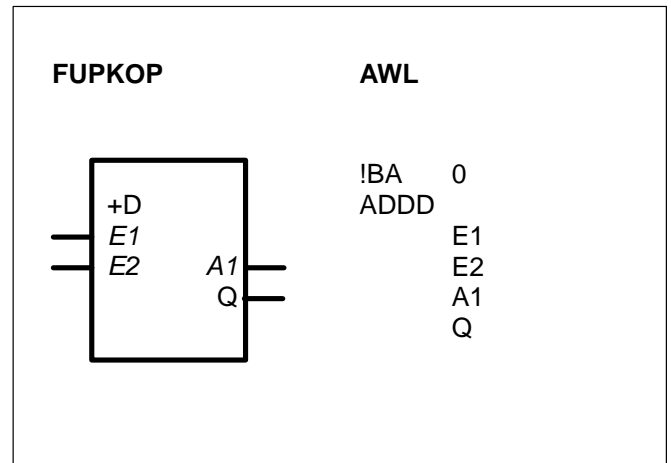
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	J	P	N	0	0
E2	E	W	N	P	N	1	0
A1	A	W	J	P	N	0	0

VE-AWL-Definition

00000	!	PP 0	E1	Eingang WORT
	[1		
00002	+	PP 1	E2	Eingang WORT (Dopplung möglich)
]	1		
00005	=	PP 0	A1	Ausgang WORT

Der Wert des Operanden am Eingang E1 wird zum Wert des Operanden am Eingang E2 addiert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Das Ergebnis wird auf den maximalen bzw. minimalen Wert des Zahlenbereichs begrenzt. Fand eine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 1-Signal zugewiesen. Fand keine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 0-Signal zugewiesen.



Parameter

E1	DOPPELWORT MD, KD	Summand 1
E2	DOPPELWORT MD, KD	Summand 2
A1	DOPPELWORT MD	Summe
Q	BINÄR A, M	Summe begrenzt

VE-Daten

Laufzeit:	
Grundlaufzeit:	49 ... 52 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang E1 wird zum Wert des Operanden am Eingang E2 addiert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Das Ergebnis wird auf den maximalen bzw. minimalen Wert des Zahlenbereichs begrenzt. Fand eine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 1-Signal zugewiesen. Fand keine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 0-Signal zugewiesen.

Die Ein- und Ausgänge sind weder doppelbar noch negierbar.

Zahlenbereich

Integer Doppelwort (32 Bit).

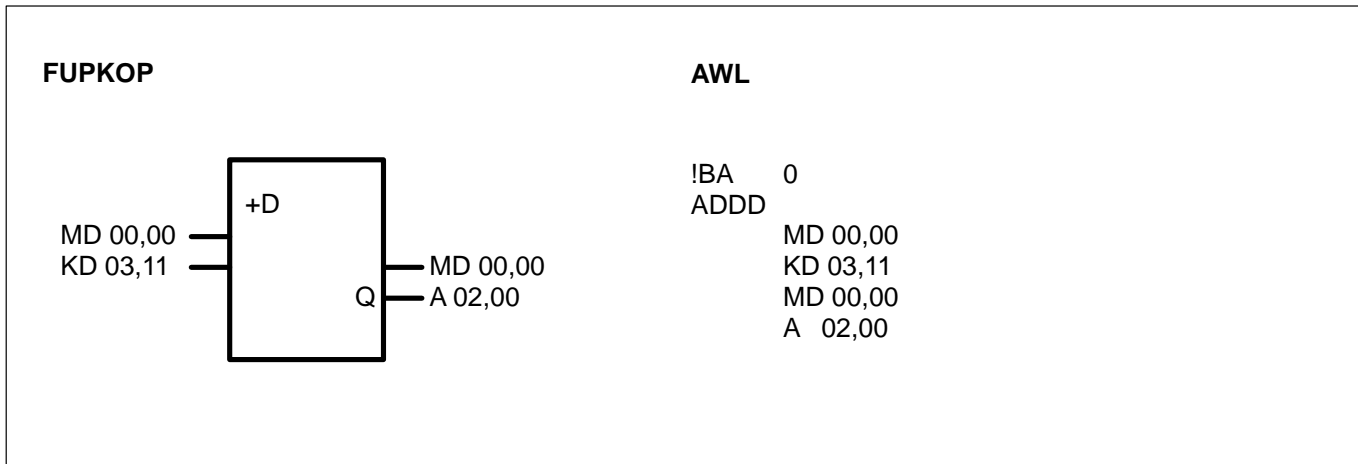
Speziell gilt hier für die Eingänge E1 und E2:

- untere Grenze: 8000 0000_H -2 147 483 648

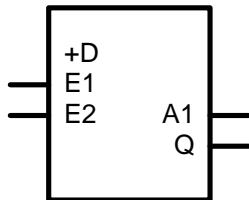
Allgemein gilt:

- untere Grenze: 8000 0001_H -2 147 483 647
- obere Grenze: 7FFF FFFF_H +2 147 483 647
- unzulässiger Wert: 8000 0000_H —

Beispiel



VE-FUP-Definition

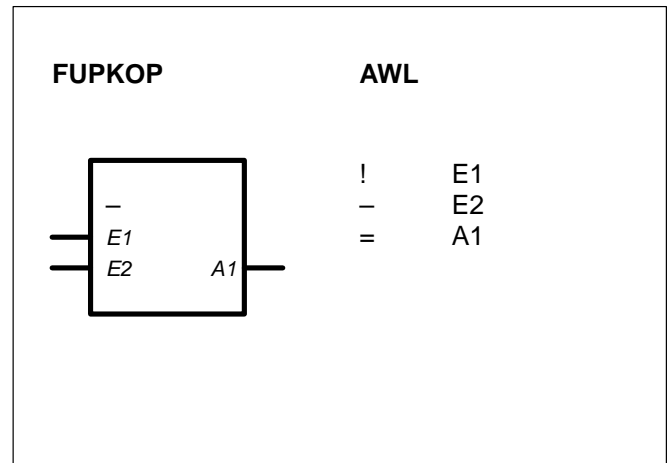


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
E2	E	D	N	P	N	0	0
A1	A	D	N	P	N	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	ADDD			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	E2	Eingang DOPPELWORT
00004		PP 0	A1	Ausgang DOPPELWORT
00005		PP 0	Q	Ausgang BINÄR

Der Wert des Operanden am Eingang E2 wird vom Wert des Operanden am Eingang E1 subtrahiert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	WORT	EW, MW, AW, KW	Minuend
E2	WORT	EW, MW, AW, KW	Subtrahend; Eingang ist doppelbar
A1	WORT	MW, AW	Ergebnis (Differenz)

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

< 12 µs

< 49 µs

Zusätzliche Laufzeit:

5 µs pro zusätzl. Eingang (E3 ... En)

21 µs pro zusätzl. Eingang (E3 ... En)

Aktualisierung der Ausgänge:

ja

ja

Anzahl der Vergangheitswerte:

keine

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Wert des Operanden am Eingang E2 wird vom Wert des Operanden am Eingang E1 subtrahiert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Der Eingang E2 ist doppelbar (E2 ... En). Wird er gedoppelt, so werden vom Operand am Eingang E1 alle Werte der Operanden an den Eingängen E2 ... En subtrahiert.

Der Eingang E1 und der Ausgang A1 sind negierbar.

Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für den *nicht negierten* Eingang E1: Liegt an E1 der unzulässige Wert 8000_H (-32768), so wird er automatisch vor seiner Verarbeitung auf den zulässigen Wert 8001_H (-32767) korrigiert.

Allgemein gilt:

- untere Grenze: 8001_H -32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (-32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

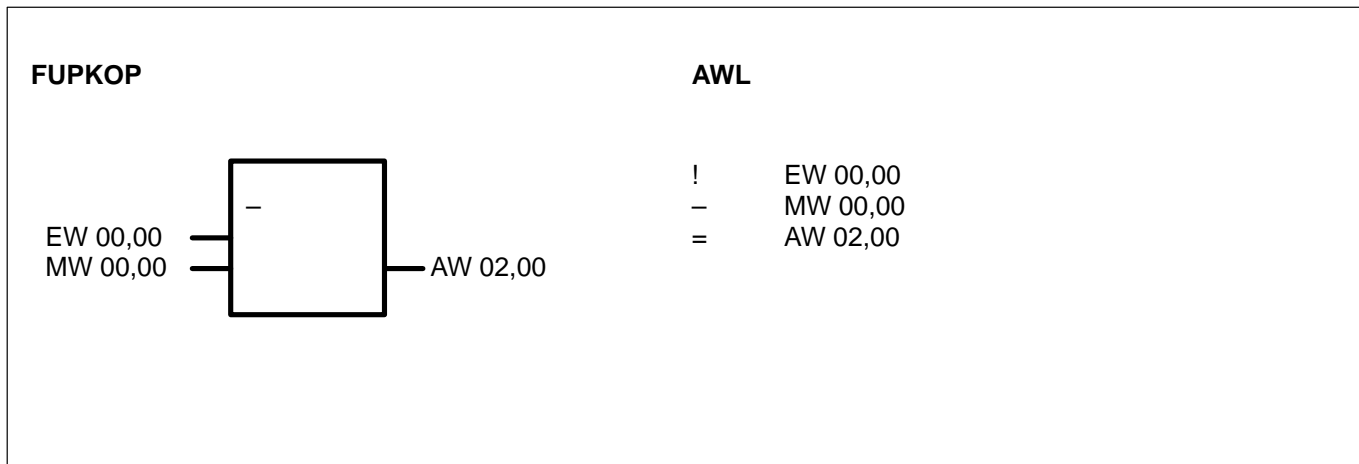
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

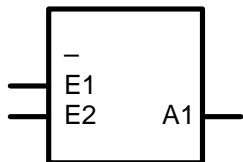
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (-32768) auf den erlaubten Wert 8001_H (-32767) korrigiert.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	J	P	N	0	0
E2	E	W	N	P	N	1	0
A1	A	W	J	P	N	0	0

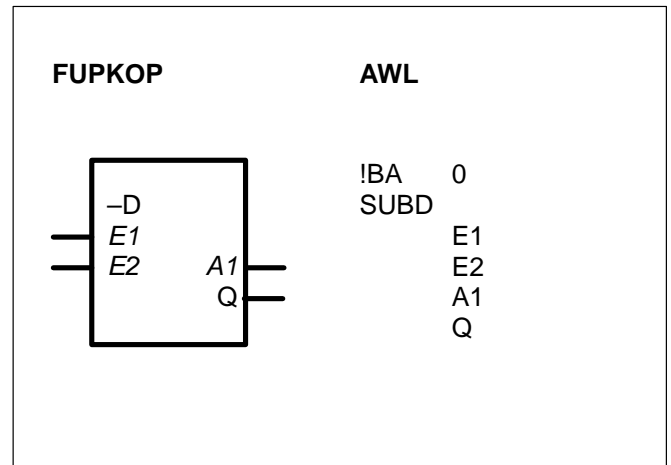
VE-AWL-Definition

00000	!	PP 0	E1	Eingang WORT
	[1		
00002	-	PP 1	E2	Eingang WORT (Dopplung möglich)
]	1		
00005	=	PP 0	A1	Ausgang WORT

Der Wert des Operanden am Eingang E2 wird vom Wert des Operanden am Eingang E1 subtrahiert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Das Ergebnis wird auf den maximalen bzw. minimalen Wert des Zahlenbereichs begrenzt. Fand eine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 1-Signal zugewiesen. Fand keine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 0-Signal zugewiesen.

Der Wert des Operanden am Eingang E2 wird vor Ausführung der Subtraktion überprüft, ob er außerhalb des zulässigen Zahlenbereiches liegt (8000 0000_H). Ist dies der Fall, so wird anstelle dieses unzulässigen Wertes mit dem Wert -2 147 483 647 (8000 0001_H) gerechnet.



Parameter

E1	DOPPELWORT MD, KD	Minuend
E2	DOPPELWORT MD, KD	Subtrahend
A1	DOPPELWORT MD	Ergebnis (Differenz)
Q	BINÄR A, M	Ergebnis begrenzt

VE-Daten

Laufzeit:	
Grundlaufzeit:	50 ... 55 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang E2 wird vom Wert des Operanden am Eingang E1 subtrahiert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Das Ergebnis wird auf den maximalen bzw. minimalen Wert des Zahlenbereichs begrenzt. Fand eine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 1-Signal zugewiesen. Fand keine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 0-Signal zugewiesen.

Der Wert des Operanden am Eingang E2 wird vor Ausführung der Subtraktion überprüft, ob er außerhalb des zulässigen Zahlenbereiches liegt (8000 0000_H). Ist dies der Fall, so wird anstelle dieses unzulässigen Wertes mit dem Wert -2 147 483 647 (8000 0001_H) gerechnet.

Die Ein- und Ausgänge sind weder doppelbar noch negierbar.

Zahlenbereich

Integer Doppelwort (32 Bit).

Speziell gilt hier für den Eingang E1:

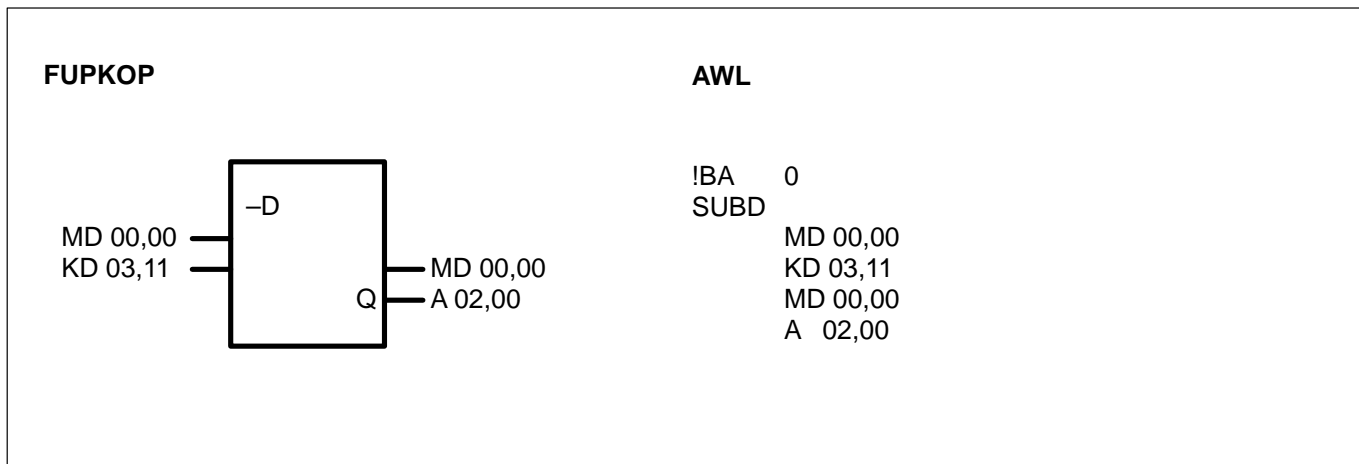
- untere Grenze: 8000 0000_H -2 147 483 647.

Allgemein gilt:

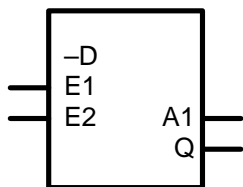
Integer Doppelwort (32 Bit)

- untere Grenze: 8000 0001_H -2 147 483 647
- obere Grenze: 7FFF FFFF_H +2 147 483 647
- unzulässiger Wert: 8000 0000_H —

Beispiel



VE-FUP-Definition

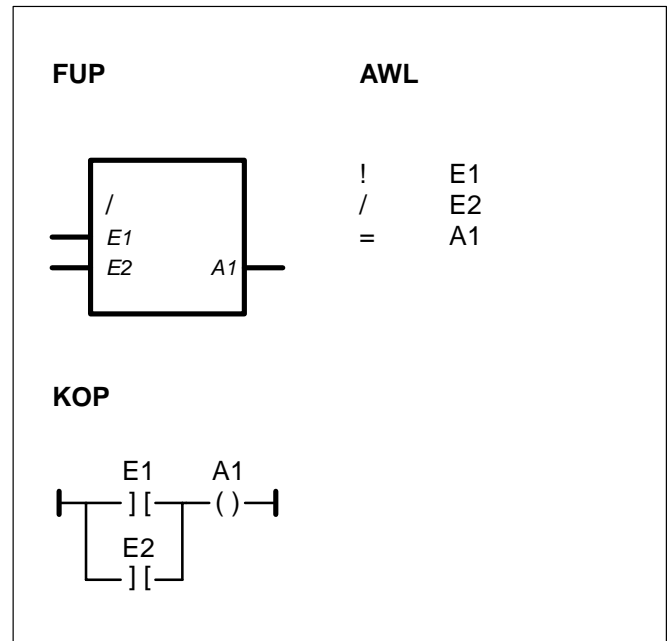


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
E2	E	D	N	P	N	0	0
A1	A	D	N	P	N	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	SUBD			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	E2	Eingang DOPPELWORT
00004		PP 0	A1	Ausgang DOPPELWORT
00005		PP 0	Q	Ausgang BINÄR

Das Verknüpfungselement realisiert eine logische ODER-Verknüpfung der Operanden an den Eingängen. Das Ergebnis wird dem Operanden am Ausgang zugewiesen.



Parameter

E1	BINÄR	E, M, A, S, K	Operand 1 der ODER-Verknüpfung
E2	BINÄR	E, M, A, S, K	Operand 2 der ODER-Verknüpfung, doppelbar
A1	BINÄR	M, A, S	Ergebnis der ODER-Verknüpfung

VE-Daten

Laufzeit:	
Grundlaufzeit:	6,6 µs
Zusätzliche Laufzeit:	2,3 µs pro zusätzl. Eingang
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

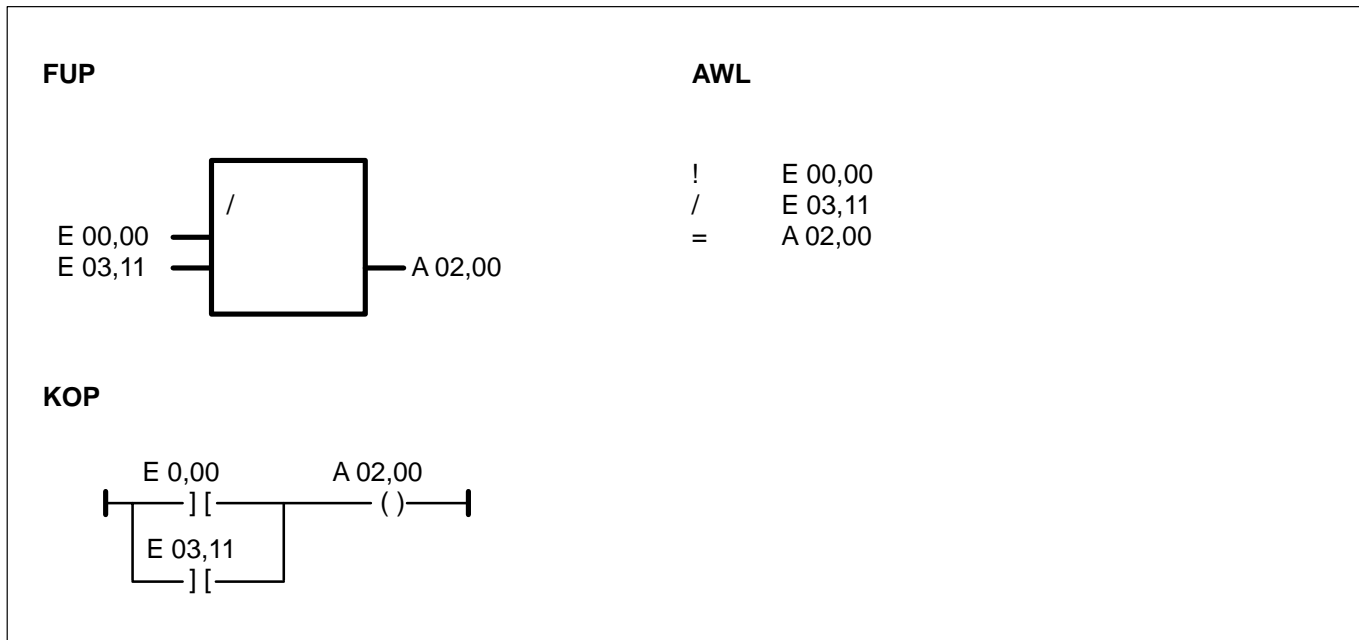
Das Verknüpfungselement realisiert eine logische ODER-Verknüpfung der Operanden an den Eingängen. Das Ergebnis wird dem Operanden am Ausgang zugewiesen.

Der Eingang E2 ist doppelbar. Alle Eingänge und der Ausgang sind invertierbar.

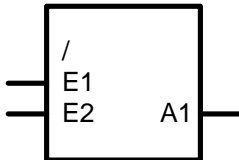
Wahrheitstabelle:

E1	E2	A1
0	0	0
1	0	1
0	1	1
1	1	1

Beispiel



VE-FUP-Definition



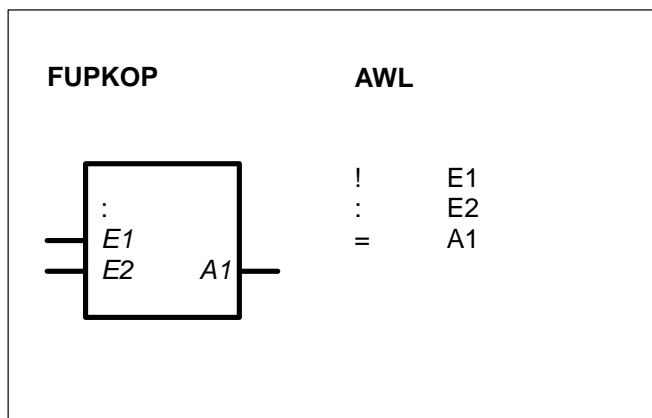
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	L	J	P	N	0	0
E2	E	L	J	P	N	1	0
A1	A	L	J	P	N	0	0

VE-AWL-Definition

00000	!	PP 0	E1	Eingang BINÄR
	[1		
00002	/	PP 1	E2	Eingang BINÄR (Dopplung möglich)
]	1		
00004	=	PP 0	A1	Ausgang BINÄR

Der Wert des Operanden am Eingang E1 wird durch den Wert des Operanden am Eingang E2 dividiert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Das Ergebnis der Division ist stets eine ganze Zahl. Eine Rundung (Berücksichtigung der Ziffern nach dem Komma) findet nicht statt.



Parameter

E1	WORT	EW, MW, AW, KW	Dividend
E2	WORT	EW, MW, AW, KW	Divisor
A1	WORT	MW, AW	Ergebnis (Quotient)

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	< 31 µs	< 781 µs
Zusätzliche Laufzeit:	24 µs pro zusätzl. Eingang (E3 ... En)	753 µs
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Wert des Operanden am Eingang E1 wird durch den Wert des Operanden am Eingang E2 dividiert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Das Ergebnis der Division ist stets eine ganze Zahl. Eine Rundung (Berücksichtigung der Ziffern nach dem Komma) findet nicht statt.

Der Eingang E2 ist doppelbar (E2 ... En). Wird er gedoppelt, so gilt:

$$E1 : E2 : E3 \dots : En = A1.$$

Alle Eingänge und der Ausgang sind negierbar.

Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für die *nicht negierten* Eingänge E1 und E2:

untere Grenze: 8000_H (-32768)

Allgemein gilt:

- untere Grenze: 8001_H -32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (-32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

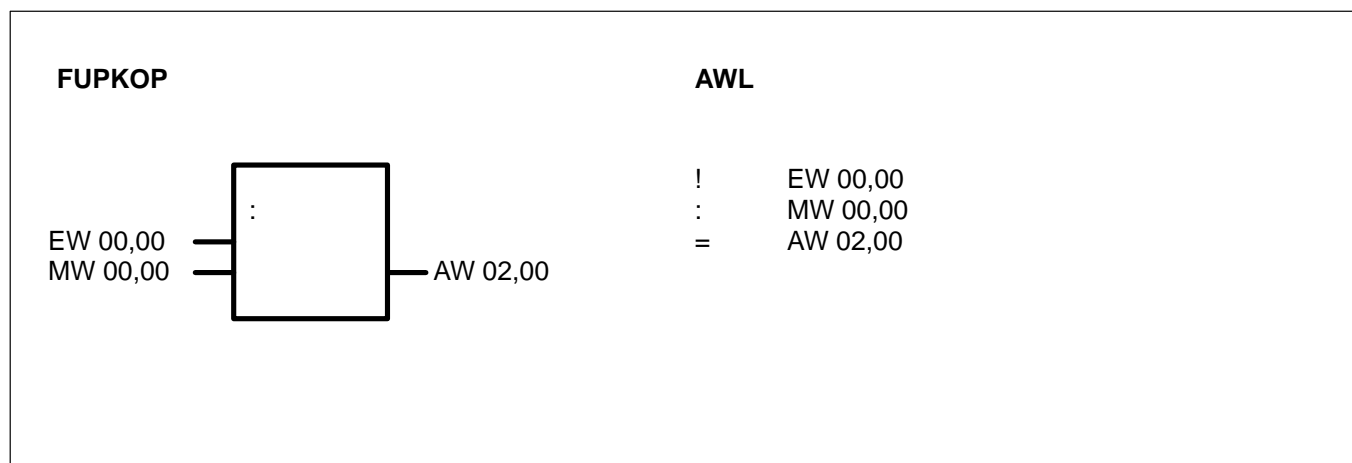
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

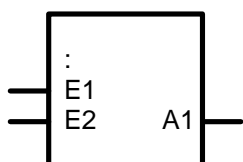
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (-32768) auf den erlaubten Wert 8001_H (-32767) korrigiert.

Beispiel



VE-FUP-Definition

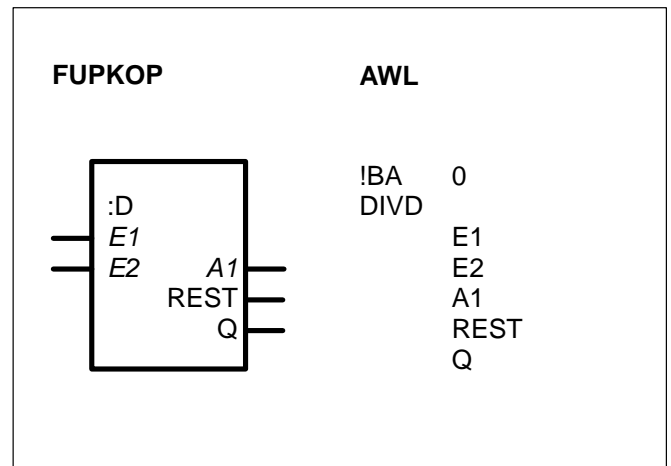


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	J	P	N	0	0
E2	E	W	J	P	N	1	0
A1	A	W	J	P	N	0	0

VE-AWL-Definition

00000	!	PP 0	E1	Eingang WORT
	[1		
00002	:	PP 1	E2	Eingang WORT (Dopplung möglich)
]	1		
00005	=	PP 0	A1	Ausgang WORT

Der Wert des Operanden am Eingang E1 wird durch den Wert des Operanden am Eingang E2 dividiert und das Ergebnis dem Operanden am Ausgang A1, der Rest dem Operanden am Ausgang REST zugewiesen. Falls ein Rest entsteht, wird das Ergebnis stets abgerundet. Liegt das Ergebnis außerhalb des zulässigen Zahlenbereichs, wird es auf den maximalen bzw. minimalen Wert des Zahlenbereichs begrenzt. Fand eine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 1-Signal und dem Ausgang REST der Wert 0 zugewiesen. Fand keine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 0-Signal zugewiesen.



Parameter

E1	DOPPELWORT MD, KD	Dividend
E2	DOPPELWORT MD, KD	Divisor
A1	DOPPELWORT MD	Ergebnis (Quotient)
REST	DOPPELWORT MD	Rest
Q	BINÄR A, M	Ergebnis begrenzt

VE-Daten

Laufzeit:	
Grundlaufzeit:	300 ... 324 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang E1 wird durch den Wert des Operanden am Eingang E2 dividiert und das Ergebnis dem Operanden am Ausgang A1, der Rest dem Operanden am Ausgang REST zugewiesen. Falls ein Rest entsteht, wird das Ergebnis stets abgerundet. Liegt das Ergebnis außerhalb des zulässigen Zahlenbereichs, wird es auf den maximalen bzw. minimalen Wert des Zahlenbereichs begrenzt. Fand eine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 1-Signal und dem Ausgang REST der Wert 0 zugewiesen. Fand keine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 0-Signal zugewiesen.

Eine Division durch "Null" wird somit auch am Binär-Ausgang Q signalisiert.

Die Ein- und Ausgänge sind weder doppelbar noch negierbar.

Restbehandlung

Entsteht bei der Division ein Rest, so steht dieser am Doppelwort-Ausgang REST zur Verfügung. Bei der Division wird das Ergebnis stets abgerundet, falls ein Rest entsteht.

Beispiel:	3 : 3 = 1	Rest 0
	4 : 3 = 1	Rest 1
	5 : 3 = 1	Rest 2
	6 : 3 = 2	Rest 0

Da der Rest am Ausgang REST zur Verfügung steht, kann der Anwender diesen mit dem Divisor vergleichen und das Ergebnis am Ausgang A1 nach eigenen Vorstellungen runden.

Beispiel:

Rest ≥ Divisor/2 → Ergebnis an A1 aufrunden

Division durch "Null"

Hat der Divisor den Wert "Null", so wird dem Ausgang A1 der positive bzw. negative Grenzwert des Zahlenbereiches zugewiesen.

Für die Division durch "Null" gilt:

$A1 = -2\,147\,483\,647$ ($8000\,0001_H$) falls Dividend negativ.

$A1 = +2\,147\,483\,647$ ($7FFF\,FFFF_H$) falls Dividend positiv.

REST = 0 Ausgang für Rest

Q = 1 Ausgang für Signalisierung, daß der Wert am Ausgang A1 begrenzt wurde

Unzulässiger Ergebniswert

Entsteht bei der Division als Ergebnis der unzulässige Wert $8000\,0000_H$, so wird dieser auf den zulässigen Grenzwert $8000\,0001_H$ ($-2\,147\,483\,647$) korrigiert, der Binär-Ausgang Q auf den Wert 1 und der Ausgang REST auf den Wert 0 gesetzt.

Zahlenbereich

Integer Doppelwort (32 Bit).

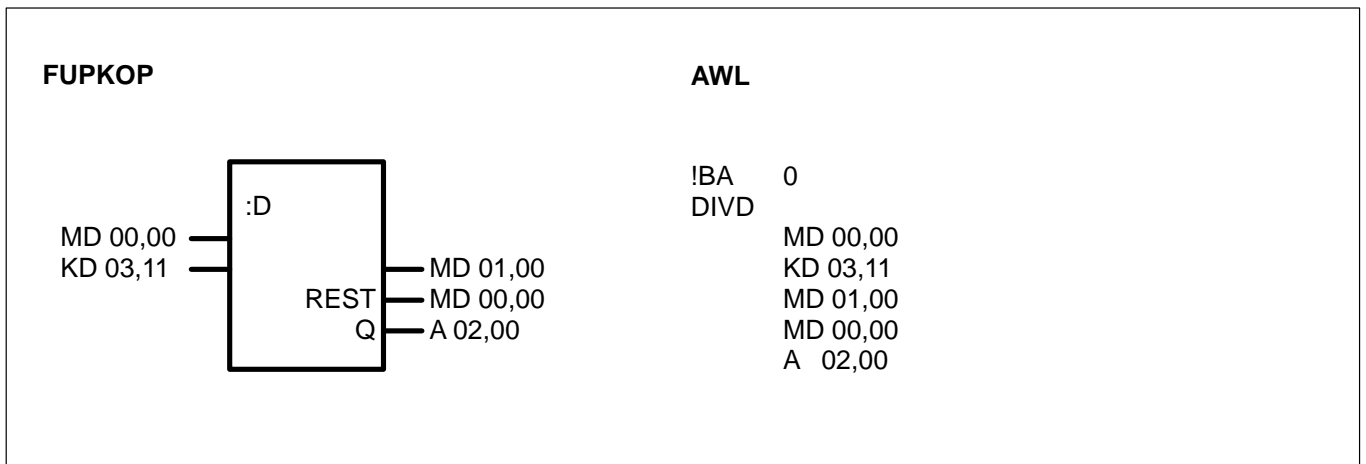
Speziell gilt hier für die Eingänge E1 und E2:

- untere Grenze: $8000\,0000_H$ $-2\,147\,483\,648$

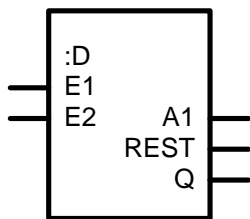
Allgemein gilt:

- untere Grenze: $8000\,0001_H$ $-2\,147\,483\,647$
- obere Grenze: $7FFF\,FFFF_H$ $+2\,147\,483\,647$
- unzulässiger Wert: $8000\,0000_H$ —

Beispiel



VE-FUP-Definition



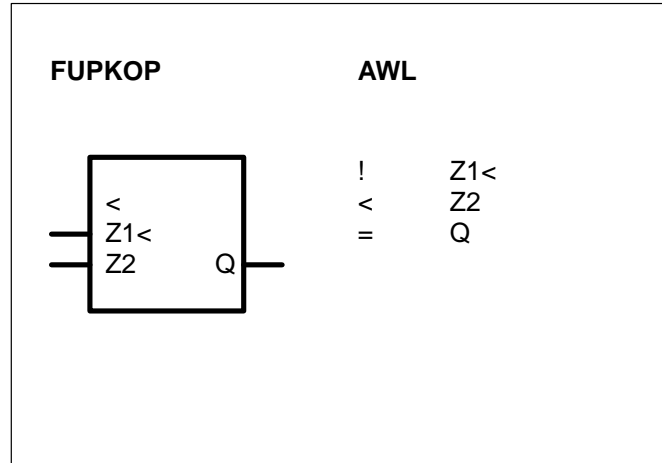
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
E2	E	D	N	P	N	0	0
A1	A	D	N	P	N	0	0
REST	A	D	N	P	J	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DIVD			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	E2	Eingang DOPPELWORT
00004		PP 0	A1	Ausgang DOPPELWORT
00005		PP 0	REST	Ausgang DOPPELWORT
00005		PP 0	Q	Ausgang BINÄR

Der Wert des Operanden am Eingang Z1< wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1< kleiner als an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1< gleich oder größer als Z2 wird Q der Zustand 0 zugewiesen.



Parameter

Z1<	WORT	EW, MW, AW, KW	zu vergleichender Wert
Z2	WORT	EW, MW, AW, KW	Vergleichswert
Q	BINÄR	A, M, S	Ergebnis des Vergleichs

VE-Daten

Laufzeit:

Grundlaufzeit: < 12 μ s

Zusätzliche Laufzeit: —

Aktualisierung der Ausgänge: ja

Anzahl der Vergangenheitswerte: keine

Verfügbar ab: ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang Z1< wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1< kleiner als an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1< gleich oder größer als Z2 wird Q der Zustand 0 zugewiesen.

Die Eingänge sind negierbar und nicht doppelbar. Der Ausgang ist invertierbar und nicht doppelbar.

Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für die nicht negierten Eingänge:

- untere Grenze 8000_H -32768
- obere Grenze 7FFF_H +32767

Allgemein gilt:

- untere Grenze: 8001_H -32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H //

Der Wert 8000_H (-32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

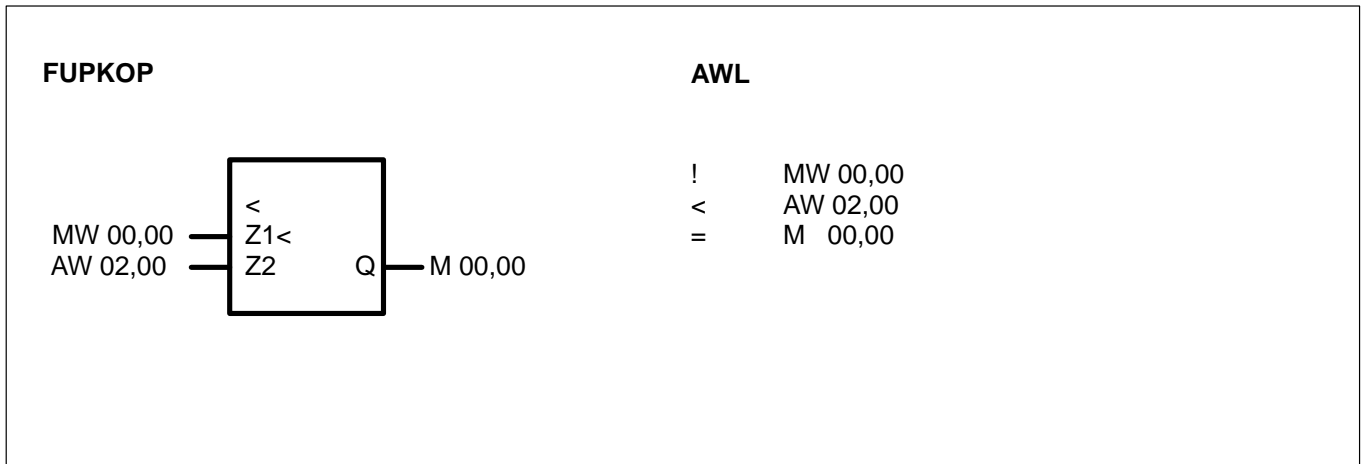
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

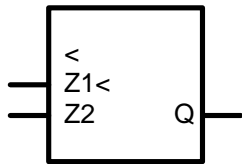
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (-32768) auf den erlaubten Wert 8001_H (-32767) korrigiert.

Beispiel



VE-FUP-Definition



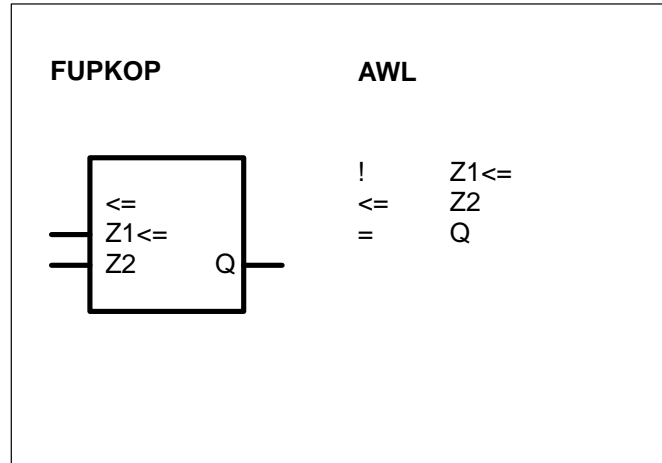
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
Z1<	E	W	J	P	J	0	0
Z2	E	W	J	P	J	0	0
Q	A	L	J	P	J	0	0

VE-AWL-Definition

00000	!	PP	Z1<	Eingang WORT
00002	<	PP	Z2	Eingang WORT
00004	=	PP	Q	Ausgang BINÄR

Der Wert des Operanden am Eingang Z1<= wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1<= kleiner als oder gleich dem Wert an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1<= größer als Z2 wird Q der Zustand 0 zugewiesen.



Parameter

Z1<=	WORT	EW, MW, AW, KW	zu vergleichender Wert
Z2	WORT	EW, MW, AW, KW	Vergleichswert
Q	BINÄR	A, M, S	Ergebnis des Vergleichs

VE-Daten

Laufzeit:		
Grundlaufzeit:	< 12 µs	
Zusätzliche Laufzeit:	—	
Aktualisierung der Ausgänge:	ja	
Anzahl der Vergangenheitswerte:	keine	
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	

Beschreibung

Der Wert des Operanden am Eingang Z1<= wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1<= kleiner als oder gleich dem Wert an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1<= größer als Z2 wird Q der Zustand 0 zugewiesen.

Die Eingänge sind negierbar und nicht doppelbar. Der Ausgang ist invertierbar und nicht doppelbar.

Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für die nicht negierten Eingänge:

- untere Grenze 8000_H -32768
- obere Grenze 7FFF_H +32767

Allgemein gilt:

- untere Grenze: 8001_H -32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (-32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

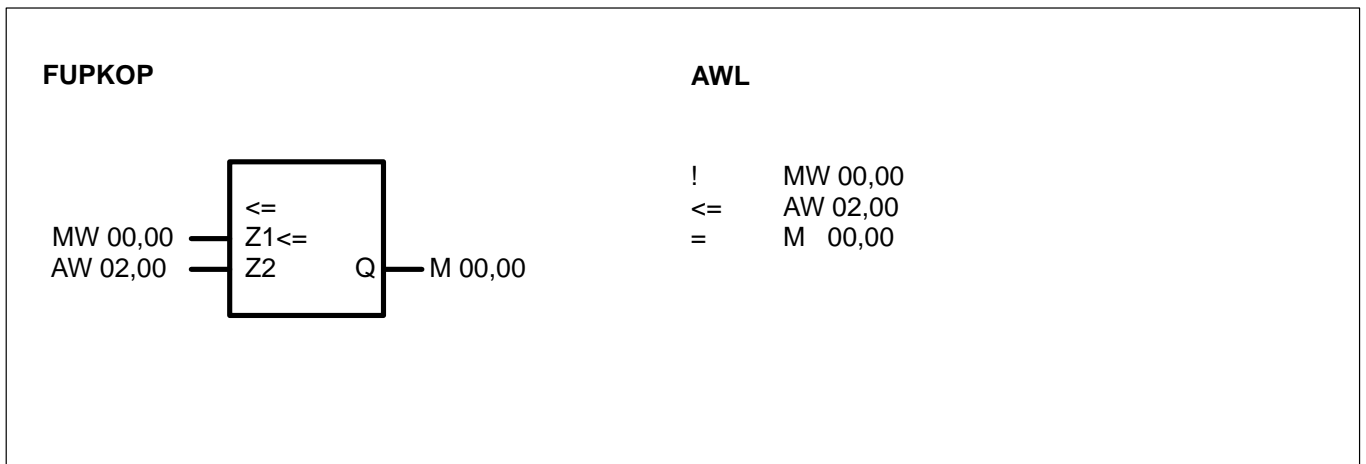
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

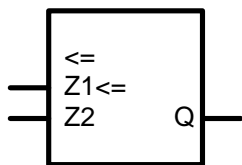
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (-32768) auf den erlaubten Wert 8001_H (-32767) korrigiert.

Beispiel



VE-FUP-Definition

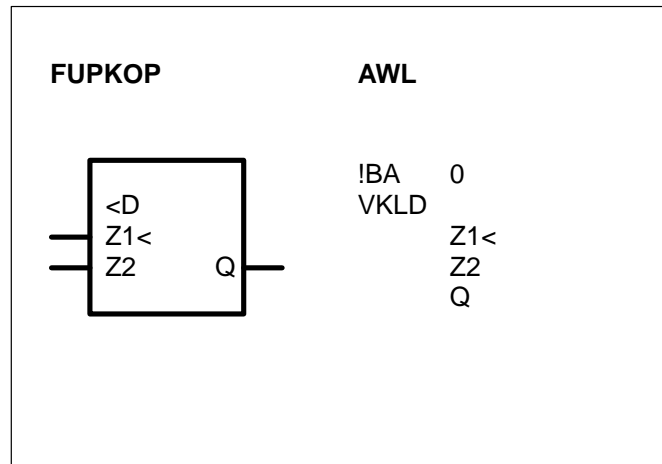


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
Z1<=	E	W	J	P	J	0	0
Z2	E	W	J	P	J	0	0
Q	A	L	J	P	J	0	0

VE-AWL-Definition

00000	!	PP 0	Z1<=	Eingang WORT
00002	<	PP 0	Z2	Eingang WORT
00004	=	PP 0	Q	Ausgang BINÄR

Der Wert des Operanden am Eingang Z1< wird mit dem Wert des Operanden am Eingang Z2 verglichen. Ist der Wert an Z1< kleiner als an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1< gleich oder größer Z2 wird Q der Zustand 0 zugewiesen.



Parameter

Z1<	DOPPELWORT MD, KD	zu vergleichender Wert
Z2	DOPPELWORT MD, KD	Vergleichswert
Q	BINÄR A, M,	Ergebnis des Vergleichs

VE-Daten

Laufzeit:	
Grundlaufzeit:	35 ... 36 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang Z1< wird mit dem Wert des Operanden am Eingang Z2 verglichen. Ist der Wert an Z1< kleiner als an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1< gleich oder größer Z2 wird Q der Zustand 0 zugewiesen.

Die Eingänge sind negierbar und nicht doppelbar. Der Ausgang ist invertierbar und nicht doppelbar.

Zahlenbereich

Integer Doppelwort (32 Bit)

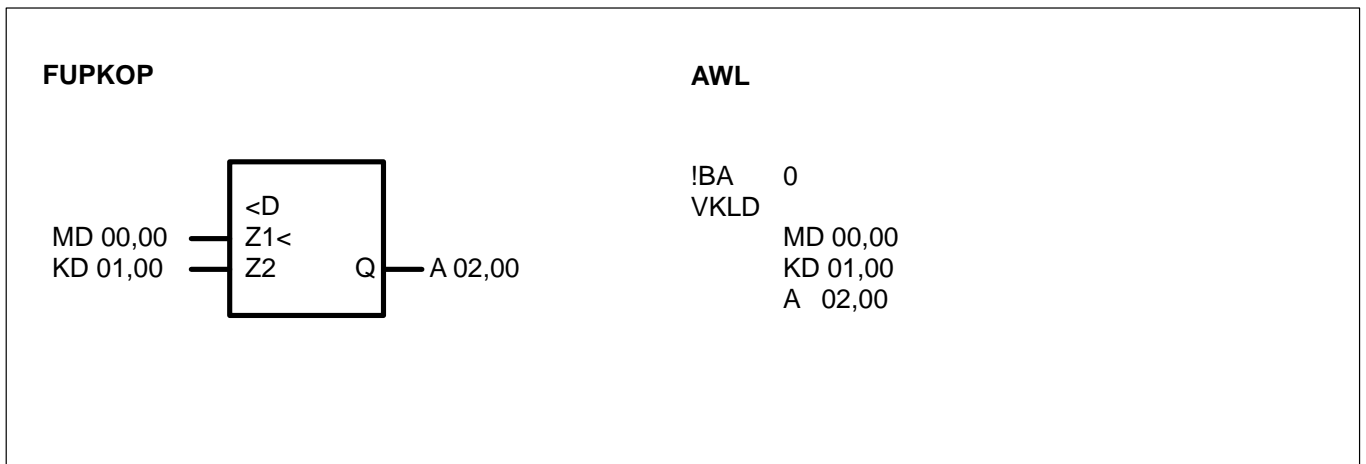
Es gilt hier speziell für die Eingänge Z1< und Z2:

- untere Grenze: 8000 0000_H -2 147 483 648
- obere Grenze: 7FFF FFFF_H +2 147 483 647

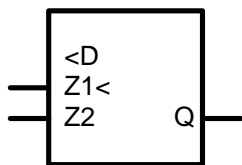
Allgemein gilt:

- untere Grenze: 8000 0001_H -2 147 483 647
- obere Grenze: 7FFF FFFF_H +2 147 483 647
- unzulässiger Wert: 8000 0000_H —

Beispiel



VE-FUP-Definition

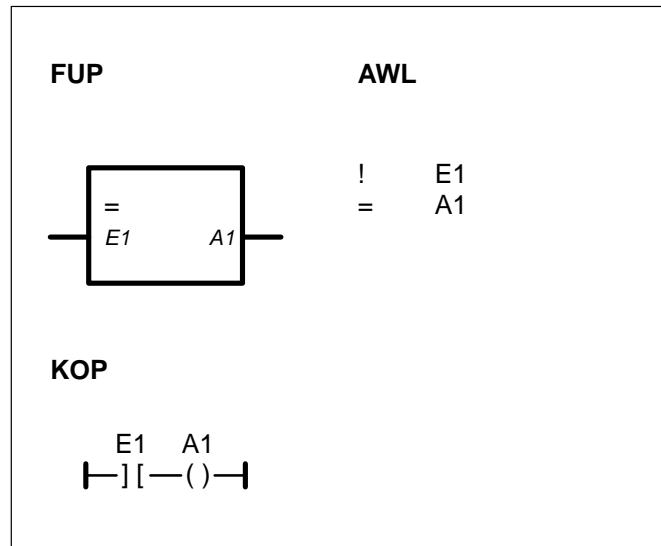


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
Z1<	E	D	N	P	J	0	0
Z2	E	D	N	P	J	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	VKLD			
00002		PP 0	Z1<	Eingang DOPPELWORT
00003		PP 0	Z2	Eingang DOPPELWORT
00004		PP 0	Q	Ausgang BINÄR

Das Verknüpfungselement weist den Wert des Operanden am Eingang dem Operanden am Ausgang zu.



Parameter

E1	BINÄR	E, M, A, S, K	Quelle
A1	BINÄR	M, A, S	Ziel, doppelbarer Ausgang

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

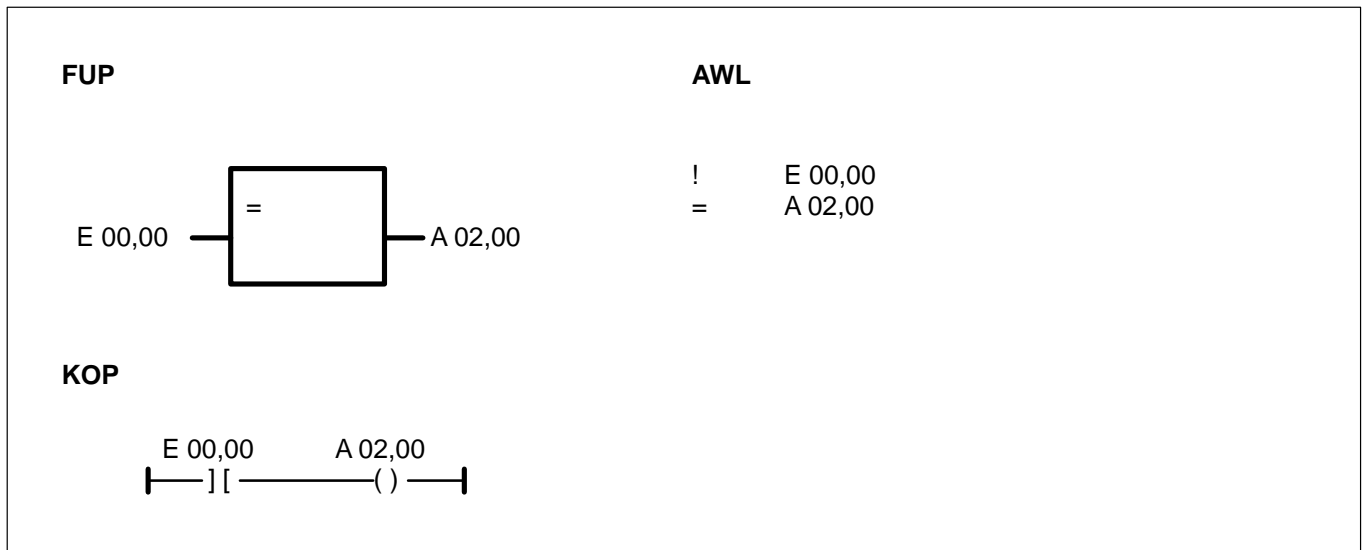
Grundlaufzeit:	4,3 µs	14 µs
Zusätzliche Laufzeit:	2,3 µs pro zusätzl. Ausgang (A2 ... An)	8 µs
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

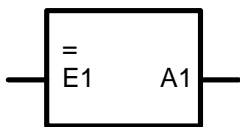
Das Verknüpfungselement weist den Wert des Operanden am Eingang dem Operanden am Ausgang zu.

Der Ausgang A1 ist doppelbar. Der Eingang und der Ausgang sind invertierbar.

Beispiel



VE-FUP-Definition

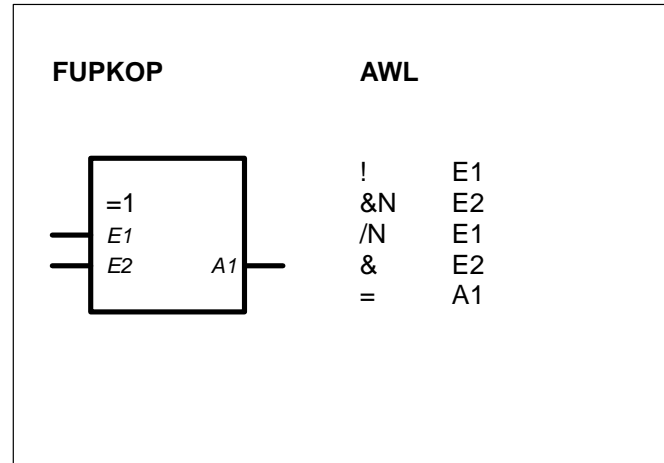


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	L	J	P	N	0	0
A1	A	L	J	P	N	1	0

VE-AWL-Definition

00000	!	PP 0	E1	Eingang BINÄR
	[1		
00002	=	PP 1	A1	Ausgang BINÄR (Dopplung möglich)
]	1		

Das Verknüpfungselement realisiert eine logische EXKLUSIV-ODER-Verknüpfung der Operanden an den Eingängen. Das Ergebnis wird dem Operanden am Ausgang zugewiesen.



Parameter

E1	BINÄR	E, M,A,S,K	Operand 1 der XOR-Verknüpfung
E2	BINÄR	E,M,A,S,K,	Operand 2 der XOR-Verknüpfung
A1	BINÄR	M,A,S	Ergebnis der XOR-Verknüpfung

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

10,9 µs

32 µs

Zusätzliche Laufzeit:

—

—

Aktualisierung der Ausgänge:

ja

ja

Anzahl der Vergangenheitswerte:

keine

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

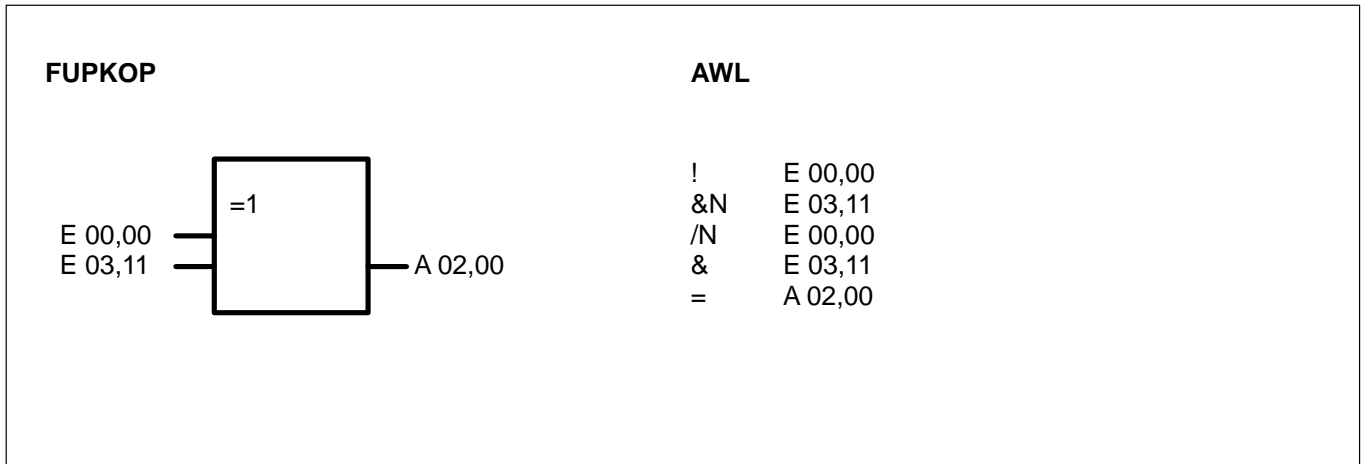
Das Verknüpfungselement realisiert eine logische EXKLUSIV-ODER-Verknüpfung der Operanden an den Eingängen. Das Ergebnis wird dem Operanden am Ausgang zugewiesen.

Eine Invertierung und Dopplung der Eingänge und des Ausgangs ist nicht möglich.

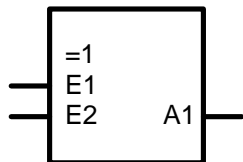
Wahrheitstabelle:

E1	E2	A1
0	0	0
1	0	1
0	1	1
1	1	0

Beispiel



VE-FUP-Definition

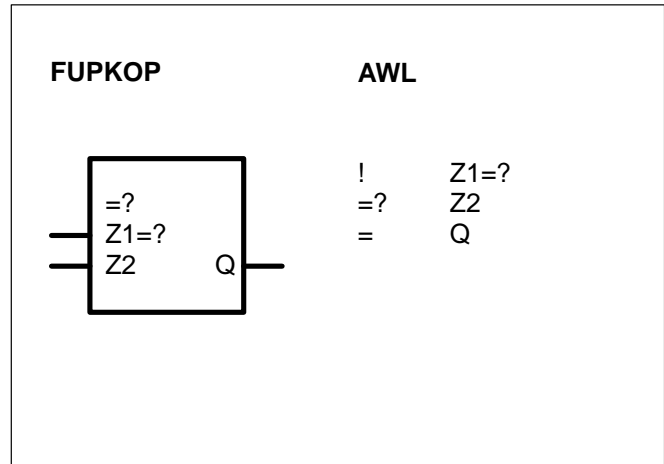


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	L	N	P	N	0	0
E2	E	L	N	P	N	0	0
A1	A	L	N	P	N	0	0

VE-AWL-Definition

00000	!	PP 0	E1	Eingang BINÄR
00002	&N	PP 0	E2	Eingang BINÄR
00004	/N	PP 0	E1	Eingang BINÄR
00006	&	PP 0	E2	Eingang BINÄR
00008	=	PP 0	A1	Ausgang BINÄR

Der Wert des Operanden am Eingang Z1=? wird mit dem Wert des Operanden am Eingang Z2 verglichen. Ist der Wert an Z1=? gleich dem an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1=? ungleich Z2 wird Q der Zustand 0 zugewiesen.



Parameter

Z1=?	WORT	EW, MW, AW, KW	zu vergleichender Wert
Z2	WORT	EW, MW, AW, KW	Vergleichswert
Q	BINÄR	A, M, S	Ergebnis des Vergleichs

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

< 12 μ s

< 47 μ s

Zusätzliche Laufzeit:

—

—

Aktualisierung der Ausgänge:

ja

ja

Anzahl der Vergangenheitswerte:

keine

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Wert des Operanden am Eingang Z1=? wird mit dem Wert des Operanden am Eingang Z2 verglichen. Ist der Wert an Z1=? gleich dem an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1=? ungleich Z2 wird Q der Zustand 0 zugewiesen.

Die Eingänge sind negierbar und nicht doppelbar. Der Ausgang ist invertierbar und nicht doppelbar.

Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für die nicht negierten Eingänge:

- untere Grenze 8000_H -32768
- obere Grenze 7FFF_H +32767

Allgemein gilt:

- untere Grenze: 8001_H -32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (-32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

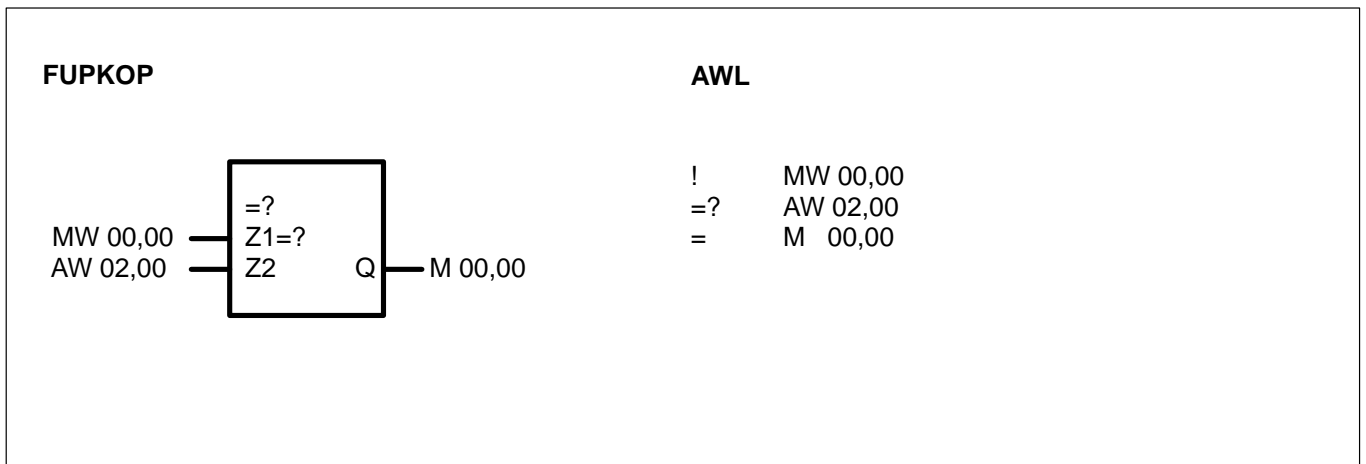
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

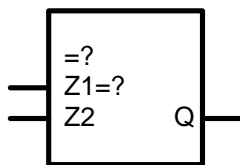
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (-32768) auf den erlaubten Wert 8001_H (-32767) korrigiert.

Beispiel



VE-FUP-Definition

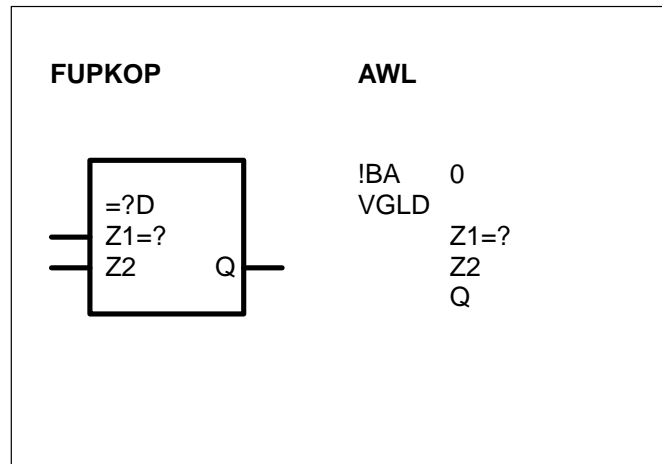


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
Z1=?	E	W	J	P	J	0	0
Z2	E	W	J	P	J	0	0
Q	A	L	J	P	J	0	0

VE-AWL-Definition

00000	!	PP 0	Z1=?	Eingang WORT
00002	=?	PP 0	Z2	Eingang WORT
00004	=	PP 0	Q	Ausgang BINÄR

Der Wert des Operanden am Eingang Z1=? wird mit dem Wert des Operanden am Eingang Z2 verglichen. Ist der Wert an Z1=? gleich dem an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1=? ungleich Z2 wird Q der Zustand 0 zugewiesen.



Parameter

Z1=?	DOPPELWORT MD, KD	zu vergleichender Wert
Z2	DOPPELWORT MD, KD	Vergleichswert
Q	BINÄR A, M	Ergebnis des Vergleichs

VE-Daten

Laufzeit:	
Grundlaufzeit:	35 ... 37 μ s
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang Z1=? wird mit dem Wert des Operanden am Eingang Z2 verglichen. Ist der Wert an Z1=? gleich dem an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1=? ungleich Z2 wird Q der Zustand 0 zugewiesen.

Die Eingänge sind weder doppelbar noch negierbar. Der Ausgang ist weder doppelbar noch invertierbar.

Zahlenbereich

Integer Doppelwort (32 Bit)

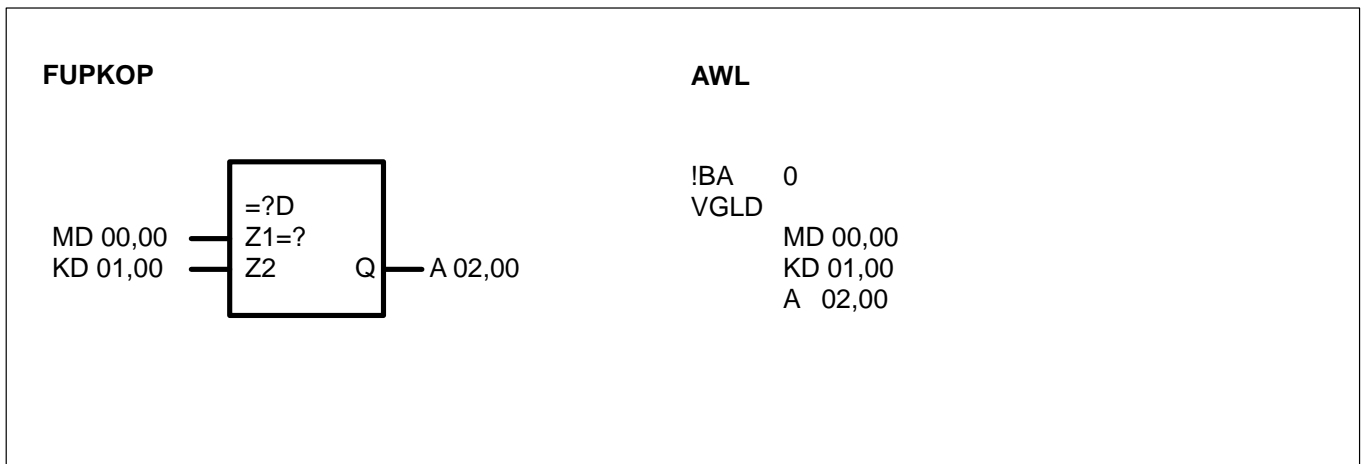
Es gilt hier speziell für die Eingänge Z1=? und Z2:

- untere Grenze: 8000 0000_H -2 147 483 648
- obere Grenze: 7FFF FFFF_H +2 147 483 647

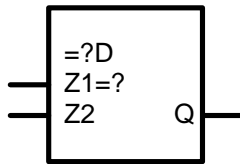
Allgemein gilt:

- untere Grenze: 8000 0001_H -2 147 483 647
- obere Grenze: 7FFF FFFF_H +2 147 483 647
- unzulässiger Wert: 8000 0000_H —

Beispiel



VE-FUP-Definition

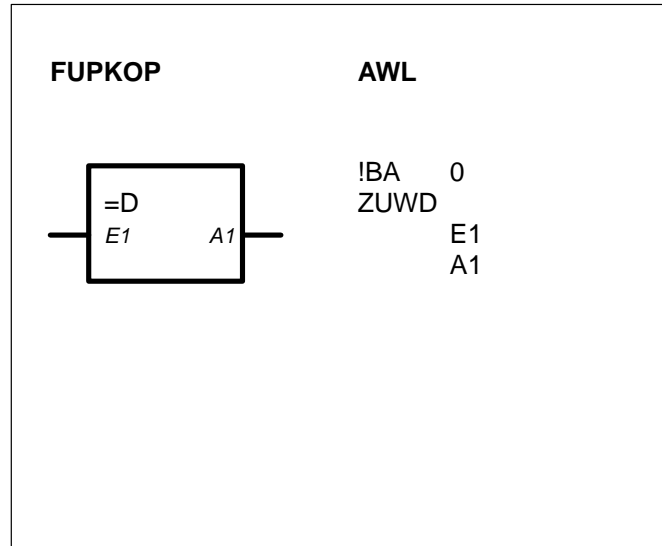


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
Z1=?	E	D	N	P	J	0	0
Z2	E	D	N	P	J	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	VGLD			
00002		PP 0	Z1=?	Eingang DOPPELWORT
00003		PP 0	Z2	Eingang DOPPELWORT
00004		PP 0	Q	Ausgang BINÄR

Der Wert des Operanden am Eingang E1 wird dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	DOPPELWORT MD, KD	Quelle
A1	DOPPELWORT MD	Ziel

VE-Daten

Laufzeit:	
Grundlaufzeit:	31 μ s
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang E1 wird dem Operanden am Ausgang A1 zugewiesen.

Liegt am Eingang aus irgendeinem Grund der unzulässige Wert 8000 0000_H an, so wird dem Ausgang A1 der zulässige Wert 8000 0001_H (−2 147 483 647) zugewiesen. Der unzulässige Wert wird also korrigiert.

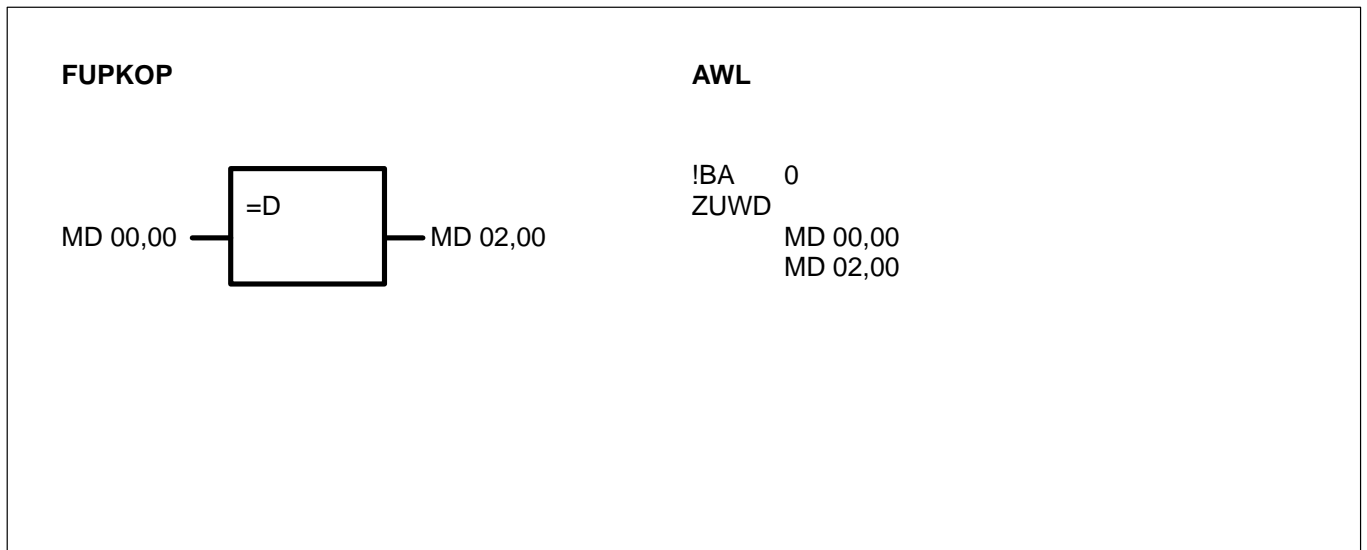
Der Eingang und der Ausgang sind weder doppelbar noch negierbar.

Zahlenbereich

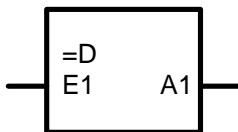
Integer Doppelwort (32 Bit)

- untere Grenze: 8000 0001_H −2 147 483 647
- obere Grenze: 7FFF FFFF_H +2 147 483 647
- unzulässiger Wert: 8000 0000_H —

Beispiel



VE-FUP-Definition

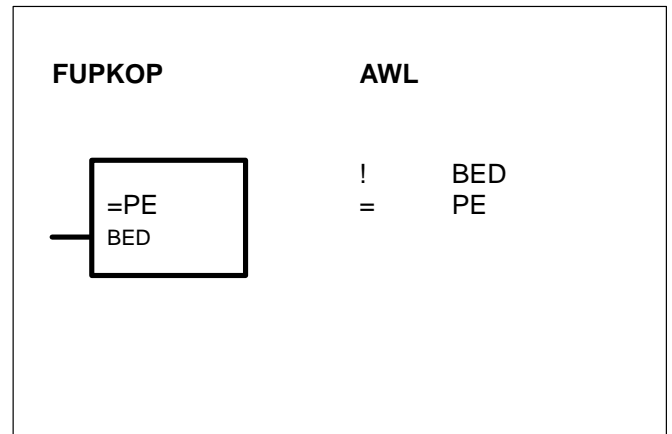


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
A1	A	D	N	P	N	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	ZUWD			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	A1	Ausgang DOPPELWORT

Das bedingte Programmende kann innerhalb eines SPS-Programms projiziert werden. In Abhängigkeit vom Zustand des Operanden am Eingang BED wird die Bearbeitung des SPS-Programms an dieser Stelle beendet oder nicht beendet.



Parameter

BED	BINÄR	E, M, A, S, K	Bedingung für Programmende
-----	-------	---------------	----------------------------

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

2 µs

4 µs

Zusätzliche Laufzeit:

—

—

Aktualisierung der Ausgänge:

entfällt

entfällt

Anzahl der Vergangenheitswerte:

keine

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Das bedingte Programmende kann innerhalb eines SPS-Programms projiziert werden. In Abhängigkeit vom Zustand des Operanden am Eingang BED wird die Bearbeitung des SPS-Programms an dieser Stelle beendet oder nicht beendet.

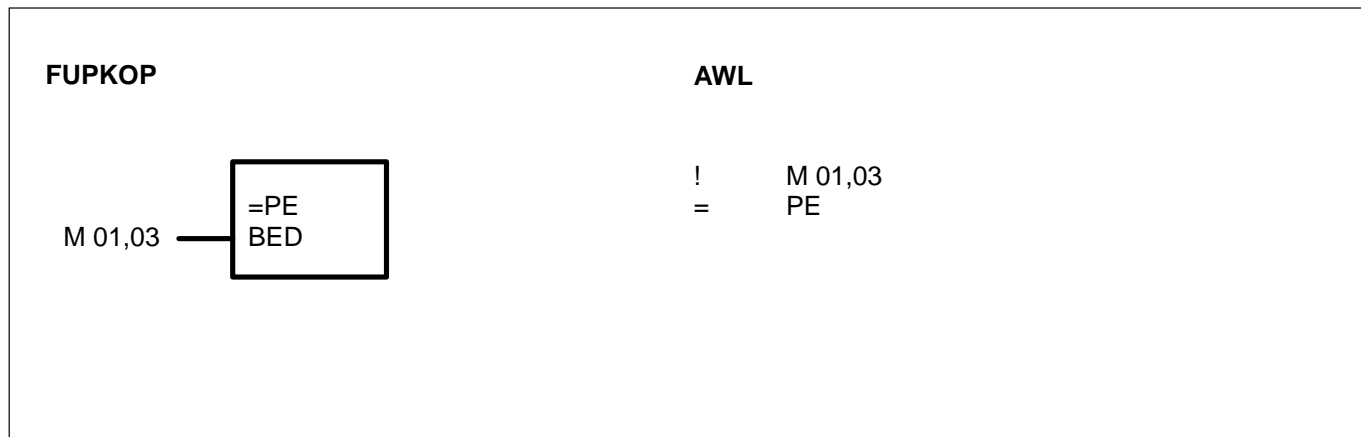
Für die Bedingung am Eingang BED gilt:

BED = 0: das SPS-Programm wird weiter bearbeitet

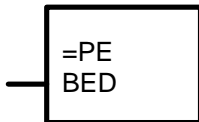
BED = 1: das SPS-Programm wird nur bis zu dieser Stelle bearbeitet. Der nachfolgende SPS-Programmteil wird *nicht* bearbeitet.

Der Eingang ist weder doppelbar noch invertierbar.

Beispiel



VE-FUP-Definition



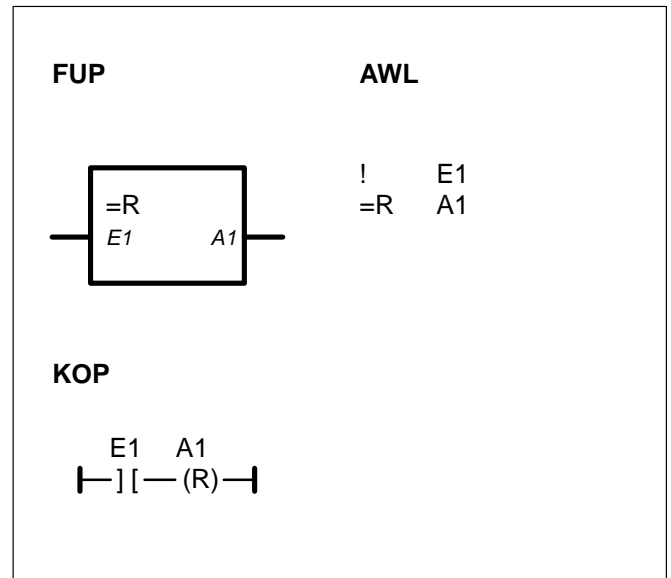
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
BED	E	L	N	P	J	0	0

VE-AWL-Definition

00000	!	PP 0	BED	Bedingung für Prg.ende (BINÄR)
00002	=PE			

Mit diesem Verknüpfungselement wird eine gespeicherte Binär-Variable zurückgesetzt.

Zustand 1 am Eingang setzt den Operanden am Ausgang auf Zustand 0. Zustand 0 am Eingang hat keinen Einfluß auf den Operanden am Ausgang.



Parameter

E1	BINÄR	E, M, A, S, K	Rücksetzbedingung
A1	BINÄR	M, A	Speicher-Variable

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

Zusätzliche Laufzeit:

4,3 µs

2,3 µs pro zusätzl. Ausgang
(A3 ... An)

5 µs

9 µs

Aktualisierung der Ausgänge:

ja

ja

Anzahl der Vergangenheitswerte:

keine

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

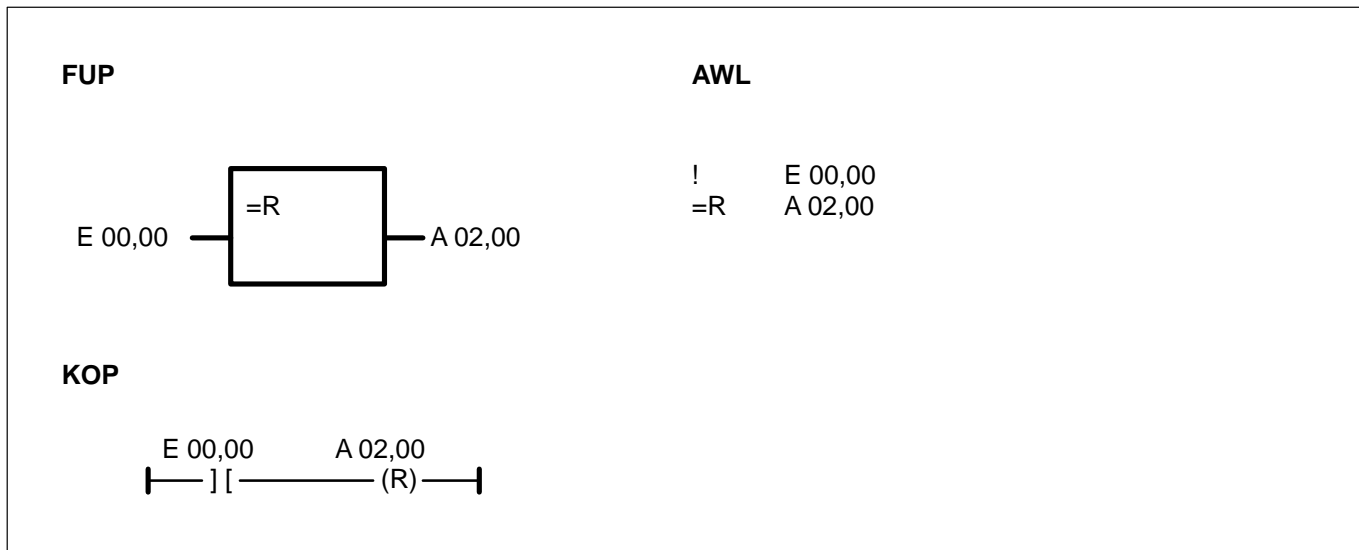
Zustand 1 am Eingang setzt den Operanden am Ausgang auf Zustand 0. Zustand 0 am Eingang hat keinen Einfluß auf den Operanden am Ausgang.

ACHTUNG:

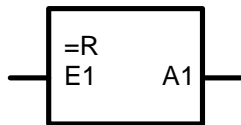
Dieses VE darf nur als Ausgangs-VE verwendet werden, d. h. es darf im FUP ausgangsseitig nicht durch eine Linie weiterverbunden werden. Am Ausgang *muß* ein Operand (M oder A) angegeben werden.

Der Ausgang A1 ist doppelbar (A2 ... An). Der Eingang E1 ist invertierbar.

Beispiel



VE-FUP-Definition



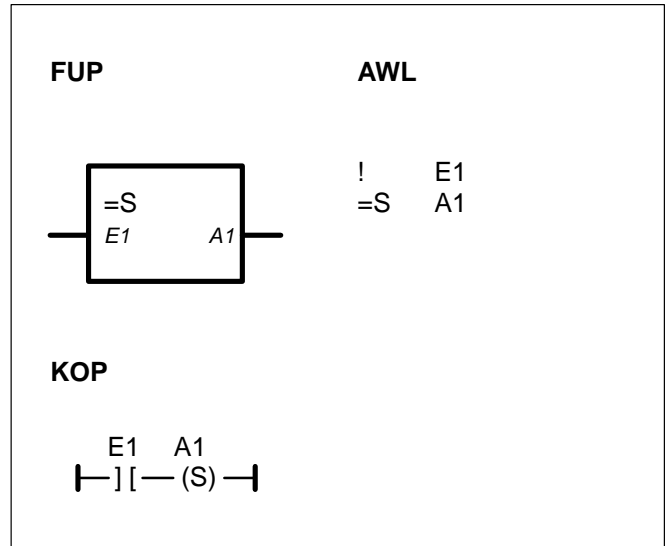
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	L	J	P	N	0	0
A1	A	L	N	P	N	1	0

VE-AWL-Definition

00000	!	PP 0	E1	Eingang BINÄR
	[1		
00002	=R	PP 1	A1	Ausgang BINÄR (Dopplung möglich)
]	1		

Mit diesem Verknüpfungselement wird eine Binär-Variablen speichernd gesetzt.

Zustand 1 am Eingang setzt den Operanden am Ausgang auf den Zustand 1. Zustand 0 am Eingang hat keinen Einfluß auf den Operanden am Ausgang.



Parameter

E1	BINÄR	E, M, A, S, K	Setzbedingung
A1	BINÄR	M, A	Speicher-Variable

VE-Daten

Serie 90

Serie 30, 40, 50

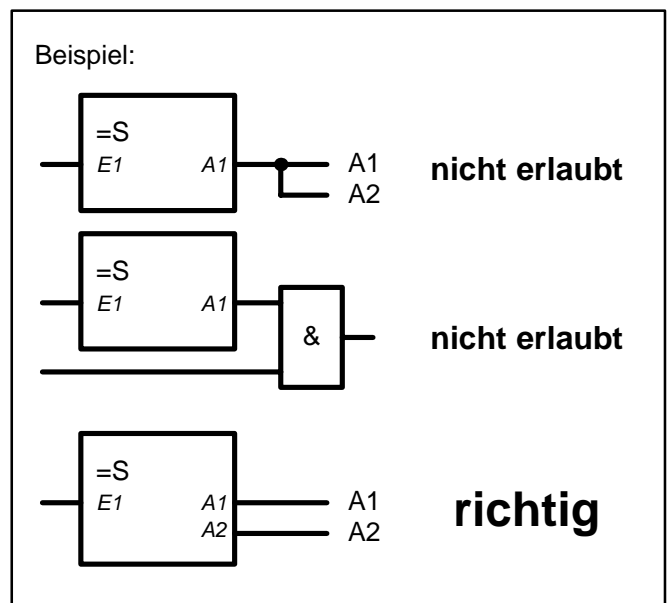
Laufzeit:		
Grundlaufzeit:	4,3 µs	15 µs
Zusätzliche Laufzeit:	2,3 µs pro zusätzl. Ausgang (A3 ... An)	9 µs
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

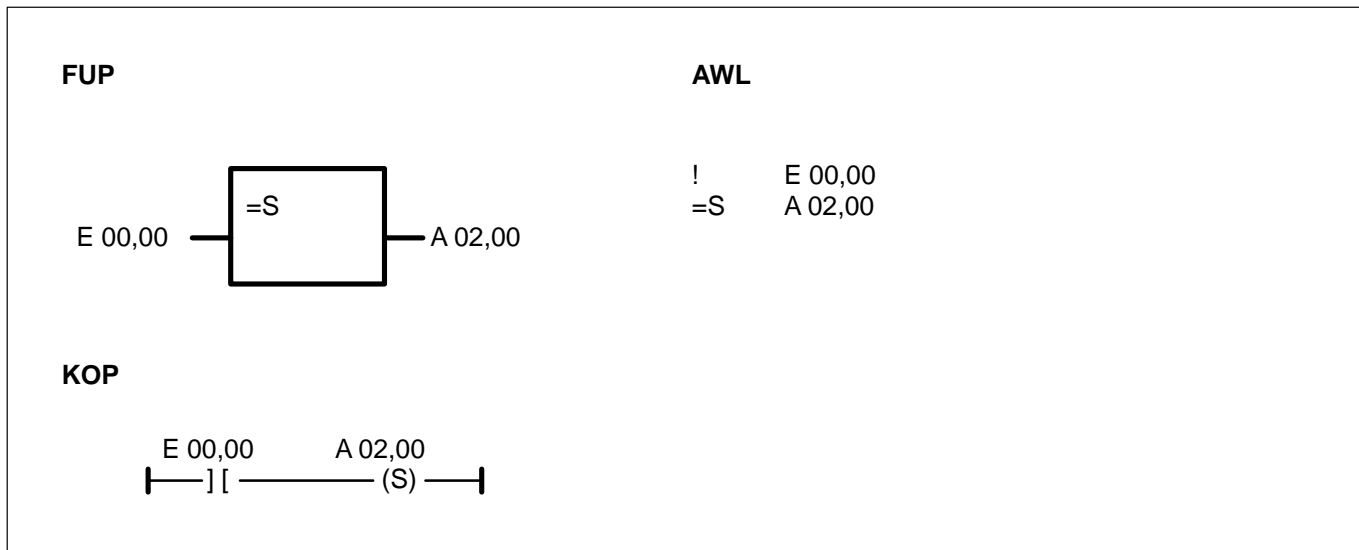
Zustand 1 am Eingang setzt den Operanden am Ausgang auf den Zustand 1. Zustand 0 am Eingang hat keinen Einfluß auf den Operanden am Ausgang.

ACHTUNG:
Dieses VE darf nur als Ausgangs-VE verwendet werden, d. h. es darf im FUP ausgangsseitig nicht durch eine Linie weiter verknüpft werden. Am Ausgang A1 muß ein Operand (M oder A) angegeben werden (siehe Beispiel).

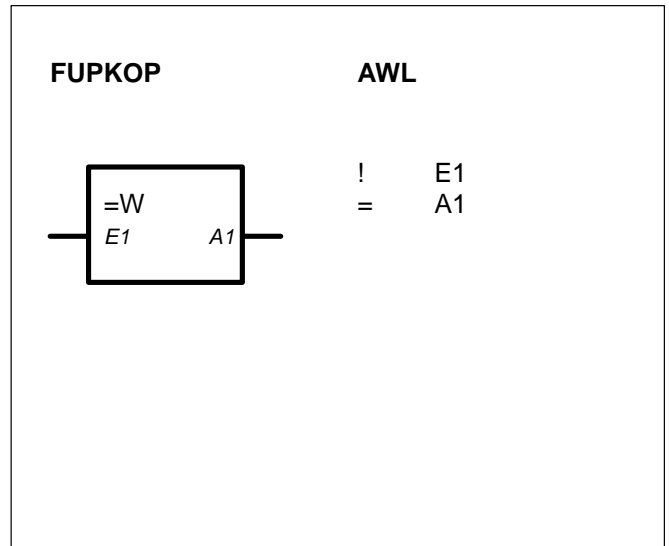
Der Ausgang A1 ist doppelbar (A2 ... An). Der Eingang E1 ist invertierbar.



Beispiel



Das Verknüpfungselement weist den Wert des Operanden am Eingang dem doppelbaren Ausgang zu.



Parameter

E1	WORT	EW, MW, AW, KW	Quelle
A1	WORT	MW,AW	Ziel; doppelbarer Ausgang

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	7 µs	28 µs
Zusätzliche Laufzeit:	5 µs pro zusätzl. Ausgang	14 µs
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Das Verknüpfungselement weist den Wert des Operanden am Eingang dem doppelbaren Ausgang zu.

Der Ausgang A1 ist doppelbar (A2 ... An). Der Eingang und der Ausgang sind negierbar.

Zahlenbereich

Integer Wort (16 Bit).

Speziell gilt hier für den nicht negierten Eingang E1: Liegt am Eingang E1 der unzulässige Wert 8000_H (–32768) an, so wird dem Ausgang A1 der zulässige Wert 8001_H (–32767) zugewiesen.

Allgemein gilt:

- untere Grenze: 8001_H –32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (–32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

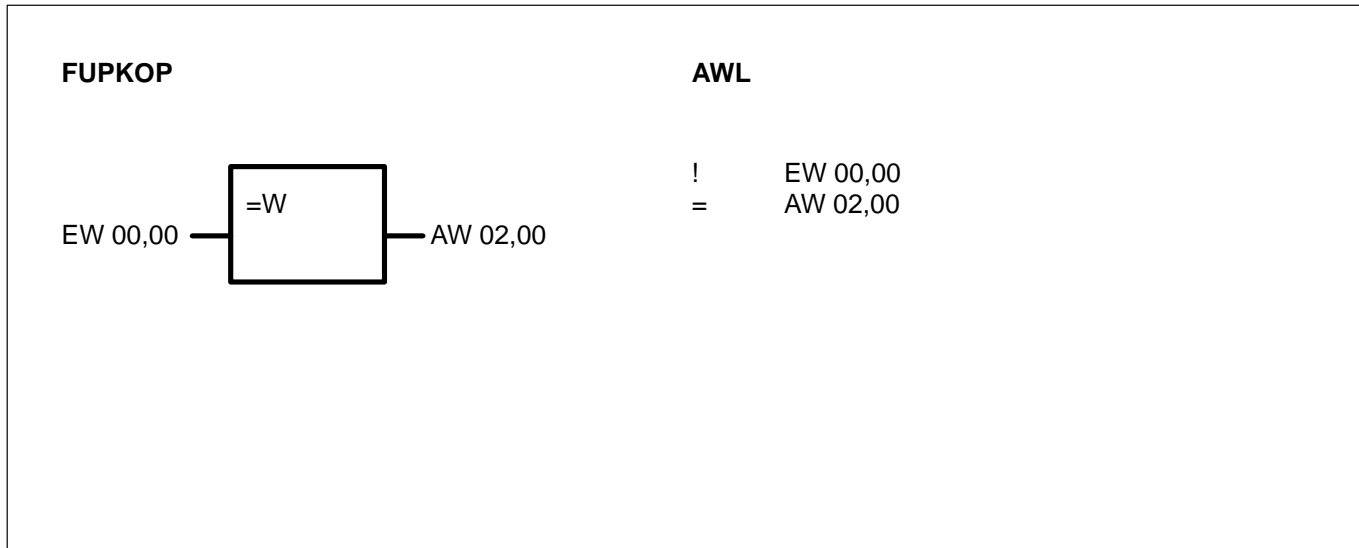
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

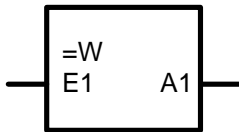
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (–32768) auf den erlaubten Wert 8001_H (–32767) korrigiert.

Beispiel



VE-FUP-Definition



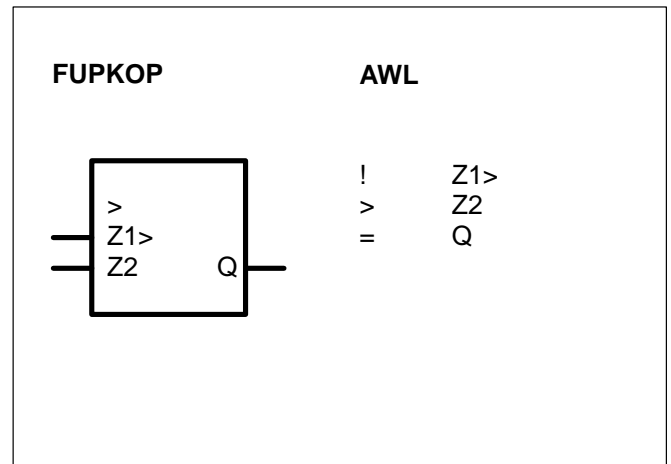
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	J	P	N	0	0
A1	A	W	J	P	N	1	0

VE-AWL-Definition

00001	!	PP 0	E1	Eingang WORT
	[1		
00002	=	PP 1	A1	Ausgang WORT (Dopplung möglich)
]	1		

Der Wert des Operanden am Eingang Z1> wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1> größer als der an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1> gleich oder kleiner Z2 wird Q der Zustand 0 zugewiesen.



Parameter

Z1>	WORT	EW, MW, AW, KW	zu vergleichender Wert
Z2	WORT	EW, MW, AW, KW	Vergleichswert
Q	BINÄR	A, M, S	Ergebnis des Vergleichs

VE-Daten

Laufzeit:	
Grundlaufzeit:	< 12 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang Z1> wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1> größer als der an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1> gleich oder kleiner Z2 wird Q der Zustand 0 zugewiesen.

Die Eingänge sind negierbar und nicht doppelbar. Der Ausgang ist invertierbar und nicht doppelbar.

Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für die nicht negierten Eingänge:

- untere Grenze 8000_H –32768
- obere Grenze 7FFF_H +32767

Allgemein gilt:

- untere Grenze: 8001_H –32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (–32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

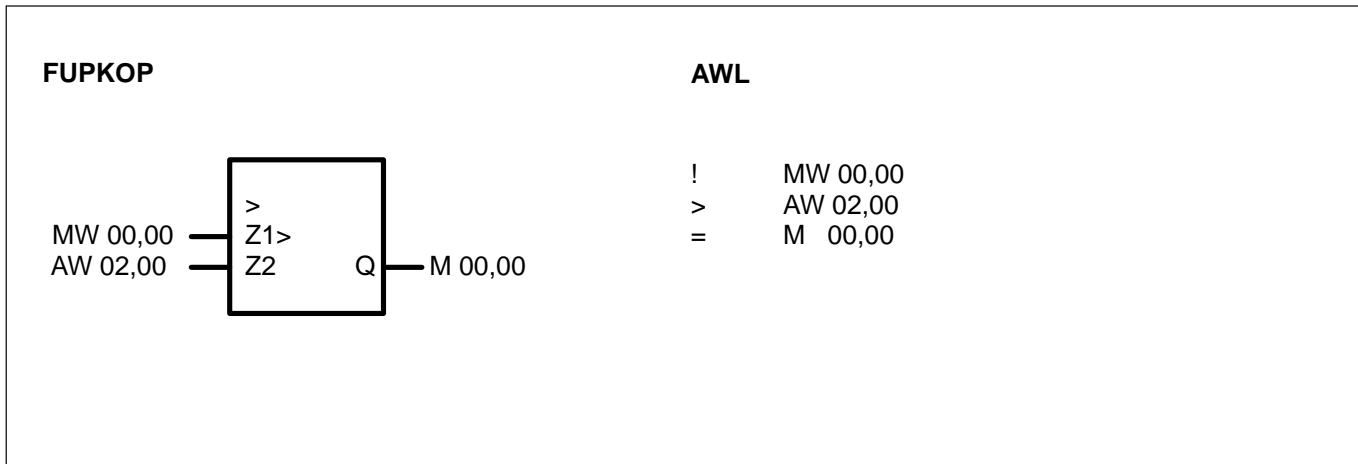
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

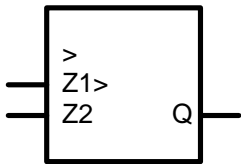
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (–32768) auf den erlaubten Wert 8001_H (–32767) korrigiert.

Beispiel



VE-FUP-Definition



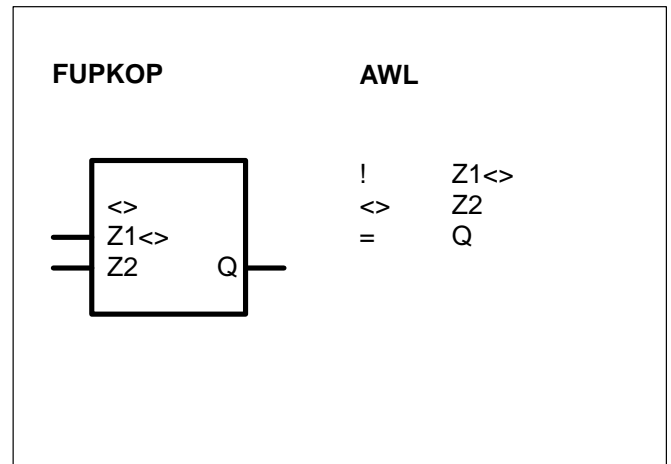
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
Z1>	E	W	J	P	J	0	0
Z2	E	W	J	P	J	0	0
Q	A	L	J	P	J	0	0

VE-AWL-Definition

00000	!	PP 0	Z1>	Eingang WORT
00002	>	PP 0	Z2	Eingang WORT
00004	=	PP 0	Q	Ausgang BINÄR

Der Wert des Operanden am Eingang Z1<> wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1<> größer oder kleiner als der an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1<> gleich Z2, wird Q der Zustand 0 zugewiesen.



Parameter

Z1<>	WORT	EW, MW, AW, KW	zu vergleichender Wert
Z2	WORT	EW, MW, AW, KW	Vergleichswert
Q	BINÄR	A, M, S	Ergebnis des Vergleichs

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit: < 12 µs

Zusätzliche Laufzeit: —

Aktualisierung der Ausgänge: ja

Anzahl der Vergangenheitswerte: keine

Verfügbar ab: ABB Procontic CS31 / 907 PC 331

< 49 µs

—

ja

keine

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Wert des Operanden am Eingang Z1<> wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1<> größer oder kleiner als der an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1<> gleich Z2, wird Q der Zustand 0 zugewiesen.

Die Eingänge sind negierbar und nicht doppelbar. Der Ausgang ist invertierbar und nicht doppelbar.

Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für die nicht negierten Eingänge:

- untere Grenze 8000_H –32768
- obere Grenze 7FFF_H +32767

Allgemein gilt:

- untere Grenze: 8001_H –32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

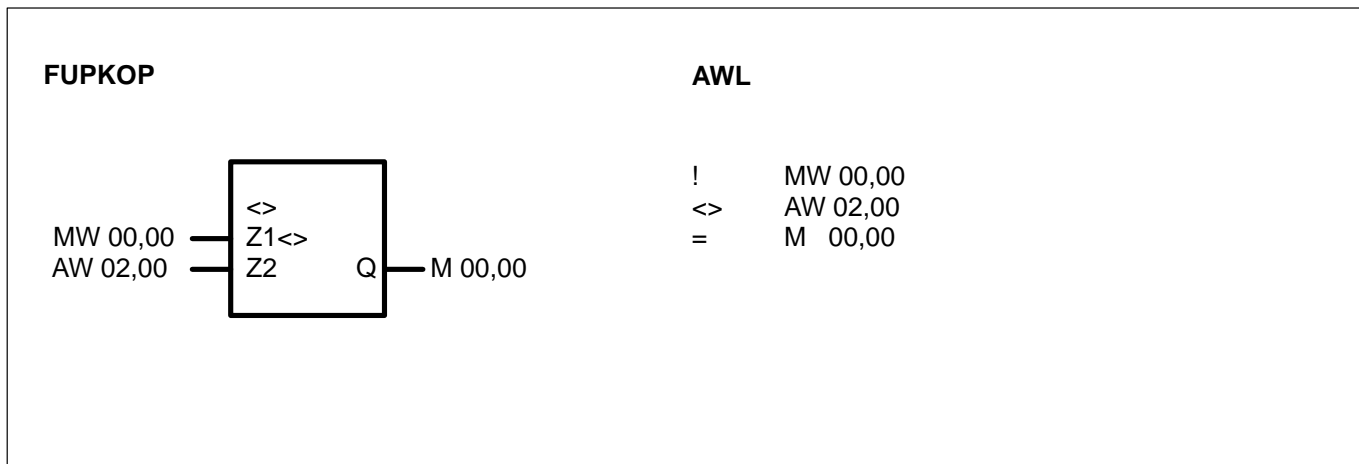
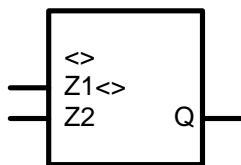
Der Wert 8000_H (–32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (–32768) auf den erlaubten Wert 8001_H (–32767) korrigiert.

Beispiel**VE-FUP-Definition**

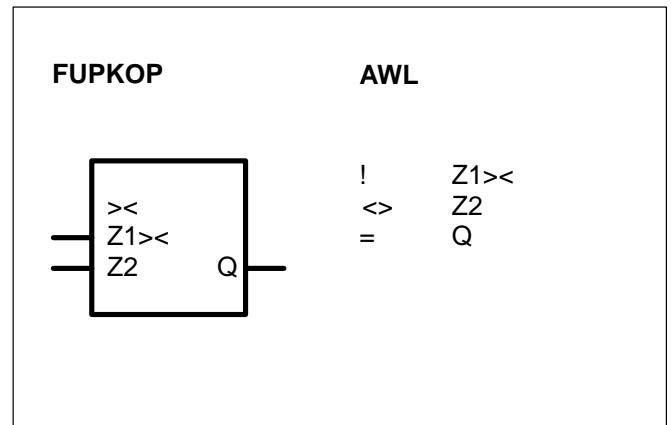
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
Z1<>	E	W	J	P	J	0	0
Z2	E	W	J	P	J	0	0
Q	A	L	J	P	J	0	0

VE-AWL-Definition

00000	!	PP 0	Z1<>	Eingang WORT
00002	<>	PP 0	Z2	Eingang WORT
00004	=	PP 0	Q	Ausgang BINÄR

Der Wert des Operanden am Eingang Z1>< wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1>< größer oder kleiner als der an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1>< gleich Z2, wird Q der Zustand 0 zugewiesen.



Parameter

Z1<>	WORT	EW, MW, AW, KW	zu vergleichender Wert
Z2	WORT	EW, MW, AW, KW	Vergleichswert
Q	BINÄR	A, M, S	Ergebnis des Vergleichs

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	< 12 μ s	< 49 μ s
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Wert des Operanden am Eingang Z1>< wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1>< größer oder kleiner als der an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1>< gleich Z2, wird Q der Zustand 0 zugewiesen.

Die Eingänge sind negierbar und nicht doppelbar. Der Ausgang ist invertierbar und nicht doppelbar.

Anmerkung:

Wegen der Kompatibilität (Schaltzeichen und Symbole) zur Procontic b wird im FUP der Aufruf >< für die Abfrage auf "Ungleich" verwendet, obwohl der Operator in der AWL das <> ist.

Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für die nicht negierten Eingänge:

- untere Grenze 8000_H –32768
- obere Grenze 7FFF_H +32767

Allgemein gilt:

- untere Grenze: 8001_H –32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

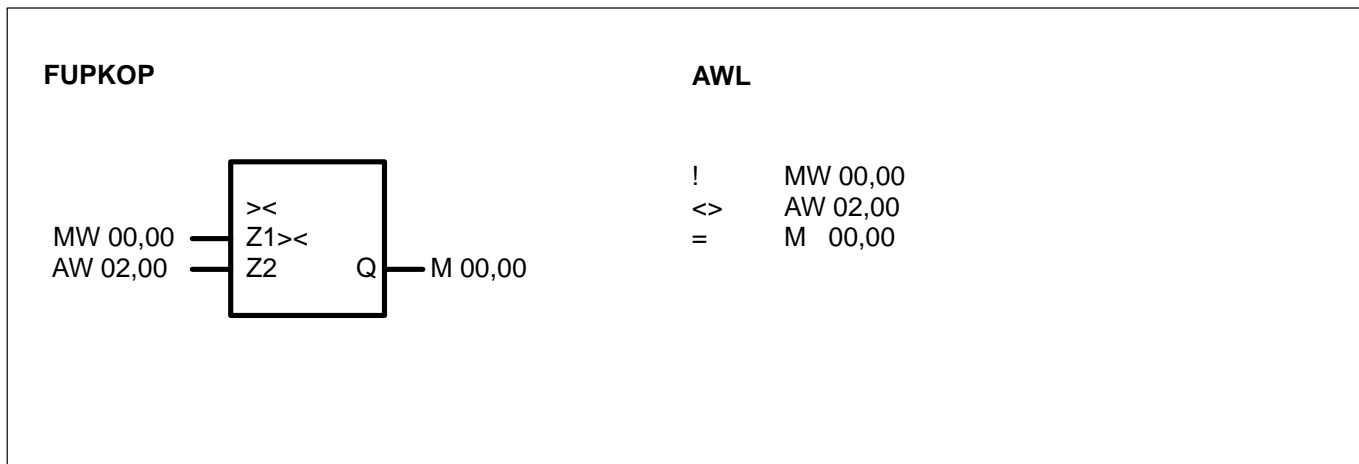
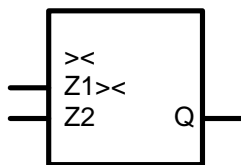
Der Wert 8000_H (–32768) liegt bei der Zweierkomplement–Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (–32768) auf den erlaubten Wert 8001_H (–32767) korrigiert.

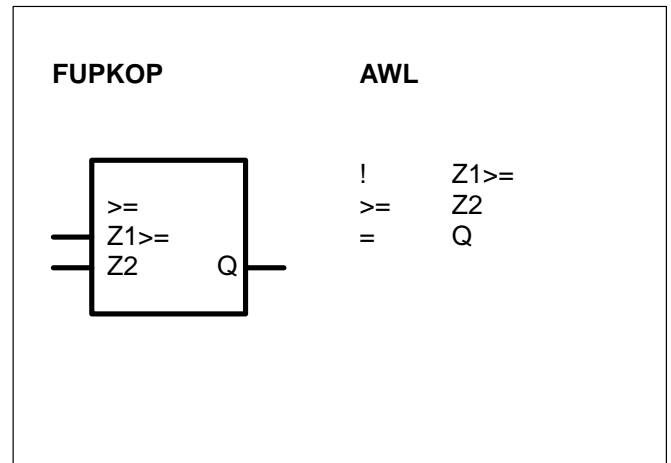
Beispiel**VE-FUP-Definition**

Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
Z1><	E	W	J	P	J	0	0
Z2	E	W	J	P	J	0	0
Q	A	L	J	P	J	0	0

VE-AWL-Definition

00000	!	PP 0	Z1><	Eingang WORT
00002	<>	PP 0	Z2	Eingang WORT
00004	=	PP 0	Q	Ausgang BINÄR

Der Wert des Operanden am Eingang Z1 \geq wird mit dem Wert des Operanden am Eingang Z2 verglichen. Ist der Wert an Z1 \geq größer als oder gleich dem Wert an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1 \geq kleiner Z2 wird Q der Zustand 0 zugewiesen.



Parameter

Z1 \geq	WORT	EW, MW, AW, KW	zu vergleichender Wert
Z2	WORT	EW, MW, AW, KW	Vergleichswert
Q	BINÄR	A, M, S	Ergebnis des Vergleichs

VE-Daten

Laufzeit:		
Grundlaufzeit:		< 12 μ s
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang Z1 \geq wird mit dem Wert des Operanden am Eingang Z2 verglichen. Ist der Wert an Z1 \geq größer als oder gleich dem Wert an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1 \geq kleiner Z2 wird Q der Zustand 0 zugewiesen.

Die Eingänge sind negierbar und nicht doppelbar. Der Ausgang ist invertierbar und nicht doppelbar.

Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für die nicht negierten Eingänge:

- untere Grenze 8000_H -32768
- obere Grenze 7FFF_H +32767

Allgemein gilt:

- untere Grenze: 8001_H -32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (-32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

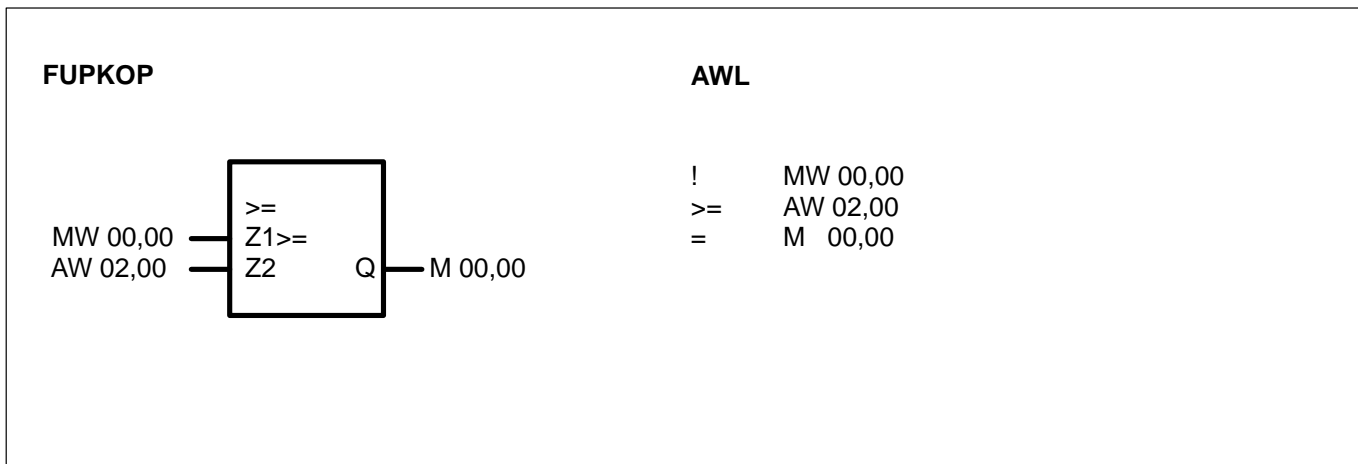
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

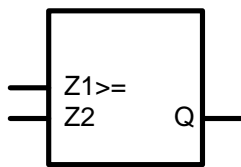
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (-32768) auf den erlaubten Wert 8001_H (-32767) korrigiert.

Beispiel



VE-FUP-Definition



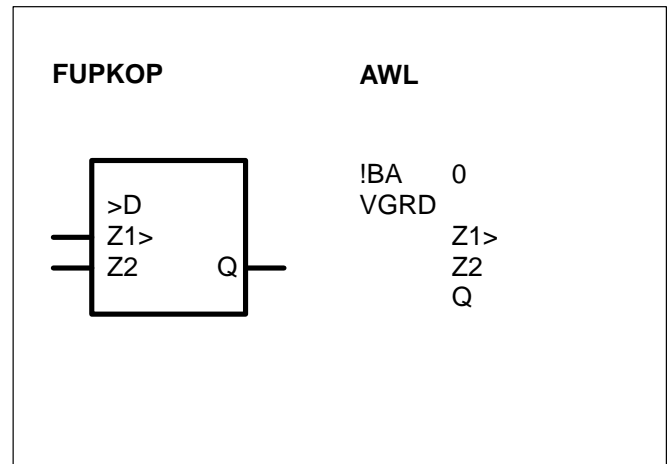
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
Z1>=	E	W	J	P	J	0	0
Z2	E	W	J	P	J	0	0
Q	A	L	J	P	J	0	0

VE-AWL-Definition

00000	!	PP 0	Z1>=	Eingang WORT
00002	>=	PP 0	Z2	Eingang WORT
00004	=	PP 0	Q	Ausgang BINÄR

Der Wert des Operanden am Eingang Z1> wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1> größer als an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1> gleich oder kleiner Z2, wird Q der Zustand 0 zugewiesen.



Parameter

Z1>	DOPPELWORT MD, KD	zu vergleichender Wert
Z2	DOPPELWORT MD, KD	Vergleichswert
Q	BINÄR A, M	Ergebnis des Vergleichs

VE-Daten

Laufzeit:	
Grundlaufzeit:	35 – 37 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang Z1> wird mit dem Wert des Operanden am Eingang Z2 verglichen.

Ist der Wert an Z1> größer als an Z2, wird dem Operanden am Ausgang Q der Zustand 1 zugewiesen. Ist Z1> gleich oder kleiner Z2, wird Q der Zustand 0 zugewiesen.

Die Eingänge sind weder doppelbar noch negierbar. Der Ausgang ist weder doppelbar noch invertierbar.

Zahlenbereich

Integer Doppelwort (32 Bit)

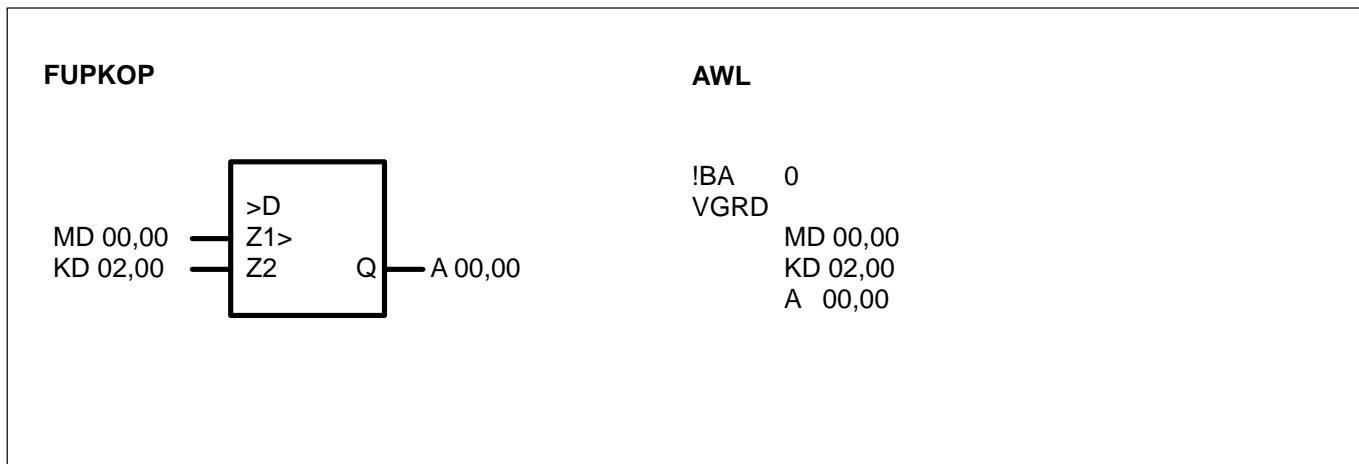
Es gilt hier speziell für die Eingänge Z1> und Z2:

- untere Grenze: 8000 0000_H –2 147 483 648
- obere Grenze: 7FFF FFFF_H +2 147 483 647

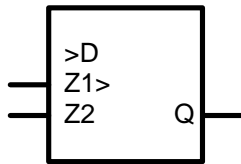
Allgemein gilt:

- untere Grenze: 8000 000_H –2 147 483 647
- obere Grenze: 7FFF FFFF_H +2 147 483 647
- unzulässiger Wert: 8000 0000_H —

Beispiel



VE-FUP-Definition

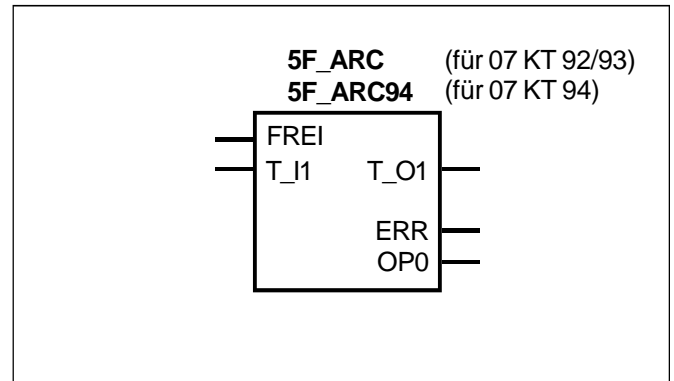


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
Z1>	E	D	N	P	J	0	0
Z2	E	D	N	P	J	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	VGRD			
00002		PP 0	Z1>	Eingang DOPPELWORT
00003		PP 0	Z2	Eingang DOPPELWORT
00004		PP 0	Q	Ausgang BINÄR

Der Funktionsbaustein koordiniert den Datentransfer zwischen dem PC und einer SPS über ARCNET.



Parameter

FREI	BIT	Freigabe
SEC	WORT	1. Nutzdaten Merker Empfang
T_O1	WORT	1. Nutzdaten Merker Empfang
ERR	WORT	Fehler
OP0		1. Operand im Operandenspeicher

Beschreibung

Um eine möglichst einfache, effektive Datenübertragung zu Visualisierungssystemen zu erhalten, wurde ein Protokoll entsprechend MODBUS realisiert.

Im SPS-Programm wird außer den ARCNET-Bausteinen AINIT, AREC, ASEND, ASEND+ und APOLL das Verknüpfungselement 5F_ARC (für 07 KT 92/93) oder 5F_ARC94 (für 07 KT 94 verwendet). Die beiden Verknüpfungselemente 5F_ARC und 5F_ARC94 sind in ihrer Funktion identisch.

Diese Verknüpfungselemente rufen über den Baustein CALLC ein C-Unterprogramm auf, das die empfangenen Daten interpretiert. Das Protokoll arbeitet nach einem Master/Slave-Prinzip. Der PC arbeitet immer als Master und die SPS immer als Slave. Welche Daten gesendet und empfangen werden sollen, wird im PC konfiguriert. In der SPS wird immer ein "Einheitsprogramm" verwendet.

Es ist möglich, mehrere Master mit einer SPS zu koppeln. In diesem Fall muß das SPS-Programm mit mehreren 5F_ARC oder 5F_ARC94 erweitert werden.

Zusätzlich zu diesem Protokoll kann auch eine Kommunikation zwischen den SPS entsprechend der Bausteinbeschreibungen ASEND, ASEND+ und AREC realisiert werden.

Im SPS-Programm muß für das Senden und Empfangen ein Bereich von jeweils 125 Merkerworten freigehalten werden. Die Telegramme bestehen aus einem ARCNET-Kopf (CP, DIN, JOB_NR) und dem MODBUS-Telegramm, das maximal diese 125 Worte ausfüllen darf. Alle Telegramme werden mit der DIN-Kennung 5F und der Nutzdatenlänge (AREC/ASEND) 125 (maximale Länge) gesendet. Als Jobnummer wird immer die ARCNET-Adresse des entsprechenden Teilnehmers verwendet. Diese Job-Nr. wird an den ASEND bzw. ASEND+ Bausteinen und am AREC eingetragen.

Beispiel: Wenn der Master (PC) die ARCNET-Adresse 1 hat und die SPS die ARCNET-Adresse 2, muß im SPS-Programm am AREC Jobnummer = 1 verwendet werden und am ASEND Jobnummer = 1.

Aufbau des ARCNET-Telegramms (siehe auch Baustein-Doku AREC)

Sender	Receiver	CP	DIN	MASTER Low	MASTER High	Zähler	Zähler	SLAVE	FCT	Nutzdaten
		03	5F	xx	00	aa	aa	yy	zz	
0	1	2	3	4	5	6	7	8	9	10-256

xx = ARCNET-Adresse des Masters = JOB-Nr. am AREC/ASEND-Baustein

yy = ARCNET-Adresse des Slave, ab hier beginnt das MODBUS-Telegramm

aa = Zähler, wird vom Master aktualisiert und vom Slave gespiegelt, um Telegrammverdopplung zu vermeiden. Dieser Zähler wird nicht für die Checksumme verwendet.

FCT = Funktionscode

Der Zähler ist das 1. Wort im empfangenen Telegramm. Dieses Wort muß auf das 1. Wort im Antworttelegramm zugewiesen werden.

Mit dem 2. Wort fängt das eigentliche Nutzdatentelegramm an. Dieses Wort muß am Baustein 5F_ARC an T_I1 angelegt werden. Der Baustein interpretiert das Telegramm und gibt eine Antwort an T_O1 (2. Wort im Antworttelegramm) aus. Wenn das Telegramm ein "Schreibbefehl" war, werden die entsprechenden Bit-, Wort- oder Doppelwortmerker vom Baustein überschrieben. Wenn das Telegramm ein "Lesebefehl" war, wird der Inhalt der zu lesenden Merker ins Antworttelegramm eingetragen. Falls das empfangene Telegramm fehlerhaft war, wird an ERR die Fehlernummer ausgegeben, zusätzlich ist ab T_O1 ein Fehlertelegramm eingetragen. In jedem Fall enthalten die Merker ab T_O1 nach einem Durchlauf des Bausteins die richtige Antwort, die dann über ASEND oder ASEND+ an den Master gesendet werden muß.

Die Freigabe FREI am Baustein 5F_ARC darf nur gesetzt werden, wenn vom entsprechenden Master ein Telegramm am AREC mit der DIN-Kennung 5Fempfungen wurde.

Am Ausgang OP0 muß der 1. Operand im Operandenspeicher (KW00,00) angelegt werden. Damit wird die Adresse des Operandenspeichers identifiziert.

FCT 1: Lesen n Bits
FCT 3: Lesen n Worte
FCT F: Schreiben n Bits
FCT 10: Schreiben n Worte

Zu übertragende Bits werden gepackt. Bit- und Wortübertragungen werden über den Funktionscode unterschieden.

Achtung:

High- und Low-Byte werden getauscht, wie im MODBUS-Telegramm. Es wird die MODBUS-CRC-Checksumme mitgeschickt. Da die Master identifiziert werden können, sind mehrere Master möglich. Die Antwort auf eine Anfrage wird immer an den richtigen Master zurückgeschickt.

Zusätzlich zu diesem Protokoll können über andere DIN-Kennungen auch benutzerdefinierte Telegramme in der üblichen Weise programmiert werden.

Adressen

Die Adressen werden folgendermaßen definiert (siehe auch Dokumentation 07 KR 31):

Binäre Operanden

Schrittketten dürfen nur gelesen werden.

Adresse	Operand	
0000 _H	E00,00	binäre Eingänge
0001 _H	E00,01	
0002 _H	E00,02	
03FF _H	E63,15	binäre Ausgänge
1000 _H	A00,00	
1001 _H	A00,01	
1002 _H	A00,02	
13FF _H	A63,15	
2000 _H	M000,00	binäre Merker
2001 _H	M000,01	
2002 _H	M000,02	
2FFF _H	M255,15	Schrittketten
3000 _H	S00,00	
3001 _H	S00,01	
3002 _H	S00,02	
37FF _H	S127,15	

Wort- Operanden:

Indirekte Konstanten werden nur im Operandenbereich überschrieben, d. h. bei Neustart des SPS-Programms wird auf den ursprünglich eingestellten Wert zurückgeschaltet.

Adresse	Operand	
0000 _H	EW00,00	Wort-Eingänge
0001 _H	EW00,0	
0002 _H	EW00,02	
007F _H	EW07,15	Wort-Ausgänge
1000 _H	AW00,00	
1001 _H	AW00,01	
1002 _H	AW00,02	
107F _H	AW07,15	
2000 _H	MW000,00	Wort-Merker
2001 _H	MW000,01	
2002 _H	MW000,02	
2FFF _H	MW255,15	indirekte Konstanten
3000 _H	KW00,00	
3001 _H	KW00,01	
3002 _H	KW00,02	
327F _H	KW39,15	
4000 _H	MD00,00	Doppelwort-Merker
41FF _H	MD31,15	
5000 _H	KD00,00	Doppelwort-Konstante
507F _H	KD7,15	

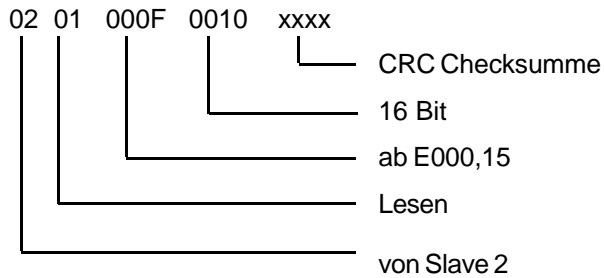
Aufbau des Telegramms anhand von Beispielen

Das ARCNET-Telegramm ist entsprechend der Beschreibung der Bausteine ASEND/AREC aufgebaut, mit DIN-Kennung 5F_H und Nutzdatenlänge 125 Worte.

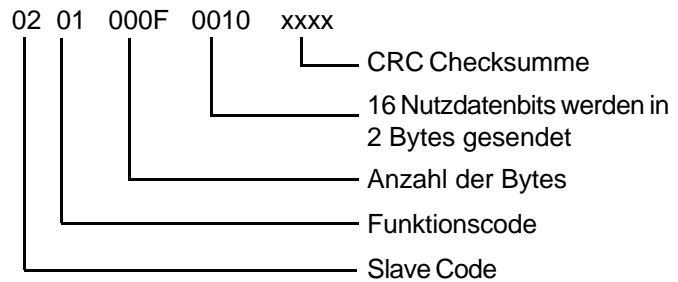
Das Nutzdatenfeld setzt sich folgendermaßen zusammen:

Beispiel: Lesen von n Bits

Anfrage:

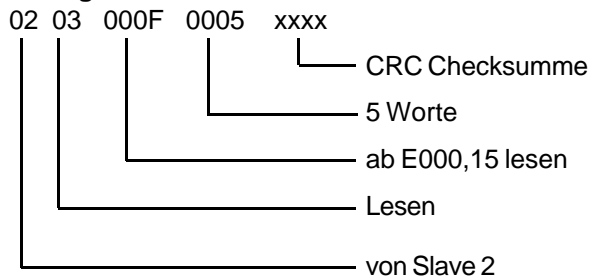


Antwort:

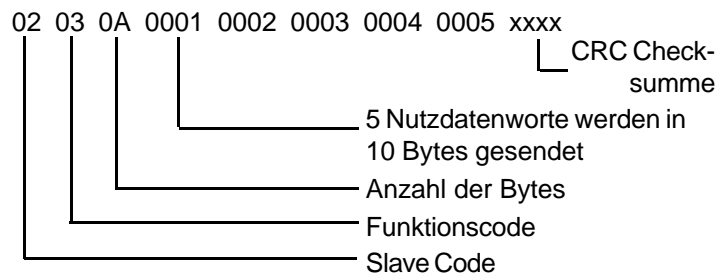


Beispiel: Lesen von n Worten

Anfrage:

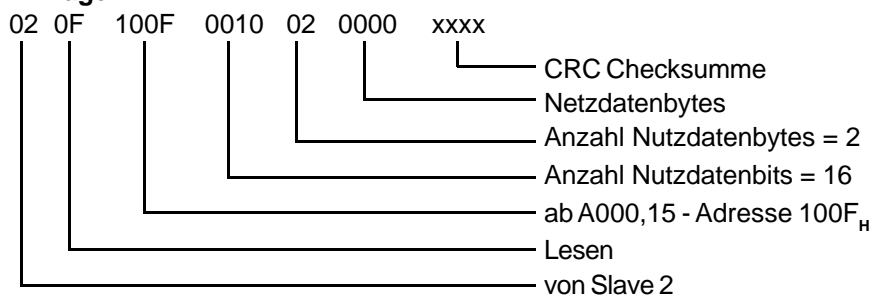


Antwort:

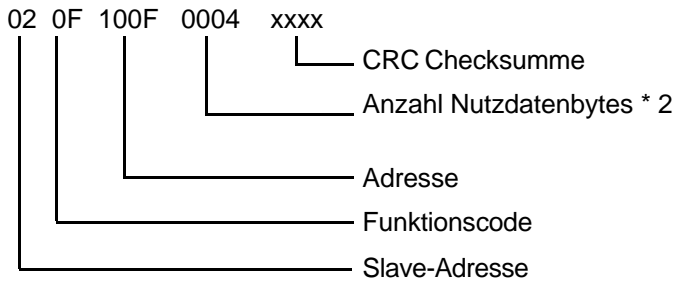


Beispiel: Schreiben von n Bits

Anfrage:

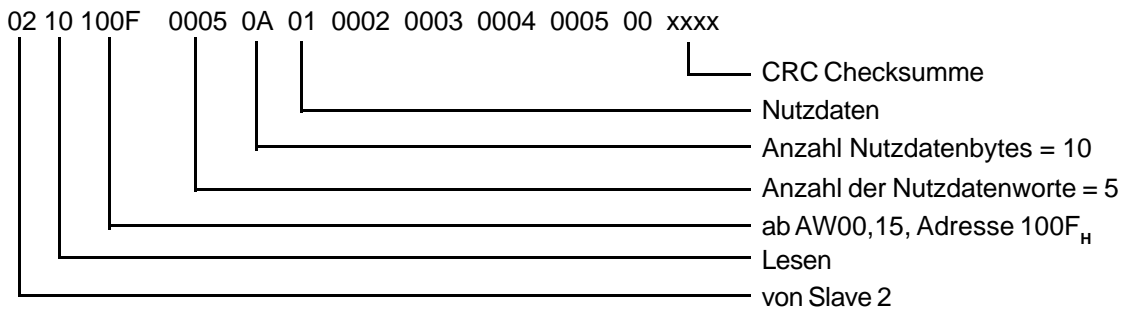


Antwort:

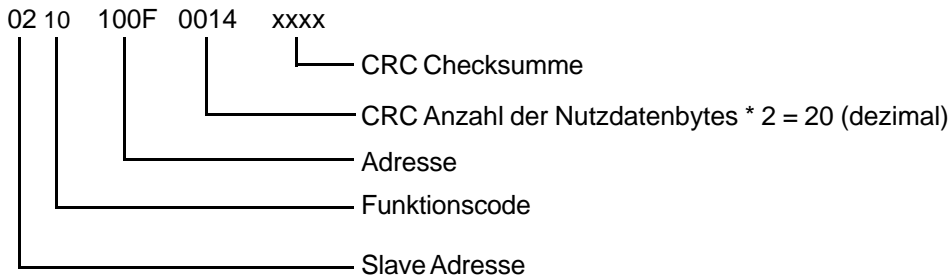


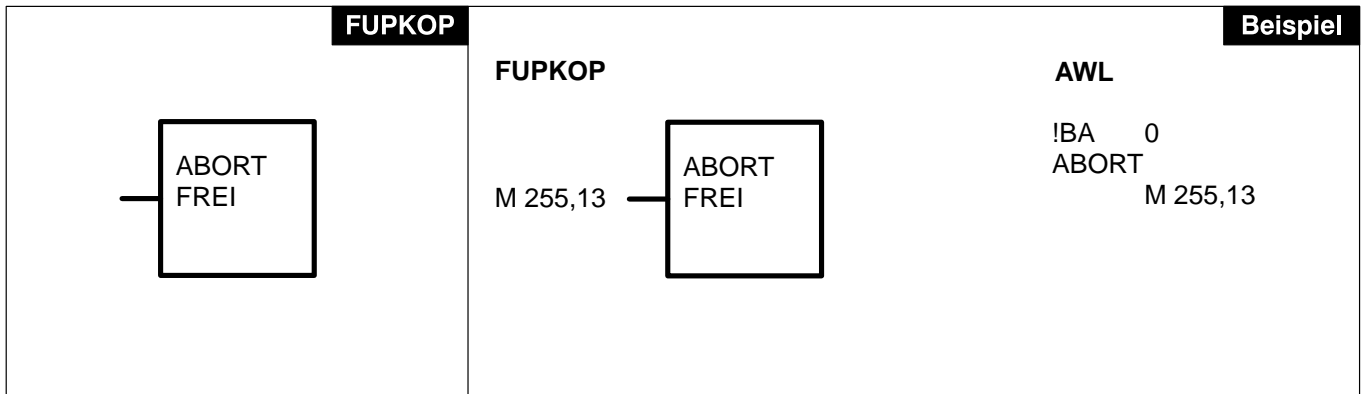
Beispiel: Schreiben von n Worten

Schreiben:



Antwort:





Parameter

FREI	BINÄR	E, M, A, K	Baustein-Freigabe
------	-------	------------	-------------------

Beschreibung

Der Funktionsbaustein dient zum Abbrechen eines SPS-Programmes. Beim Abbruch des SPS-Programmes werden alle Ausgänge auf den Wert 0 gesetzt.

Der Baustein kann insbesondere dazu benutzt werden, das SPS-Programm abzubrechen, wenn ein bestimmter Fehler der Fehlerklasse 3 auftritt.

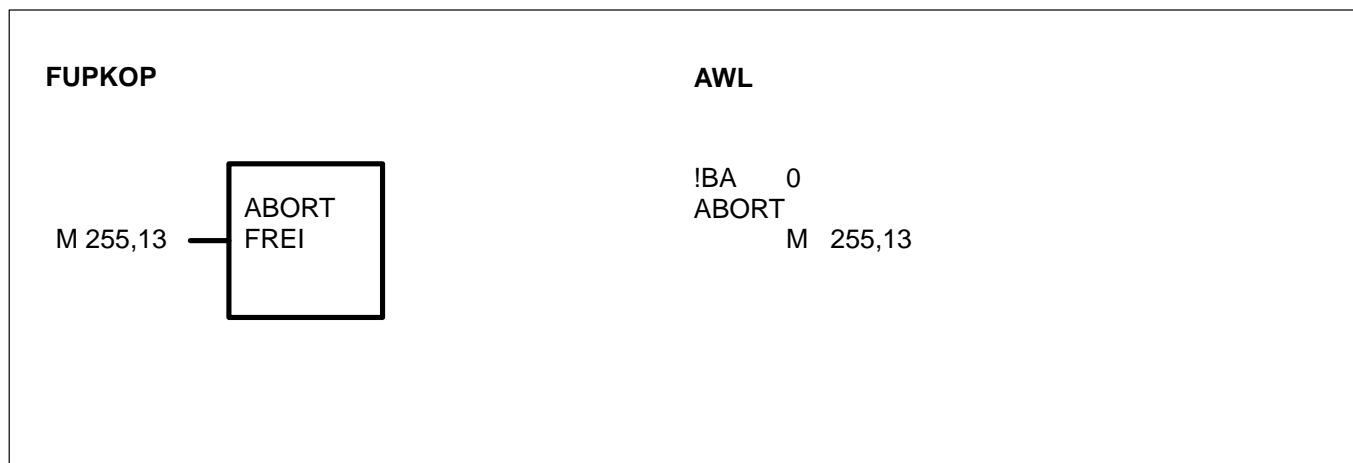
Der Funktionsbaustein wird durch ein 1-Signal an seinem Freigabeeingang aktiviert. Beim Anliegen eines 0-Signals am Freigabeeingang ist der Funktionsbaustein wirkungslos. es gilt also:

FREI = 0: —> der Baustein ist wirkungslos
 FREI = 1: —> der Baustein bricht das Programm ab

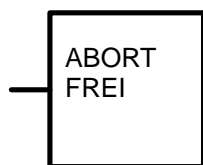
VE-Daten

Laufzeit:	
Grundlaufzeit:	10 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	—
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / Kommunikationsprozessor 07 KP 62

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt)
00001	ABORT			
00002		PP 0	FREI	Baustein-Freigabe (BINÄR)

Der Adaptionbaustein dient zur Ermittlung optimierter Reglerparameter für den Regler-Baustein PIDT1. Einsetzbar ist dieser Baustein bei Temperaturregelstrecken in der Verfahrenstechnik und der kunststoffverarbeitenden Industrie.

Der Adaptionbaustein dient nicht zur kontinuierlichen Ermittlung der Reglerparameter sondern ermittelt diese nur im Bedarfsfall (Knopfdruckadaption).

FUPKOP	AWL
nicht definiert *)	!BA 0 ADAPT W X YR START READY ERROR Y INIT KP TN/TZ TV/TZ T1/TZ D-FREI KS TU TG

*) Hinweis: Für den Adaptionbaustein kann ein VE definiert werden. Siehe 907 PC 33 Beschreibung, Allgemeiner Teil, Kap. 15 (Bibliothek).

Parameter

W	WORT	EW, AW, MW, KW	Führungsgröße (Sollwert)
X	WORT	EW, AW, MW, KW	Regelgröße (Istwert)
YR	WORT	EW, AW, MW, KW	Stellgröße des Reglers beim Startzeitpunkt der Adaption
START	BINÄR	E, A, M, S	Start des Adaptionsvorgangs
READY	BINÄR	E, A, M	Fertigmeldung für Adaptionsvorgang
ERROR	WORT	AW, MW	Störmeldungen
Y	BINÄR	A, M	Anregungssignal (Stellgröße) während des Adaptionsvorgangs
INIT	WORT	AW, MW	Initialwert für den Regler
KP	WORT	AW, MW	Proportional-Beiwert
TN/TZ	WORT	AW, MW	Nachstellzeit, normiert auf SPS-Zykluszeit TZ
TV/TZ	WORT	AW, MW	Vorhaltezeit, normiert auf SPS-Zykluszeit TZ
T1/TZ	WORT	AW, MW	Rücklaufzeit, normiert auf SPS-Zykluszeit TZ
D-FREI	BINÄR	A, M	Freigabe für D-Anteil des Reglers
KS	WORT	AW, MW	Verstärkung der Regelstrecke
TU	WORT	AW, MW	Verzugszeit der Regelstrecke
TG	WORT	AW, MW	Ausgleichszeit der Regelstrecke

VE-Daten

Laufzeit:	
Grundlaufzeit:	liegt nicht vor
Zusätzliche Laufzeit:	liegt nicht vor
Aktualisierung der Ausgänge:	nein; Empfehlung: Ausgang direkt an Merker zuweisen oder Zwischenmerker global vergeben; siehe Griff 3, Kapitel 2.5.3
Anzahl der Vergangenheitswerte:	70 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

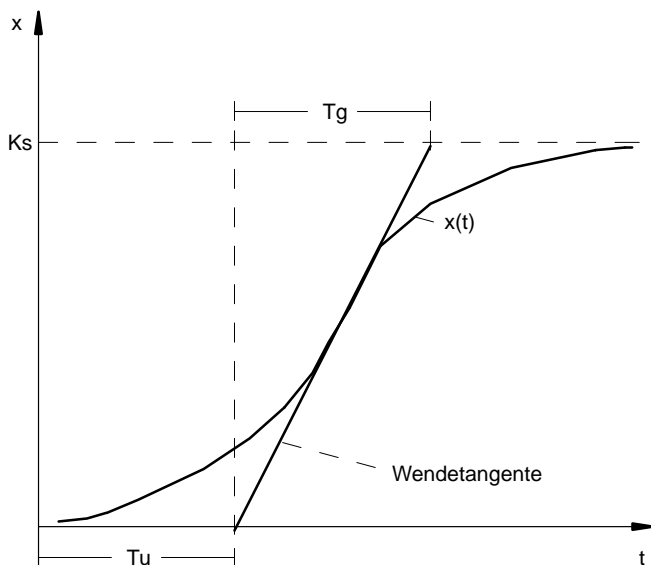
Beschreibung

Hinweis: Änderung gegenüber SPS-Version V6.0. Die Streckenparameter K_s , T_u und T_g stehen ab SPS-Version V7.0 als zusätzliche Ausgänge zur Verfügung.

Der Adaptionbaustein dient zur Ermittlung optimierter Reglerparameter für den Regler-Baustein PIDT1. Einsetzbar ist dieser Baustein bei Temperaturregelstrecken in der Verfahrenstechnik und der kunststoffverarbeitenden Industrie. Die Dynamik dieser thermischen Prozesse läßt sich durch die aus der Übergangsfunktion bestimm- baren Kenngrößen

T_u : Verzugszeit
 T_g : Ausgleichzeit
 K_s : Übertragungsfaktor (Streckenverstärkung)

ausreichend annähern. Dabei wird der tatsächliche Verlauf der Übergangsfunktion ersetzt durch die in ihrem Wendepunkt angelegte Tangente.



Das Bild zeigt die Übergangsfunktion, ihre Wendetangente und die daraus resultierenden Kenngrößen bei aperiodischen Prozessen.

Der Adaptionbaustein ist speziell für Prozesse ausgelegt, deren Kenngrößen die folgenden Bedingungen erfüllen:

$$\begin{aligned} 5 &\leq T_g/T_u \leq 100 \\ 20 \text{ sec} &\leq T_u \leq 300 \text{ sec} \\ 0.25 &\leq K_s \leq 2 \end{aligned}$$

Eine Adaption ist immer nur dann zulässig, wenn sich der Regelkreis im eingeschwungenen Zustand befindet. Die Adaption kann sowohl beim Anfahren des Prozesses (Ersteinstellung), als auch bei einer Führungsgrößenänderung durchgeführt werden. Dies ermöglicht eine Anpassung der Reglerparameter, falls der Prozeß in verschiedenen Arbeitspunkten gefahren werden muß und eine feste Einstellung der Reglerparameter nicht das gewünschte Regelverhalten liefert. Hierbei wird eine Stellgradänderung (Führungsgrößenänderung/Verstärkungsfaktor der Strecke) von mindestens 20 % vorausgesetzt. Die Identifikation der Strecke führt bei Stellgradänderungen unterhalb von 20 % zu ungenauen Reglerparametern, die ein größeres Überschwingen zur Folge haben.

Während der Adaption führt der Adaptionbaustein den Prozeß und der Regler ist abgekoppelt.

Start/Abbruch der Adaption

Durch eine 0/1-Flanke am Eingang START beginnt die Adaption. Durch eine 1/0-Flanke am Eingang START wird eine laufende Adaption abgebrochen.

Durchführung der Adaption

Während der Adaption liefert der Ausgang Y die binäre Stellgröße für die Regelstrecke. Der PIDT1-Regler ist inaktiv, d. h. es darf keine Beeinflussung der Strecke durch den Regler erfolgen. Dies muß durch eine Steuerlogik im SPS-Programm sichergestellt werden.

Zu Beginn der Adaption benötigt der Baustein die Anfangswerte der momentan vorhandenen Regel- und Stellgröße. Diese Anfangswerte sind nur bei der Ersteinstellung exakt bekannt (Regelgröße = Umgebungstemperatur, Stellgröße = 0).

Bei einer Adaption im geregelten Betrieb (Neueinstellung bei einer Sollwertänderung) wird der Anfangswert der Stellgröße durch den Stellausgang des PIDT1-Reglers bestimmt. Der PIDT1-Regler liefert aufgrund der projektierten Regelungsstruktur (PIDT1 mit nachgeschaltetem PDM-Baustein) keine stationäre Stellgröße. Es stellt sich eine Dauerschwingung der Reglerstellgröße ein, die eine exakte Messung des Gleichanteils verhindert. Es wird deshalb zur Bereitstellung des Stellgrößenanfangswertes für den Adaptionbaustein eine Tiefpaßfilterung der Reglerstellgröße empfohlen (PT1-Baustein, Zeitkonstante > 10 sec).

Durch die Dauerschwingung der Reglerstellgröße ergibt sich auch eine Dauerschwingung der Regelgröße. Diese Schwingung ist aber durch das Tiefpaßverhalten der Regelstrecke sehr stark gedämpft. Eine zusätzliche Filterung der Regelgröße ist deshalb nicht erforderlich.

Überwachung der Adaption

Der Adaptionbaustein beinhaltet eine Überwachungsebene. Diese führt eine Plausibilitätsprüfung der identifizierten Streckenparameter und der Eingangsgrößen durch. Über den Ausgang ERROR werden der korrekte Abschluß oder während der Adaption erkannte Unregelmäßigkeiten gemeldet. Hierbei wird zwischen Fehlern und Warnungen unterschieden.

Fehler:

Diese führen immer zu einem vorzeitigen Abbruch der Adaption. Es werden keine Reglerparameter berechnet und der Stellausgang Y wird auf "0" gesetzt. Das Signal READY bleibt auf "0" gesetzt.

Warnungen:

Diese zeigen an, daß die errechneten Streckenparameter außerhalb des zulässigen Wertebereichs liegen. Die Adaption wird aber regulär beendet (READY = 1). Ein befriedigendes Regelverhalten kann aber nicht garantiert werden. Das Regelverhalten muß bei der Inbetriebnahme auf Tauglichkeit überprüft werden.

- ERROR = 0: Kein Fehler, keine Warnung.

- ERROR = 1: Fehler

Die Regelgrößendifferenz ist beim Start der Adaption kleiner als der Minimalwert MINXD = 3843. Der Adaptionvorgang wird abgebrochen.

Abhilfe: Anfangsregeldifferenz vergrößern.

- ERROR = 2: Fehler

Für die Berechnung der Wendetangente standen weniger als 16 Stützstellen zur Verfügung. Der Adaptionvorgang wird abgebrochen.

Abhilfe: Zykluszeit verkleinern.

- ERROR = 3: Fehler

Die Verzugszeit TU liegt außerhalb des zulässigen Wertebereichs ($TU \leq 0$). Der Adaptionvorgang wird abgebrochen.

Abhilfe: ABB befragen.

- ERROR = -1: Warnung

Die Stellgradänderung liegt unterhalb des erforderlichen Minimalwertes. Die Adaption wird regulär abgeschlossen. Ein befriedigendes Regelverhalten kann aber nicht garantiert werden.

Abhilfe: Bei Inbetriebnahme Regelverhalten prüfen. Falls unbefriedigend ABB befragen.

- ERROR = -2: Warnung

Das Verhältnis TU/TG ist größer als der zulässige Maximalwert. Die Adaption wird regulär abgeschlossen. Ein befriedigendes Regelverhalten kann aber nicht garantiert werden.

Abhilfe: Bei Inbetriebnahme Regelverhalten prüfen. Falls unbefriedigend ABB befragen.

Abschluß der Adaption

READY = 1 zeigt den regulären Abschluß der Adaption an.

An den Bausteinausgängen stehen dann die Reglerparameter

- INIT
- KP
- TN/TZ
- TV/TZ
- T1/TZ
- D-FREI

und die Regelstreckenparameter

- KS
- TU
- TG

zur Verfügung.

Die Regelstreckenparameter KS, TU und TG sind nur für den Spezialisten von Bedeutung und brauchen vom Anwender normalerweise nicht ausgewertet werden.

Der Ausgang D-FREI zeigt an, ob ein PI- oder ein PIDT1-Regler erforderlich ist.

Es gilt:

D-FREI	Bedeutung
0	DT1-Anteil ist abgeschaltet -> PI-Regler
1	DT1-Anteil ist zugeschaltet -> PIDT1-Regler

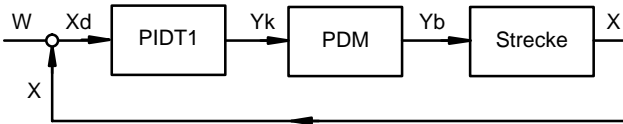
Am Ausgang INIT steht ein Abgleichwert für das Umschalten vom Adaptionbaustein auf den Regler zur Verfügung.

Der Ausgang Y wird nach Abschluß der Adaption auf "0" gesetzt.

PROJEKTIERUNG

Temperaturregelung mit schaltenden Stellgliedern

Für die Temperaturregelung mit schaltenden Stellgliedern wird bei ABB Procontic T300 die folgende Reglerstruktur eingesetzt:



W: Führungsgröße (Sollwert)

X: Regelgröße (Istwert)

Xd: Regeldifferenz

Yk: kontinuierliche Stellgröße

Yb: binäre Stellgröße

Durch den Pulsdauermodulator (PDM) tritt gegenüber einem kontinuierlichen Stellglied eine Restpendelung der Regelgröße im Arbeitspunkt auf. Diese Restpendelung wird durch das Zyklusverhältnis TA/TZ des PDM-Bausteins bestimmt. Dabei ist TA die Periodendauer und TZ die Zykluszeit mit der der PDM-Baustein bearbeitet wird.

Einstellempfehlung:

$TZ(\text{PDM}) = 100$ ms: Zykluszeit des PDM
 $TA/TZ(\text{PDM}) = 5$: Zyklusverhältnis des PDM
 $TZ(\text{PIDT1}) = 500$ ms: Zykluszeit des PIDT1

Konfiguration von Regler und Adaptionbaustein

Bei der Projektierung sind einige elementare Punkte zu beachten:

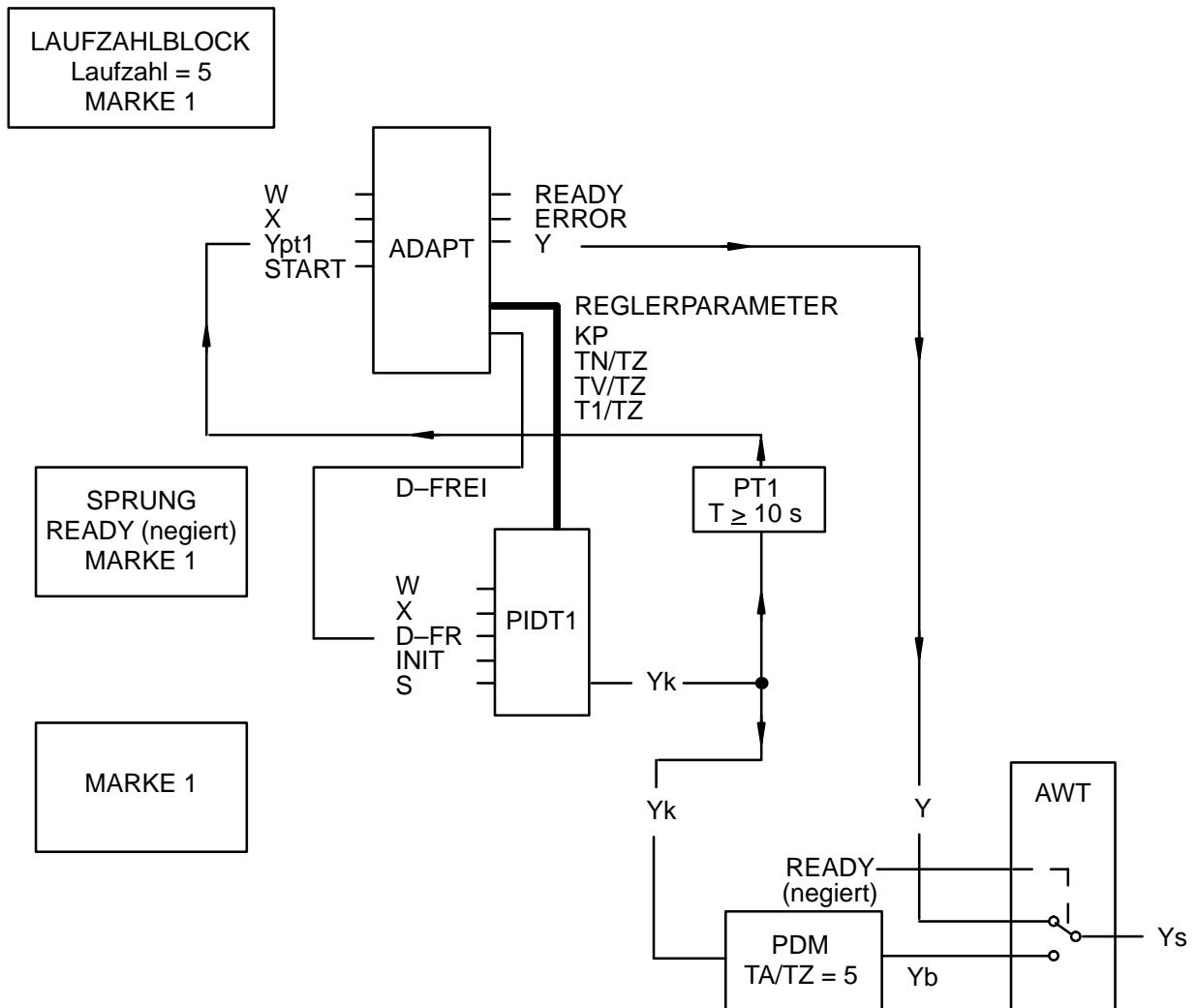
- Folgende Bausteine müssen mindestens mit einer 5 mal kleineren Zykluszeit als der Regler und Adaptionbaustein gefahren werden:
 - Pulsdauermodulator (PDM)
 - Auswahltor (AWT) für die Stellsignale von Regler und Adaptionbaustein

Dies wird dadurch erreicht, daß alle zur Adaptiven Regelung gehörenden Bausteine und Anweisungen mit Ausnahme des Pulsdauermodulators (PDM) und des Auswahltors (AWT) innerhalb eines Laufzahlblocks (LZB) mit der Laufzahl gleich 5 stehen und somit nur bei jedem fünften Programmzyklus bearbeitet werden.

- Bei aktivem Adaptionvorgang muß der PIDT1-Regler übersprungen werden. Dies wird dadurch erreicht, daß das negierte READY-Signal des Adaptionbausteins als Sprungbedingung genommen wird.

- Mit dem Auswahltor (AWT) wird entweder die Stellgröße des Adaption- oder des Regelungsbausteins ausgewählt. Damit ist sichergestellt, daß die Regelstrecke entweder nur vom Regler (Regelungsbetrieb) oder nur vom Adaptionbaustein (Adaptionbetrieb) gefahren wird. Das Auswahltor wird mit dem READY-Signal des Adaptionbausteins geschaltet.
- Beim Beginn des Adaptionvorgangs wird die aktuelle Stellgröße des Reglers gebraucht. Da diese im geregelten Betrieb einer Dauerschwingung unterliegt, ist sie über ein PT1-Glied dem Adaptionbaustein zuzuführen. Die Zeitkonstante des PT1-Glieds muß mindestens 10 sec betragen.
- Der vom Adaptionbaustein gelieferte Abgleichwert (INIT) kann wie folgt dem Regler übergeben werden:
 - Mit der 0/1-Flanke des READY-Signals des Adaptionbausteins wird ein Binärmerker für *einen* Zyklus auf den Wert "1" gesetzt. Dieser Binärmerker wird als Setzbedingung an den Eingang S des Reglers angelegt. Zweckmäßigerweise ist während des Setzvorgangs der D-Anteil des Reglers zu sperren (siehe dazu auch Baustein PIDT1, "Setzen und Rücksetzen des Reglers").

Projektierungsübersicht von Regler und Adaptionsbaustein



W : Führungsgröße
 X : Regelgröße
 Y_{pt1} : gefiltertes Y_k
 Y_b : binäre Stellgröße im Regelungsbetrieb
 Y : Stellgröße während der Adaption

Y_k : kontinuierliche Stellgröße
 Y_s : Stellgröße für Strecke
 READY: ADAPT-Fertigmeldung
 START: Start der Adaption
 D-FREI: Freigabe DT1-Anteil für Regler PIDT1

W WORT

Führungsgröße (Sollwert).

X WORT

Regelgröße (Istwert).

YR WORT

Aktuelle Stellgröße des Reglers beim Start der Adaption.

START BINÄR

Starteingang.

0/1–Flanke: Start der Adaption.

1/0–Flanke: Abbruch der Adaption.

READY BINÄR

Fertigmeldung.

1–Signal meldet den regulären Abschluß der Adaption.

ERROR WORT

Störmeldungen.

● **ERROR = 1:** Fehler

Die Regelgrößendifferenz ist beim Start der Adaption kleiner als der Minimalwert $MINXD = 3843$. Der Adaptionsvorgang wird abgebrochen.

Abhilfe: Anfangsregeldifferenz vergrößern.

● **ERROR = 2:** Fehler

Für die Berechnung der Wendetangente standen weniger als 16 Stützstellen zur Verfügung. Der Adaptionsvorgang wird abgebrochen.

Abhilfe: Zykluszeit verkleinern.

● **ERROR = 3:** Fehler

Die Verzugszeit TU liegt außerhalb des zulässigen Wertebereichs ($TU \leq 0$). Der Adaptionsvorgang wird abgebrochen.

Abhilfe: ABB befragen.

● **ERROR = -1:** Warnung

Die Stellgradänderung liegt unterhalb des erforderlichen Minimalwertes. Die Adaption wird regulär abgeschlossen. Ein befriedigendes Regelverhalten kann aber nicht garantiert werden.

Abhilfe: Bei Inbetriebnahme Regelverhalten prüfen. Falls unbefriedigend ABB befragen.

● **ERROR = -2:** Warnung

Das Verhältnis TU/TG ist größer als der zulässige Maximalwert. Die Adaption wird regulär abgeschlossen. Ein befriedigendes Regelverhalten kann aber nicht garantiert werden.

Abhilfe: Bei Inbetriebnahme Regelverhalten prüfen. Falls unbefriedigend ABB befragen.

Y BINÄR

Anregungssignal (Stellgröße) für die Regelstrecke während der Adaption.

INIT WORT

Initialwert für den Regler nach Abschluß der Adaption.

KP WORT

Proportional–Beiwert; Ausgabe erfolgt in Prozent.

TN/TZ WORT

Nachstellzeit, normiert auf die SPS–Zykluszeit TZ .

TV/TZ WORT

Vorhaltezeit, normiert auf die SPS–Zykluszeit TZ .

T1/TZ WORT

Rücklaufzeit, normiert auf die SPS–Zykluszeit TZ .

D–FREI BINÄR

Freigabe für D–Anteil des Reglers. Mit diesem Ausgang wird entschieden, ob ein PI– oder ein PIDT1–Regler erforderlich ist.

D–FREI = 0 → PI–Regler

D–FREI = 1 → PIDT1–Regler

KS WORT

Verstärkung der Regelstrecke.

TU WORT

Verzugszeit der Regelstrecke.

TG WORT

Ausgleichszeit der Regelstrecke.

Beispiel

FUPKOP	AWL
nicht definiert *)	!BA 0 ADAPT KW 01,00 EW 00,03 MW 05,11 E 00,08 M 04,07 MW 05,12 M 04,08 MW 06,00 MW 06,01 MW 06,02 MW 06,03 MW 06,04 M 04,04 MW 06,05 MW 06,06 MW 06,07

VE-FUP-Definition

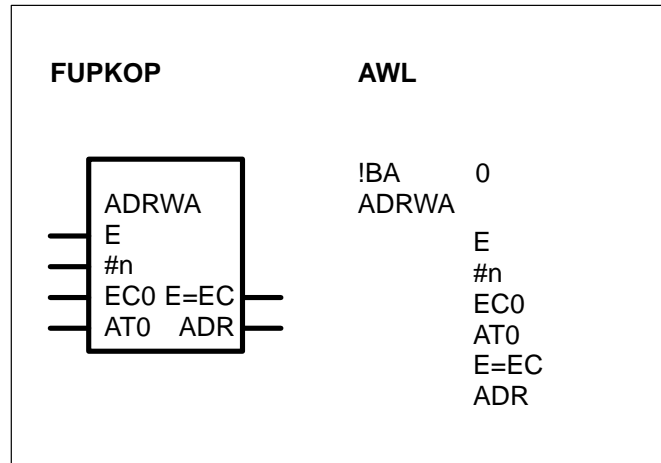
nicht definiert *)

VE-AWL-Definition

nicht definiert *)

*) Hinweis: Für den Adaptionbaustein kann ein VE definiert werden. Siehe 907 PC 32 Beschreibung, Allgemeiner Teil, Kap. 15 (Bibliothek / VE-Editor / VE-Anweisungseditor).

Mit dem Funktionsbaustein wird einer der an den Eingängen AT0 ... AT_{n-1} projektierten Operanden ausgewählt. Von diesem ausgewählten Operanden wird die indirekte Adresse gebildet und am Ausgang ADR zur Verfügung gestellt.



Parameter

E	WORT	EW, AW, MW, KW	Eingangswert
#n	DIREKTE KONSTANTE	#, #H	Anzahl EC bzw. AT
EC0	WORT	EW, AW, MW, KW	Eingangscode; Eingang ist doppelbar
AT0	WORT	EW, AW, MW, KW	Ausgabetabelle, Eingang wird mit EC0 mitgedoppelt
E=EC	BINÄR	A, M	Eingangswert=Eingangscode
ADR	WORT	AW, MW	indirekte Adresse

VE-Daten

Laufzeit:	
Grundlaufzeit:	44 µs
Zusätzliche Laufzeit:	8 µs pro Operand EC0..EC _{n-1}
Aktualisierung der Ausgänge:	nein; Empfehlung: Ausgang direkt an Merker zuweisen oder Zwischenmerker global vergeben; siehe Griff 3, Kapitel 2.5.3
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Mit dem Funktionsbaustein wird einer der an den Eingängen AT0 ... AT_{n-1} projektierten Operanden ausgewählt. Von diesem ausgewählten Operanden wird die indirekte Adresse gebildet und am Ausgang ADR zur Verfügung gestellt.

Die Ein- und Ausgänge sind nicht negierbar/invertierbar. Der Eingang EC0 ist doppelbar, wobei der Eingang AT0 automatisch mitgedoppelt wird.

Indirektes Lesen/Schreiben von Operanden

Die Funktionsbausteine AWM bzw. USM benutzen die vom Baustein ADRWA erzeugte indirekte Adresse, um den beim Baustein ADRWA ausgewählten Operanden zu lesen bzw. zu schreiben.

Zum indirekten Lesen bzw. Schreiben von Operanden benötigt man also den Baustein ADRWA und den Baustein AWM bzw. USM. Dabei werden beim Baustein ADRWA an den Eingängen AT0 ... AT_{n-1} die zu lesenden bzw. zu schreibenden Operanden aufgelistet und durch

den Baustein AWM bzw. USM wird dann der Lese- bzw. Schreibzugriff durchgeführt.

Vorteil der indirekten Adressierung:

- Bei geeigneten Anwendungen ergibt sich eine große Vereinfachung des SPS-Programms und damit weniger Projektierungsaufwand.
- Mit nur *einem* Baustein (AWM bzw. USM) erfolgt der Zugriff auf beliebig viele Operanden (Multiplex-Funktion). Der Baustein ADRWA stellt dabei ein mächtiges Werkzeug dar, mit dem auf sehr flexible Art und Weise die Operanden ausgewählt werden können, auf die ein Zugriff erfolgen soll.

Auswahl eines Operanden aus der Ausgabetabelle AT0 ... AT_{n-1}

Der Baustein vergleicht den Wert am Eingang E der Reihe nach mit den Werten an den Eingängen EC0 ... EC_{n-1}. Der Vergleich beginnt bei jedem Aufruf des Bausteins wieder von neuem, d.h. mit dem Eingang EC0.

- Bei Übereinstimmung des Wertes am Eingang E mit einem der Werte an den Eingängen EC0 ... EC_{n-1} wird:
 - der Ausgang E=EC auf 1 gesetzt (Treffer).
 - der *zugeordnete* Operand aus der Ausgabetabelle AT0 ... AT_{n-1} ausgewählt.
- Ist keine Übereinstimmung des Wertes am Eingang E mit einem der Werte an den Eingängen EC0...EC_{n-1} vorhanden, so wird:
 - der Ausgang E=EC auf 0 gesetzt (kein Treffer)
 - es wird *kein* Operand aus der Ausgabetabelle AT0 ... AT_{n-1} ausgewählt.

Zuordnungsvorschrift zwischen EC0 ... EC_{n-1} und AT0 ... AT_{n-1}:

EC0 → AT0
 EC1 → AT1
 . . .
 EC_{n-1} → AT_{n-1}

Erzeugung der indirekten Adresse

Wurde ein Operand aus der Ausgabetabelle AT0 ... AT_{n-1} ausgewählt, so wird dessen indirekte Adresse erzeugt. Dies geschieht dadurch, daß die Adresse des ausgewählten Operanden genommen wird und dem am Ausgang ADR angegebenen Operanden als Wert zugewiesen wird. Der Wert des am Ausgang ADR angegebenen Operanden ist also die Adresse des aus der Ausgabetabelle AT0 ... AT_{n-1} ausgewählten Operanden.

Def.: Eine indirekte Adresse ist ein Operand, dessen Wert die Adresse eines anderen Operanden ist.

Benutzung einer indirekten Adresse

Die Funktionsbausteine AWM und USM greifen mit indirekter Adressierung auf Operanden zu. Die Bausteine AWM und USM benötigen daher an ihrem Eingang die vom Baustein ADRWA erzeugte indirekte Adresse.

E WORT

Mit Hilfe des am Eingang E und den an den Eingängen EC0 ... EC_{n-1} angegebenen Operanden wird einer der Operanden aus der Ausgabetabelle AT0 ... AT_{n-1} ausgewählt, und anschließend dessen indirekte Adresse erzeugt.

Der Baustein vergleicht dazu den Wert am Eingang E der Reihe nach mit den Werten an den Eingängen EC0 ... EC_{n-1}. Der Vergleich beginnt bei jedem Aufruf des Bausteins wieder von neuem, d.h. beginnend mit dem Eingang EC0.

- Bei Übereinstimmung des Wertes am Eingang E mit einem der Werte an den Eingängen EC0 ... EC_{n-1} wird:
 - der Ausgang E=EC auf 1 gesetzt (Treffer).
 - der *zugeordnete* Operand aus der Ausgabetabelle AT0 ... AT_{n-1} ausgewählt und dessen indirekte Adresse erzeugt.
- Ist keine Übereinstimmung des Wertes am Eingang E mit einem der Werte an den Eingängen EC0 ... EC_{n-1} vorhanden, so wird:
 - der Ausgang E=EC auf 0 gesetzt (kein Treffer)
 - es wird *kein* Operand aus der Ausgabetabelle AT0 ... AT_{n-1} ausgewählt, und es wird demzufolge auch keine indirekte Adresse erzeugt.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der projektierten Eingänge EC0 ... EC_{n-1} angegeben. Die Angabe erfolgt als direkte Konstante.

Bsp.: Projektiert sind: EC0, EC1, EC2 → #n = 3

EC0...EC_{n-1} WORT

An den Eingängen EC0 ... EC_{n-1} werden die Operanden für die Vergleichswerte angegeben. Der Eingang EC0 ist doppelbar. Der Wert am Eingang E wird mit diesen Vergleichswerten verglichen und auf Übereinstimmung geprüft. Bei Übereinstimmung wird der zugeordnete Operand aus der Ausgabetabelle AT0...AT_{n-1} ausgewählt und dessen indirekte Adresse erzeugt. Die Werte an den Eingängen EC0 ... EC_{n-1} sind variabel, da es sich dabei um normale Operanden handelt. Aus diesem Grunde ist die Vergleichsfunktion zwischen dem Wert am Eingang E und den Werten an den Eingängen EC0 ... EC_{n-1} sehr flexibel und mächtig.

Zuordnung zwischen EC0...EC_{n-1} und AT0...AT_{n-1}:

EC0 → AT0
 EC1 → AT1
 . . .
 EC_{n-1} → AT_{n-1}

AT0...AT_{n-1} WORT

An den Ausgängen AT0 ... AT_{n-1} werden die Operanden angegeben, dessen indirekte Adressen erzeugt werden sollen.

Bei der Dopplung des Eingangs EC0 wird der Eingang AT0 automatisch mitgedoppelt.

E=EC BINÄR

Am Ausgang E=EC wird angezeigt, ob der Wert am Eingang E mit einem der Werte an den Eingängen $EC_0 \dots EC_{n-1}$ übereinstimmt.

E=EC = 0 → keine Übereinstimmung

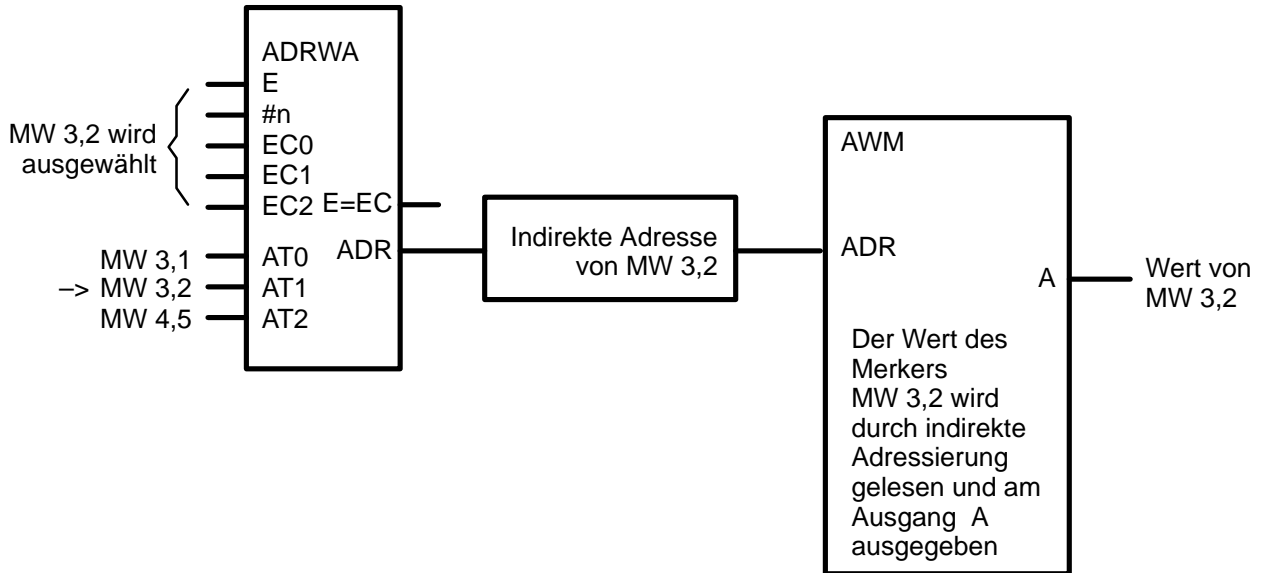
E=EC = 1 → der Wert am Eingang E stimmt überein mit einem der Werte an den Eingängen $EC_0 \dots EC_{n-1}$

ADR WORT

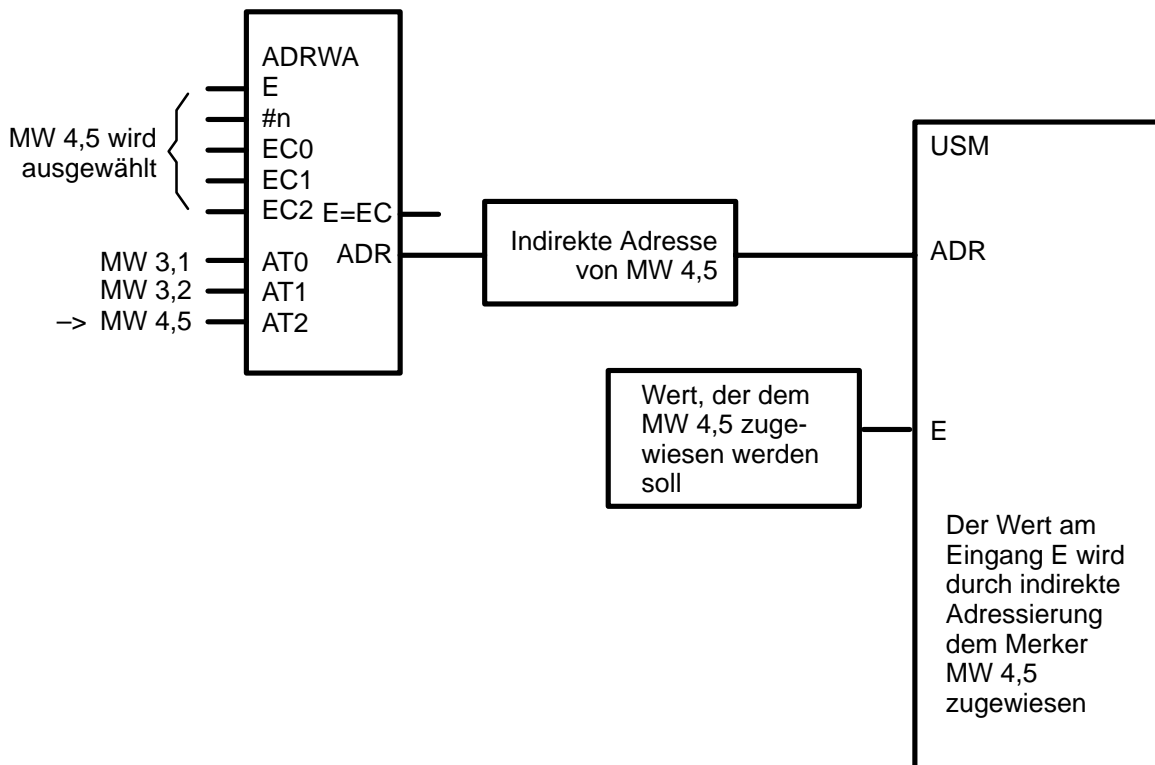
Der Operand am Ausgang ADR stellt zusammen mit seinem Wert eine indirekte Adresse dar. Es handelt sich dabei um die indirekte Adresse des aus der Ausgabetabelle $AT_0 \dots AT_{n-1}$ ausgewählten Operanden. Die indirekte Adresse kommt dadurch zustande, daß die Adresse des ausgewählten Operanden dem Operanden am Ausgang ADR als Wert zugewiesen wird.

Wird beim Vergleichsvorgang zwischen dem Eingang E und den Eingängen $EC_0 \dots EC_{n-1}$ keine Übereinstimmung festgestellt, so wird auch keine indirekte Adresse erzeugt. Dem Ausgang ADR wird also kein Wert zugewiesen. Der Ausgang ADR wird in diesem Fall nicht aktualisiert.

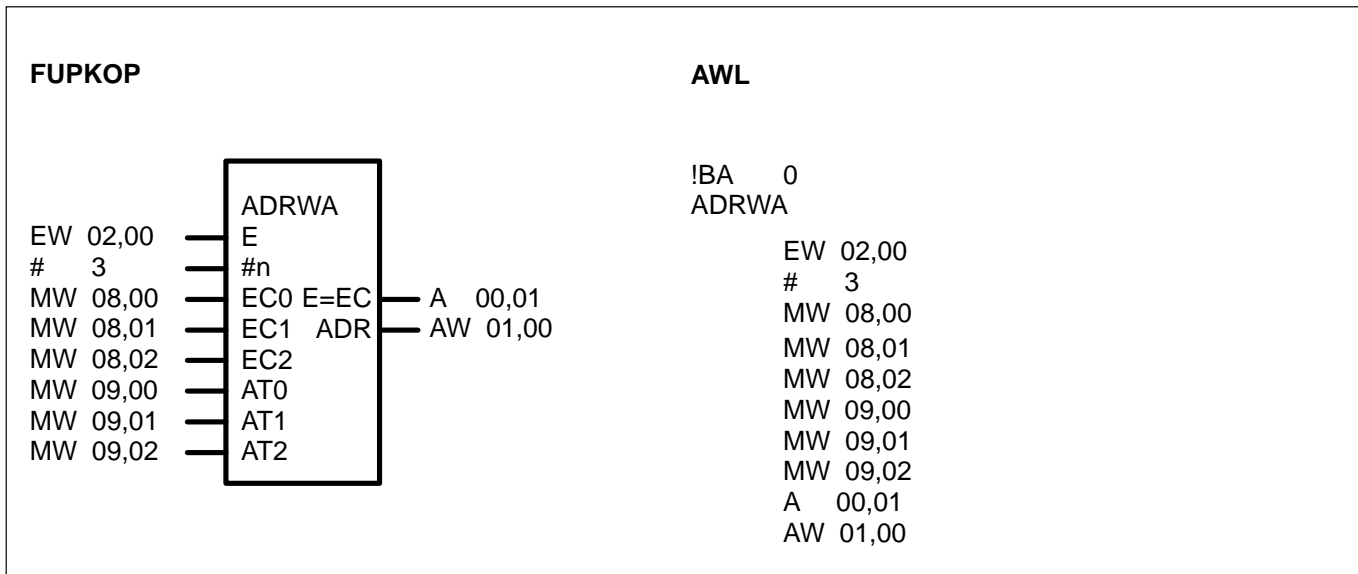
Beispiel: Indirektes Lesen des Merkers MW 3,2



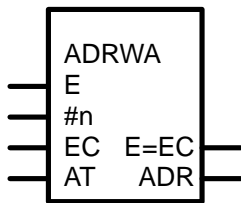
Beispiel: Indirektes Schreiben des Merkers MW 4,5



Beispiel



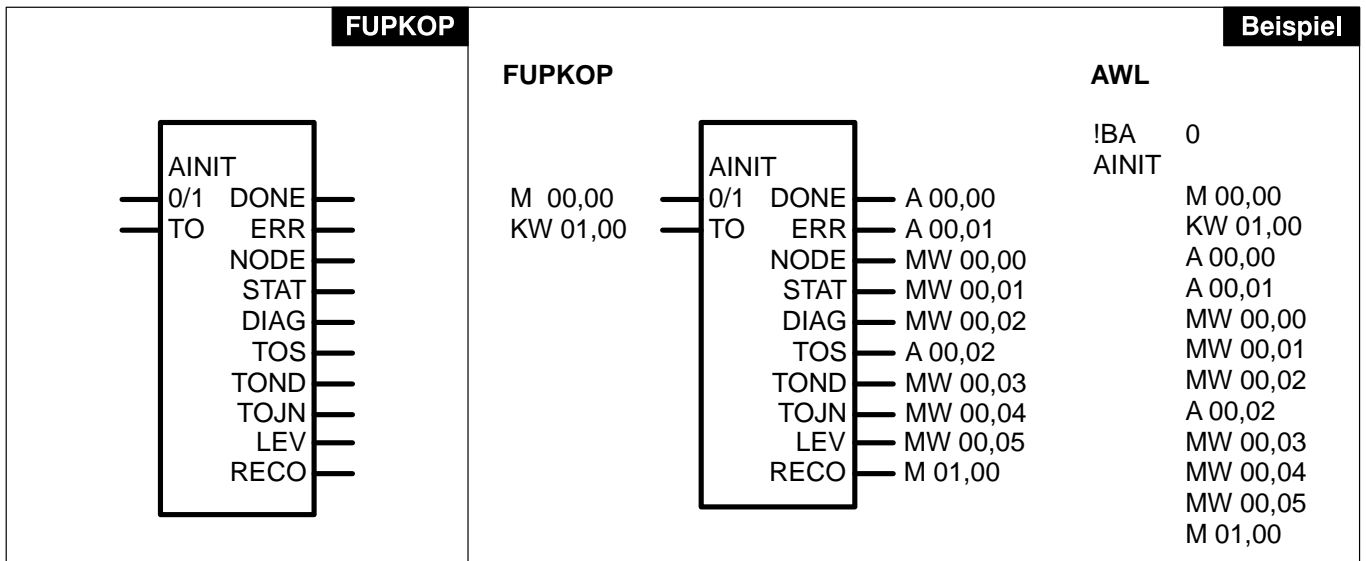
VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E	E	W	N	P	J	0	0
#n	K	W	N	P	J	0	0
EC	E	W	N	P	J	1	0
AT	E	W	N	P	J	1	0
E=EC	A	L	N	P	J	0	0
ADR	A	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	ADRWA			
00002		PP 0	E	Eingang WORT (Eingangswert)
00003		PP 0	#n	Eingang WORT (Anzahl EC und AT)
	[1		
00004		PP 0	EC	Eingang WORT (Eingangscodex)
]	1		
	[1		
00005		PP 0	AT	Eingang WORT (Ausgabetabelle)
]	1		
00006		PP 0	E=EC	Ausgang BINÄR (Eingangswert=Ecode)
00007		PP 0	ADR	Ausgang WORT (Operandenadresse)



Parameter

0/1	BINÄR	E, M, A, K, S	Initialisierung des ARCnet-Controllers mit 0/1-Flanke
TO	WORT	EW, MW, AW, KW	Timeout in ms für das Senden von Datenpaketen
DONE	BINÄR	A, M	Initialisierung abgeschlossen
ERR	BINÄR	A, M	Fehler aufgetreten
NODE	WORT	AW, MW	eigene Node-Nummer (Stations-Adresse)
STAT	WORT	AW, MW	Status-Register des ARCnet-Controllers
DIAG	WORT	AW, MW	Diagnose-Register des ARCnet-Controllers
TOS	BINÄR	A, M	Timeout beim Senden aufgetreten
TOND	WORT	AW, MW	Node-Nummer, Timeout-Datenpaket
TOJN	WORT	AW, MW	Job-Nummer, Timeout-Datenpaket
LEV	WORT	AW, MW	Level (Füllstand) des Sendebuffers
RECO	BINÄR	A, M	Netz-Rekonfiguration läuft (nach Tokenverlust)

Die Ausgänge STAT, DIAG, TOS, TOND, TOJN, LEV und RECO werden nach einer fehlerfreien Initialisierung bei jedem Bausteinaufruf aktualisiert.

Beschreibung

Der Funktionsbaustein AINIT initialisiert den ARCnet-Controller wie folgt:

- Interrupt nach dem Empfang eines Datenpakets
- nur kurze Pakete (short packages = 256 Bytes)
- Datenpakete an alle Stationen (Broadcasts)

Wichtiger Hinweis:

Bei der Verwendung einer SPS mit ARCnet-Anschluß wird ein Teil des SPS-TURBO-Programmspeicher-2 für den ARCnet reserviert.

Bei Programmen mit mehr als 2 k Anweisungen kann es aufgrund des reduzierten TURBO-Speichers-2 u. U. beim Durchführen einer Änderung eines laufenden Programmes zu einer systembedingten Vergrößerung der Auslastung kommen.

Keine Probleme treten auf, wenn die Auslastung vor der

Änderung eines laufenden Programmes kleiner als 80 % ist, oder wenn die Programmlänge kleiner als 2 k Anweisungen beträgt.

0/1 BINÄR

Mit einer 0/1-Flanke am Eingang 0/1 wird die einmalige Initialisierung des ARCnet-Controllers ausgelöst. Ist die Initialisierung noch nicht abgeschlossen (DONE = 0), wird eine neue 0/1-Flanke am Eingang 0/1 ignoriert.

TO WORT

Am Eingang TO wird die Timeout-Zeit in ms für das Senden von Datenpaketen angegeben. Kann ein Datenpaket innerhalb dieser Zeit nicht verschickt werden, wird das Senden dieses Datenpakets abgebrochen und das Datenpaket ist verloren. Dies wird am Ausgang TOS angezeigt.

DONE BINÄR

Am Ausgang DONE wird angezeigt, daß die Initialisierung abgeschlossen ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang ERR betrachtet werden.

Es gilt:

DONE = 1 und ERR = 0: Die Initialisierung ist abgeschlossen und es ist kein Fehler aufgetreten.

DONE = 1 und ERR = 1: Bei der Initialisierung ist ein Fehler aufgetreten. Der ARCnet-Controller hat innerhalb von 100 ms nicht geantwortet.

ERR BINÄR

Am Ausgang ERR wird angezeigt, daß der ARCnet-Controller nach einer Initialisierungsbefehl innerhalb von 100 ms nicht geantwortet hat. Der Ausgang muß immer im Zusammenhang mit dem Ausgang DONE betrachtet werden.

Ist ein Fehler aufgetreten, so gilt:

DONE = 1 und ERR = 1.

NODE WORT

Am Ausgang NODE wird nach einer fehlerfreien Initialisierung die eigene Node-Nummer (Stations-Adresse) ausgegeben.

STAT WORT

Am Ausgang STAT wird nach einer fehlerfreien Initialisierung der Wert des Status-Registers des ARCnet-Controllers ausgegeben.

DIAG WORT

Am Ausgang DIAG wird nach einer fehlerfreien Initialisierung der Wert des Diagnose-Registers des ARCnet-Controllers ausgegeben.

TOS BINÄR

Am Ausgang TOS wird angezeigt, daß das Senden eines Datenpakets innerhalb der Timeout-Zeit (Eingang TO) nicht möglich war und dieses Datenpaket verloren ist. Die Node-Nummer und die Job-Nummer des verlorenen Datenpaketes werden an den Ausgängen TOND und TOJN ausgegeben.

TOND WORT

Am Ausgang TOND wird nach einem Timeout die Node-Nummer des verlorenen Datenpakets ausgegeben.

TOJN WORT

Am Ausgang TOJN wird nach einem Timeout die Job-Nummer des verlorenen Datenpakets ausgegeben.

LEV WORT

Am Ausgang LEV wird nach einer fehlerfreien Initialisierung der Füllstand des Sende-Buffers ausgegeben.

RECO BINÄR

Am Ausgang RECO wird angezeigt, daß sich das Netz nach einem Tokenverlust neu konfiguriert (RECO = 1). Der Abschluß der Rekonfiguration wird durch RECO = 0 angezeigt.

Die Ausgänge STAT, DIAG, TOS, TOND, TOJN, LEV und RECO werden nach einer fehlerfreien Initialisierung bei jedem Bausteinaufruf aktualisiert.

Die Ein- und Ausgänge sind nicht doppelbar und nicht invertierbar.

VE-Daten

Laufzeit:

Grundlaufzeit:

Zusätzliche Laufzeit:

Aktualisierung der Ausgänge:

Anzahl der Vergangenheitswerte:

Verfügbar ab:

07 KT 92 / 07 KT 63

73...161 µs, typ. 93 µs

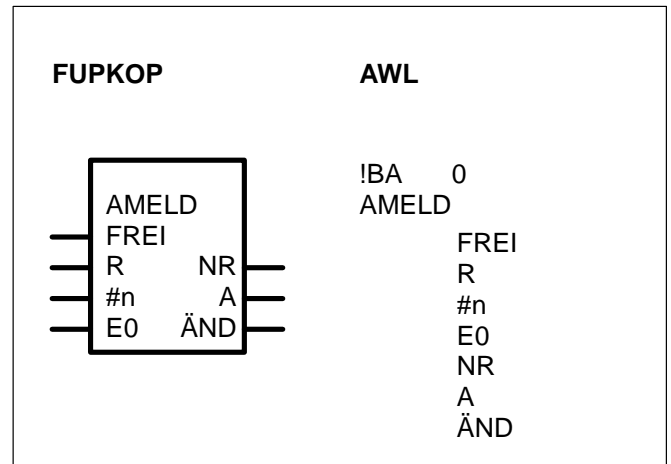
keine

ja

1 Wort

ABB Procontic CS31, 07 KT 92 R262, 07 KT 93 R171

Der Funktionsbaustein überwacht die am doppelbaren Eingang E0 anliegenden Analogwerte auf Änderung.



Parameter

FREI	BINÄR	A, E, M, S, K	Baustein-Freigabe
R	BINÄR	A, E, M, S, K	Reset
#n	DIREKTE KONSTANTE	#, #H	Anzahl Eingangswerte
E0	WORT	EW, AW, MW, KW	Eingangswerte; Eingang ist doppelbar
NR	WORT	AW, MW	Nummer des Eingangswertes
A	WORT	AW, MW	aktueller Eingangswert
ÄND	BINÄR	A, M	Änderung erkannt

VE-Daten

Laufzeit:	
Grundlaufzeit:	57 µs
Zusätzliche Laufzeit:	31 µs pro Eingang E0...E _{n-1}
Aktualisierung der Ausgänge:	ja, falls FREI = 1
Anzahl der Vergangenheitswerte:	#n + 3 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein überwacht die am doppelbaren Eingang E0 anliegenden Analogwerte auf Änderung.

Die Ein- und Ausgänge sind nicht negierbar/invertierbar. Der Eingang E0 ist doppelbar.

Erkennung einer Änderung

Bei jeder Bearbeitung des Bausteins werden der Reihe nach die aktuellen Eingangswerte an den Eingängen E0 ... E_{n-1} mit den Vergangenheitswerten (Eingangswerte der vorherigen Bausteinbearbeitung) verglichen. Wird an einem der Eingänge E0 ... E_{n-1} eine Änderung erkannt, so wird:

- dies am Ausgang ÄND signalisiert
- die Nummer des Eingangs, an dem die Änderung festgestellt wurde, am Ausgang NR ausgegeben
- der sich geänderte Eingangswert am Ausgang A ausgegeben

Pro Bearbeitung des Bausteins wird nur die Änderung an *einem* Eingang erkannt. Wird eine Änderung erkannt, so werden bei der nächsten Bearbeitung des Bausteins die Eingänge überwacht, die auf den Eingang folgen, an dem zuvor die Änderung festgestellt wurde.

Initialisierung der Vergangenheitswerte

Bei der ersten Bearbeitung nach der SPS-Initialisierung (Frei = 1) bzw. Freigabe der Bearbeitung nachdem sie gesperrt war (FREI wechselt von 0 nach 1), werden alle aktuellen Eingangswerte einmalig als Vergangenheitswerte übernommen, und alle Ausgänge werden auf den Wert 0 gesetzt. Diese initialisierten Vergangenheitswerte stellen nun die Ausgangsbasis zur Erkennung von Änderungen dar.

FREI BINÄR

Mit dem Eingang FREI wird die Bearbeitung des Baustein freigegeben.

FREI = 0 → Baustein wird nicht bearbeitet

FREI = 1 → Bearbeitung des Bausteins ist freigegeben

Ist FREI = 0, dann werden auch die Ausgänge des Bausteins nicht mehr aktualisiert.

R BINÄR

Mit dem Eingang R kann der Baustein zurückgesetzt werden (Reset).

R = 0 → kein Reset

R = 1 → Reset des Bausteins

Reset bedeutet:

- Übernahme der aktuellen Werte an den Eingängen $E_0 \dots E_{n-1}$ als Vergangenheitswerte.
- Alle Ausgänge werden auf den Wert 0 gesetzt.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der zu überwachenden Werte an den Eingängen $E_0 \dots E_{n-1}$ angegeben. Die Angabe erfolgt als direkte Konstante.

Bereich für #n: $1 \leq \#n \leq 127$

E0...E_{n-1} WORT

Der Eingang E0 ist doppelbar ($E_0 \dots E_{n-1}$).

An den Eingängen $E_0 \dots E_{n-1}$ werden die auf Änderung zu überwachenden Operanden angegeben.

NR WORT

Am Ausgang NR wird die laufende Nummer des Eingangs $E_0 \dots E_{n-1}$ ausgegeben, an dem eine Änderung festgestellt wurde.

Wird bei der Bearbeitung des Bausteins *keine* Änderung festgestellt, so wird weiterhin am Ausgang NR die Nummer des sich zuletzt geänderten Eingangs ausgegeben.

Es gilt die Zuordnung:

Änderung festgestellt an $E_0 \rightarrow NR = 0$

Änderung festgestellt an $E_1 \rightarrow NR = 1$

Änderung festgestellt an $E_{n-1} \rightarrow NR = n-1$

A WORT

Wird an einem der Eingänge $E_0 \dots E_{n-1}$ eine Änderung erkannt, so wird der sich geänderte Eingangswert dem Ausgang A zugewiesen.

Wird bei der Bearbeitung des Bausteins *keine* Änderung an den Eingängen $E_0 \dots E_{n-1}$ festgestellt, so wird am Ausgang A weiterhin der Wert des sich zuletzt geänderten Eingangs ausgegeben.

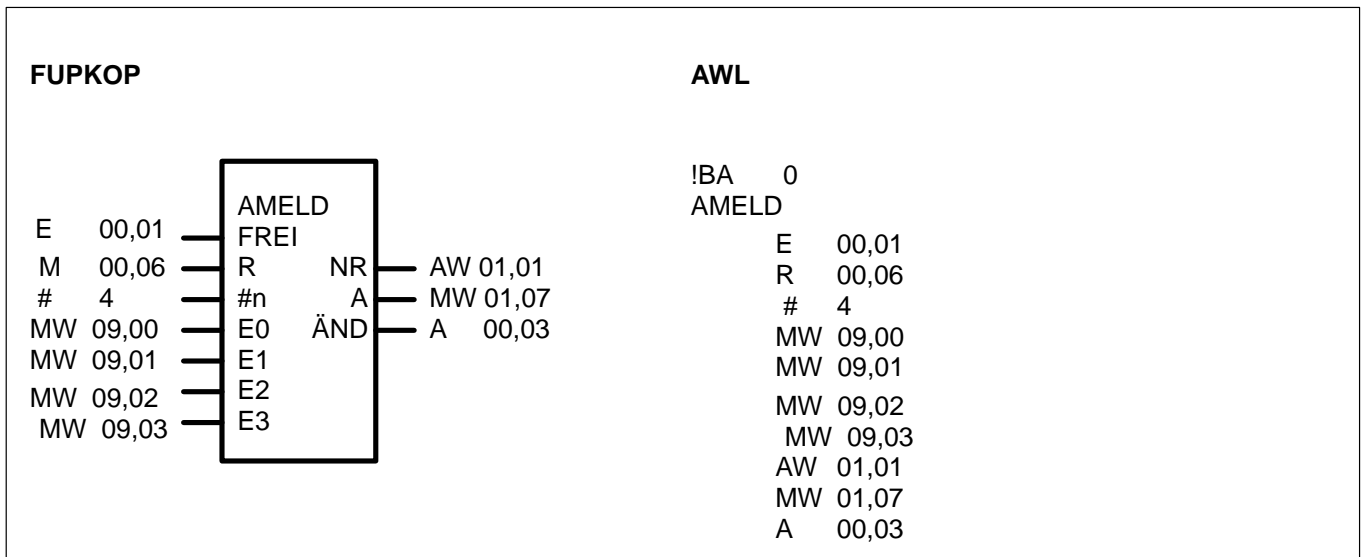
ÄND BINÄR

Am Ausgang ÄND wird angezeigt, ob eine Änderung an den Eingängen $E_0 \dots E_{n-1}$ erkannt wurde.

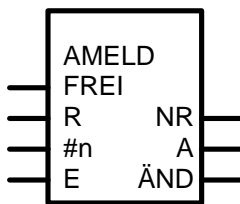
ÄND = 0 → keine Änderung festgestellt

ÄND = 1 → Änderung festgestellt

Beispiel



VE-FUP-Definition

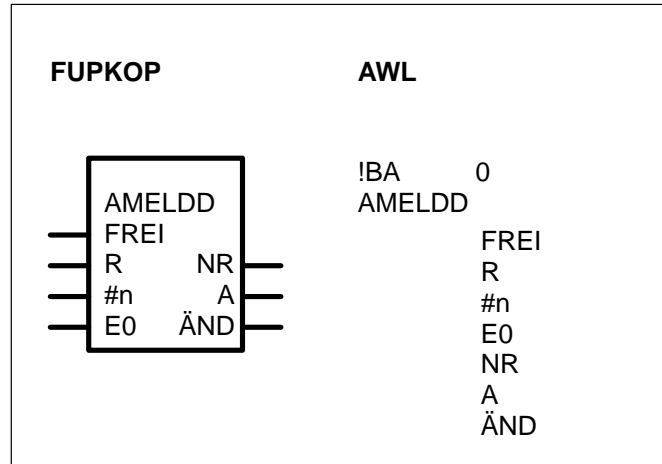


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
R	E	L	N	P	J	0	0
#n	K	W	N	P	J	0	0
E	E	W	N	P	J	1	0
NR	A	W	N	P	J	0	0
A	A	W	N	P	J	0	0
ÄND	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	AMELD			
00002		PP 0	FREI	Eingang BINÄR (Baustein-Freigabe)
00003		PP 0	R	Eingang BINÄR (Reset)
00004		PP 0	#n	# KONSTANTE (Anzahl Eingangswerte)
	[1		
00005		PP 1	E	Eingang WORT (Eingangswerte)
]	1		
00006		PP 0	NR	Ausgang WORT (Nummer)
00007		PP 0	A	Ausgang WORT (aktueller Eingangswert)
00008		PP 0	ÄND	Ausgang BINÄR (Änderung erkannt)

Der Funktionsbaustein überwacht die am doppelbaren Eingang E0 anliegenden Analogwerte auf Änderung.



Parameter

FREI	BINÄR	A, E, M, S, K	Baustein-Freigabe
R	BINÄR	A, E, M, S, K	Reset
#n	DIREKTE KONSTANTE	#, #H	Anzahl Eingangswerte
E0	DOPPELWORT	MD, KD	Eingangswerte; Eingang ist doppelbar
NR	WORT	AW, MW	Nummer des Eingangswertes
A	DOPPELWORT	MD	aktueller Eingangswert
ÄND	BINÄR	A, M	Änderung erkannt

VE-Daten

Laufzeit:	
Grundlaufzeit:	64 µs
Zusätzliche Laufzeit:	41 µs pro Eingang E0 ... E _{n-1}
Aktualisierung der Ausgänge:	ja, falls FREI = 1
Anzahl der Vergangenheitswerte:	(2 * #n) + 4 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein überwacht die am doppelbaren Eingang E0 anliegenden Analogwerte auf Änderung.

Die Ein- und Ausgänge sind nicht negierbar/invertierbar. Der Eingang E0 ist doppelbar.

Erkennung einer Änderung

Bei jeder Bearbeitung des Bausteins werden der Reihe nach die aktuellen Eingangswerte an den Eingängen E0 ... E_{n-1} mit den Vergangenheitswerten (Eingangswerte der vorherigen Bausteinbearbeitung) verglichen. Wird an einem der Eingänge E0 ... E_{n-1} eine Änderung erkannt, so wird:

- dies am Ausgang ÄND signalisiert
- die Nummer des Eingangs, an dem die Änderung festgestellt wurde, am Ausgang NR ausgegeben
- der sich geänderte Eingangswert am Ausgang A ausgegeben

Pro Bearbeitung des Bausteins wird nur die Änderung an *einem* Eingang erkannt. Wird eine Änderung erkannt, so werden bei der nächsten Bearbeitung des Bausteins die Eingänge überwacht, die auf den Eingang folgen, an dem zuvor die Änderung festgestellt wurde.

Initialisierung der Vergangenheitswerte

Bei der ersten Bearbeitung nach der SPS-Initialisierung (Frei = 1) bzw. Freigabe der Bearbeitung nachdem sie gesperrt war (FREI wechselt von 0 nach 1), werden alle aktuellen Eingangswerte einmalig als Vergangenheitswerte übernommen, und alle Ausgänge werden auf den Wert 0 gesetzt. Diese initialisierten Vergangenheitswerte stellen nun die Ausgangsbasis zur Erkennung von Änderungen dar.

FREI BINÄR

Mit dem Eingang FREI wird die Bearbeitung des Baustein freigegeben.

FREI = 0 → Baustein wird nicht bearbeitet

FREI = 1 → Bearbeitung des Bausteins ist freigegeben

Ist FREI = 0, dann werden auch die Ausgänge des Bausteins nicht mehr aktualisiert.

R BINÄR

Mit dem Eingang R kann der Baustein zurückgesetzt werden (Reset).

R = 0 → kein Reset

R = 1 → Reset des Bausteins

Reset bedeutet:

- Übernahme der aktuellen Werte an den Eingängen $E_0 \dots E_{n-1}$ als Vergangenheitswerte.
- Alle Ausgänge werden auf den Wert 0 gesetzt.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der zu überwachenden Werte an den Eingängen $E_0 \dots E_{n-1}$ angegeben. Die Angabe erfolgt als direkte Konstante.

Bereich für #n: $1 \leq \#n \leq 63$

E0...E_{n-1} DOPPELWORT

Der Eingang E0 ist doppelbar ($E_0 \dots E_{n-1}$).

An den Eingängen $E_0 \dots E_{n-1}$ werden die auf Änderung zu überwachenden Operanden angegeben.

NR WORT

Am Ausgang NR wird die laufende Nummer des Eingangs $E_0 \dots E_{n-1}$ ausgegeben, an dem eine Änderung festgestellt wurde.

Wird bei der Bearbeitung des Bausteins *keine* Änderung festgestellt, so wird weiterhin am Ausgang NR die Nummer des sich zuletzt geänderten Eingangs ausgegeben.

Es gilt die Zuordnung:

Änderung festgestellt an $E_0 \rightarrow NR = 0$

Änderung festgestellt an $E_1 \rightarrow NR = 1$

Änderung festgestellt an $E_{n-1} \rightarrow NR = n-1$

A DOPPELWORT

Wird an einem der Eingänge $E_0 \dots E_{n-1}$ eine Änderung erkannt, so wird der sich geänderte Eingangswert dem Ausgang A zugewiesen.

Wird bei der Bearbeitung des Bausteins *keine* Änderung an den Eingängen $E_0 \dots E_{n-1}$ festgestellt, so wird am Ausgang A weiterhin der Wert des sich zuletzt geänderten Eingangs ausgegeben.

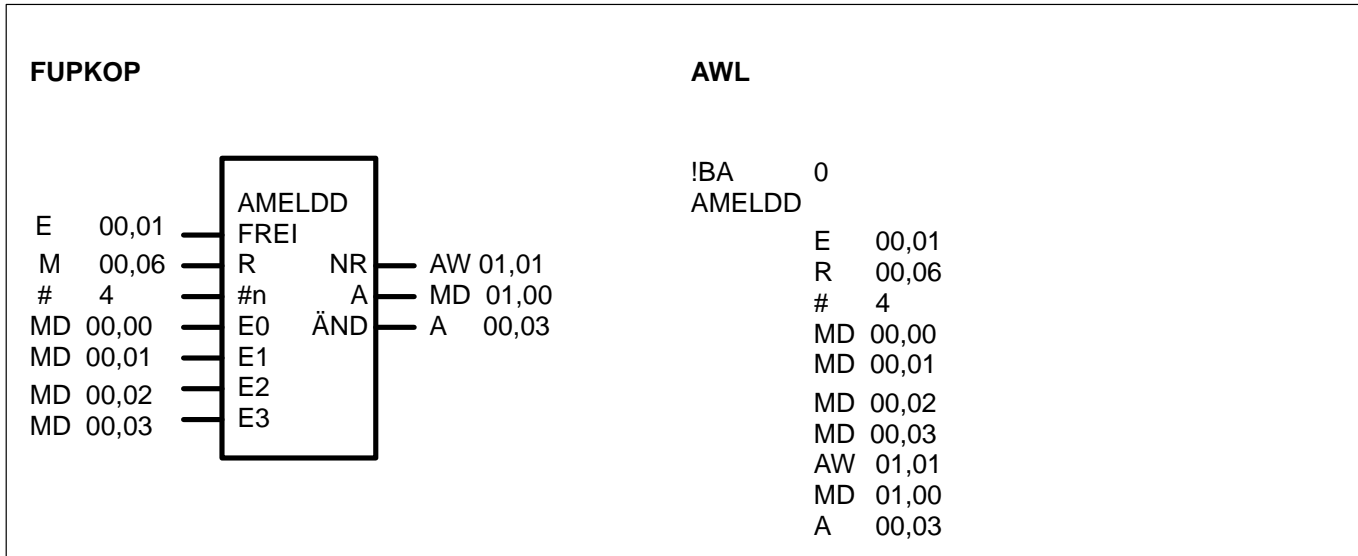
ÄND BINÄR

Am Ausgang ÄND wird angezeigt, ob eine Änderung an den Eingängen $E_0 \dots E_{n-1}$ erkannt wurde.

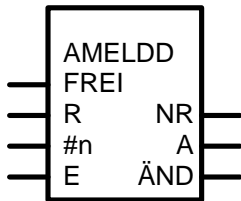
ÄND = 0 → keine Änderung festgestellt

ÄND = 1 → Änderung festgestellt

Beispiel



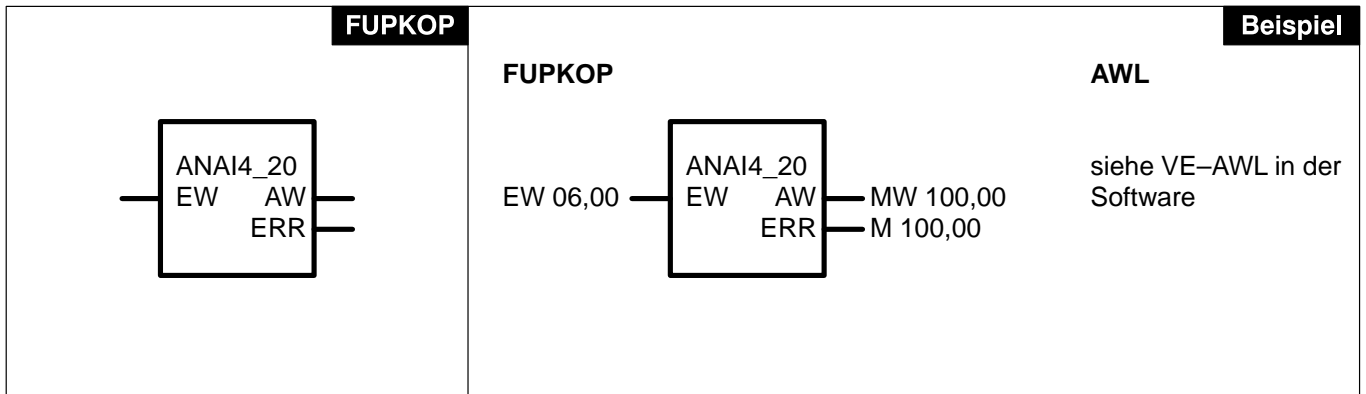
VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
R	E	L	N	P	J	0	0
#n	K	W	N	P	J	0	0
E	E	D	N	P	J	1	0
NR	A	W	N	P	J	0	0
A	A	D	N	P	J	0	0
ÄND	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	AMELDD			
00002		PP 0	FREI	Eingang BINÄR (Baustein-Freigabe)
00003		PP 0	R	Eingang BINÄR (Reset)
00004		PP 0	#n	# KONSTANTE (Anzahl Eingangswerte)
	[1		
00005		PP 1	E	Eingang DOPPELWORT (Eingangswerte)
]	1		
00006		PP 0	NR	Ausgang WORT (Nummer)
00007		PP 0	A	Ausgang DOPPELWORT (aktueller Eingangswert)
00008		PP 0	ÄND	Ausgang BINÄR (Änderung erkannt)



Parameter

EW	WORT	EW, MW	Direkter Analogeingang
AW	WORT	MW, AW	Wort-Variable, der das Ergebnis der Auswertung zugewiesen wird (0...32760 entspricht 4...20 mA)
ERR	BINÄR	M, A	Ausgang Drahtbruchüberwachung (1 = Drahtbruch)

Beschreibung

Das Verknüpfungselement ANAI4_20 dient zum Einlesen von Stromsignalen mit 4...20 mA Signalbereich an einem der vier direkten Analogeingänge EW06,00...EW06,03 der Kompaktsteuerung 07 KT 92. Der Eingangsbereich 4...20 mA wird in den internen Zahlenbereich 0...32760 gewandelt. Eine Drahtbruchauswertung findet statt.

Die zusammen mit diesem Verknüpfungselement benutzten Eingänge müssen als Stromeingänge beschaltet sein. Für jeden Eingang, der als 4...20 mA Stromeingang benutzt wird, ist ein Verknüpfungselement ANAI4_20 erforderlich.

Der Analogeingang ist konstruktiv für einen Eingangsbereich von 0...20 mA ausgelegt. Die Analog-Eingangswerte EW06,00...EW06,03 liefern einen Wert im Bereich 0...32760 entsprechend 0...20 mA. Das Verknüpfungselement ANAI4_20 bildet den Zahlenbereich (an EW) von 6552...32760 für 4...20 mA auf den Bereich (an AW) 0...32760 ab. Die Schrittweite des Ausgangswertes AW beträgt 10, das sind 0,004883 mA. Eingangswerte an EW, die kleiner 6552 bzw. 4 mA sind, führen zu einem Ausgangswert AW von 0.

EW WORT Direkter Analogeingang
Am Eingang EW wird der zu lesende Analog-Eingang spezifiziert.

AW WORT Analogwert in interner Darstellung
Am Wortausgang AW wird der in die interne Darstellung gewandelte Wert des Analogsignals ausgegeben. Im Falle einer Drahtbruchmeldung geht der Ausgang auf 0. Das Diagramm 1 zeigt den Zusammenhang zwischen dem Eingangsstrom am direkten Analogeingang und dem Ausgangswert AW.

ERR BINÄR Drahtbruchüberwachung
Die im Verknüpfungselement ANAI4_20 integrierte Drahtbruchüberwachung signalisiert am Binärausgang ERR des Verknüpfungselementes einen Drahtbruch (mit ERR = 1). Ein Eingangsstrom am Analogeingang der Kompaktsteuerung, der kleiner als 3,6 mA ist, führt zu einer Drahtbruchmeldung (siehe auch Diagramm 2). Gleichzeitig wird der Ausgang AW auf 0 gesetzt.

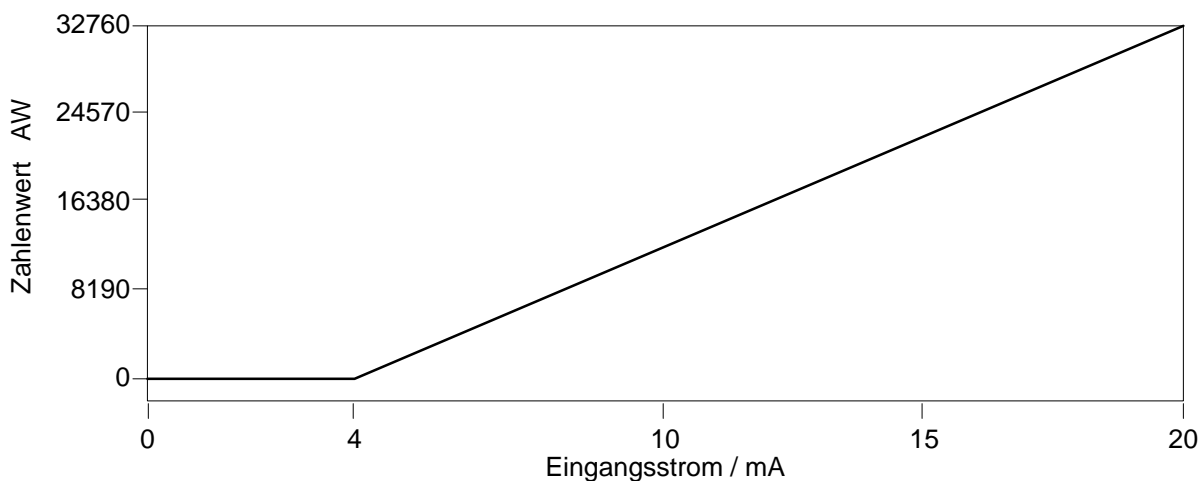


Diagramm 1: Abhängigkeit des Ausgangswertes AW vom Eingangsstrom am direkten Analogeingang

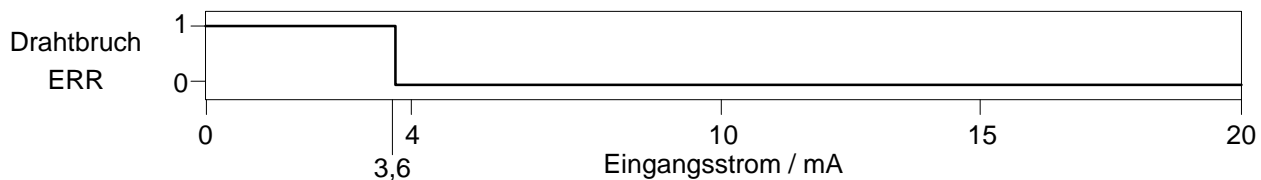


Diagramm 2: Der Ausgang ERR signalisiert einen Drahtbruch bei einem Eingangsstrom < 3,6 mA

VE-Daten

Laufzeit:

Grundlaufzeit: 180 μ s

Zusätzliche Laufzeit: —

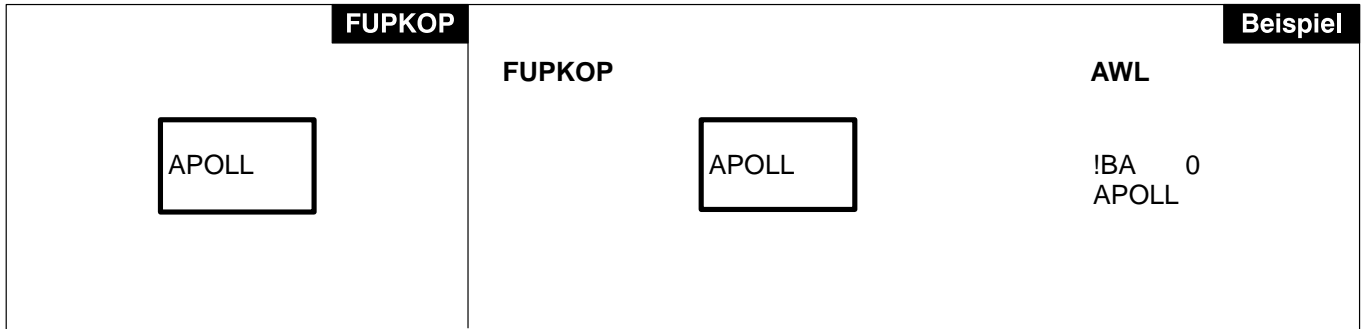
Aktualisierung der Ausgänge: ja

Anzahl der Vergangenheitswerte: keine

Benutzte Operanden:

Lokale Wortmerker: 7

Verfügbar ab: ABB Procontic CS31 / 907 PC 331 R202



Beschreibung

Der Funktionsbaustein APOLL hat zwei Aufgaben:

- Er übergibt ein Datenpaket aus dem Zwischenbuffer dem ARCnet–Controller zum Versenden. Die Datenpakete werden vom Funktionsbaustein ASEND in dem Zwischenbuffer abgelegt.
- Er überwacht, ob dieses Datenpaket innerhalb der am Funktionsbaustein AINIT projektierten Timeout–Zeit verschickt wurde.

Kann das Datenpaket innerhalb der projektierten Timeout–Zeit vom ARCnet–Controller nicht verschickt werden, bricht der Baustein den Sendevorgang ab und dieses Datenpaket ist verloren. Dies wird beim nächsten Aufruf des Bausteins AINIT am Ausgang TOS signalisiert.

Werden über das Anwenderprogramm verteilt mehrere APOLL–Bausteine projektiert, so können mehrere Datenpakete innerhalb eines Zyklusses verschickt werden.

Bevor mit dem Funktionsbaustein APOLL ein Datenpaket aus dem Zwischenbuffer dem ARCnet–Controller übergeben werden kann, muß der ARCnet–Controller initialisiert sein (Funktionsbaustein AINIT).

Wichtiger Hinweis:

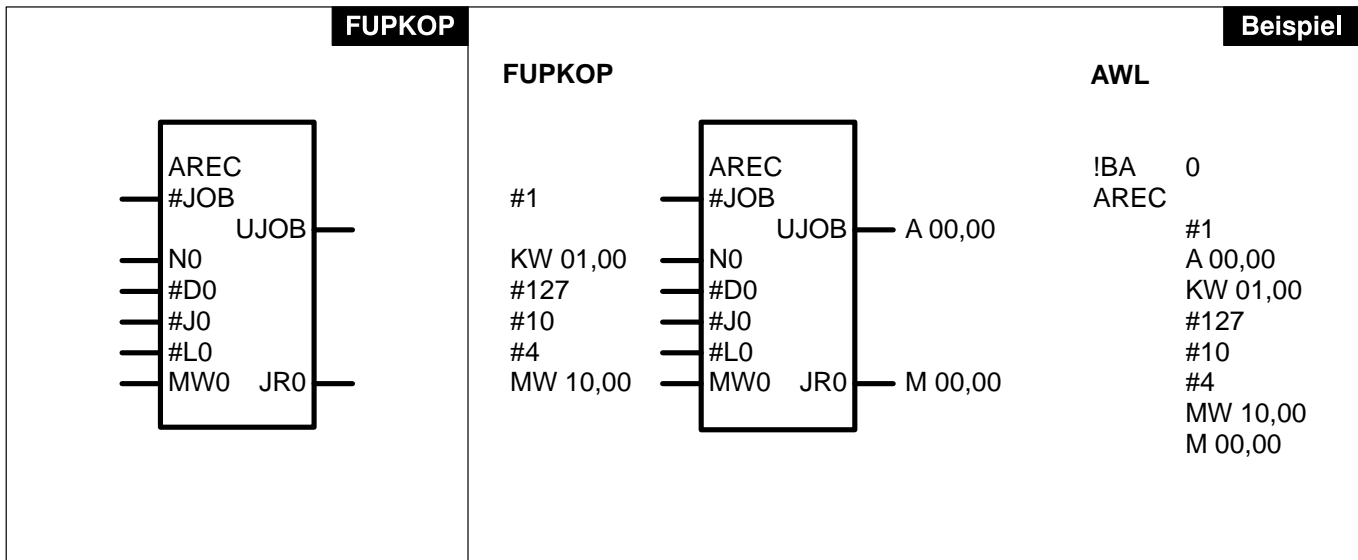
Bei der Verwendung einer SPS mit ARCnet–Anschluß wird ein Teil des SPS–TURBO–Programmspeicher–2 für den ARCnet reserviert.

Bei Programmen mit mehr als 2 k Anweisungen kann es aufgrund des reduzierten TURBO–Speichers–2 u. U. beim Durchführen einer *Änderung* eines *laufenden* Programmes zu einer systembedingten Vergrößerung der Auslastung kommen.

Keine Probleme treten auf, wenn die Auslastung *vor* der Änderung eines laufenden Programmes *kleiner* als 80 % ist, oder wenn die Programmlänge kleiner als 2 k Anweisungen beträgt.

VE–Daten

Laufzeit:	07 KT 92 / 07 KT 93
Grundlaufzeit:	40 µs
Zusätzliche Laufzeit:	47...187 µs
Aktualisierung der Ausgänge:	—
Anzahl der Vergangenheitswerte:	0 Worte
Verfügbar ab:	ABB Procontic CS31, 07 KT 92 R262, 07 KT 93 R171



Parameter

#JOB	DIREKTE KONSTANTE	#, #H	Gesamtanzahl im Baustein projektierter Jobs
N0	WORT	EW, MW, AW, KW	Node-Nummer (Stations-Adresse) des Absenders
#D0	DIREKTE KONSTANTE	#, #H	DIN-Kennung
#J0	DIREKTE KONSTANTE	#, #H	Job-Nummer
#L0	DIREKTE KONSTANTE	#, #H	Anzahl der zu empfangenden Nutzdaten-Worte
MW0	WORT	EW, MW, AW, KW	erste Wortvariable, ab der die empfangenen Nutzdaten abgelegt werden
UJOB	BINÄR	A, M	unbekannter Job empfangen
JR0	BINÄR	A, M	Job empfangen

Die Eingänge N0, #D0, #J0, #L0 und MW0 und der Ausgang JR0 sind als Block doppelbar.

Beschreibung

Das Betriebssystem liest die empfangenen Datenpakete interruptgesteuert aus dem ARCNET-Controller aus und legt sie in einem Zwischenbuffer ab. Die Größe des Zwischenbuffers beträgt 31 Datenpakete.

Der Baustein liest den gesamten Zwischenbuffer aus und ordnet die Datenpakete den im Baustein projizierten Jobs zu.

Mit einem Job werden dem Empfänger logisch zusammengehörige Nutzdaten übermittelt. Zur eindeutigen Identifizierung eines Jobs dienen die Parameter:

- Node-Nummer des Absenders,
- DIN-Kennung und
- Job-Nummer.

Die Parameterwerte der empfangenen Datenpakete wer-

den mit den Parameterwerten des projizierten Jobs verglichen. Bei Übereinstimmung werden die Nutzdaten des Datenpakets ab der Wortvariablen MW0 fortlaufend abgelegt und der Ausgang JR0 auf 1 gesetzt. Wird die Übereinstimmung bei mehreren Datenpaketen festgestellt, so wertet der Baustein nur das Datenpaket aus, das als letztes empfangen wurde (neuestes Datenpaket).

Kann einem projizierten Job kein Datenpaket aus dem Zwischenbuffer zugeordnet werden, so wird der zu diesem projizierten Job gehörende Ausgang JR = 0 gesetzt.

Der Ausgang UJOB signalisiert, daß ein im Zwischenbuffer vorhandenes Datenpaket keinem Job zugeordnet werden konnte.

Es darf **maximal 1** Bausteinaufruf AREC im Anwenderprogramm projiziert sein.

Bevor mit dem Funktionsbaustein AREC Datenpakete aus dem Zwischenbuffer ausgelesen werden können, muß der ARCnet–Controller initialisiert sein (Funktionsbaustein AINIT).

Wichtiger Hinweis:

Bei der Verwendung einer SPS mit ARCnet–Anschluß wird ein Teil des SPS–TURBO–Programmspeicher–2 für den ARCnet reserviert.

Bei Programmen mit mehr als 2 k Anweisungen kann es aufgrund des reduzierten TURBO–Speichers–2 u. U. beim Durchführen einer *Änderung* eines *laufenden* Programmes zu einer systembedingten Vergrößerung der Auslastung kommen.

Keine Probleme treten auf, wenn die Auslastung *vor* der *Änderung* eines laufenden Programmes *kleiner* als 80 % ist, oder wenn die Programmlänge kleiner als 2 k Anweisungen beträgt.

#JOB DIREKTE KONSTANTE

Am Eingang #JOB wird die Gesamtanzahl der Jobs angegeben, die in diesem Baustein projiziert sind.

Es gilt: $1 \leq \#JOB$

N0 WORT

Am Eingang N0 wird die Node–Nummer (Stations–Adresse) des Absenders für den projizierten Job angegeben.

Es gilt: $0 \leq N0 \dots Nn-1 \leq 255$

#D0 DIREKTE KONSTANTE

Am Eingang #D0 wird die DIN–Kennung für den projizierten Job angegeben.

Es gilt: $0 \leq \#D \leq 127$

verboten: 111 (6FH)

#J0 DIREKTE KONSTANTE

Am Eingang #J0 wird die Job–Nummer für den projizierten Job angegeben.

Es gilt: $-32767 \leq \#J \leq +32767$

#L0 DIREKTE KONSTANTE

Am Eingang #L0 wird die Anzahl der Nutzdaten–Worte für den projizierten Job angegeben.

Es gilt: $0 \leq \#L \leq 125$

MW0 WORT

Am Eingang MW0 wird der Anfangswortmerker für die Nutzdaten des projizierten Jobs angegeben.

UJOB BINÄR

Am Ausgang UJOB wird signalisiert, daß im Zwischenbuffer ein Datenpaket abgelegt war, das keinem projizierten Job zugeordnet werden konnte.

Es gilt:

UJOB = 1: unbekannter Job empfangen

UJOB = 0: kein unbekannter Job empfangen

JR0 BINÄR

Am Ausgang JR0 (Job received) wird signalisiert, daß ein Datenpaket aus dem Zwischenbuffer diesem Job zugeordnet wurde. Die Nutzdaten des Datenpakets sind ab der Wortvariablen MW0 fortlaufend abgelegt.

Es gilt:

JR0 = 1: Job empfangen

JR0 = 0: kein Job empfangen

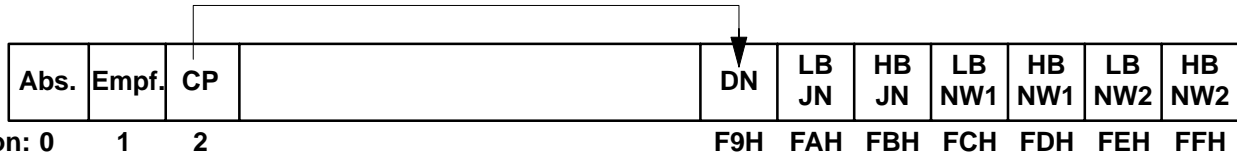
Hinweis

Die Ein– und Ausgänge sind nicht invertierbar. Die Eingänge N0, #D0, #J0, #L0 und MW0 und der Ausgang JR0 sind als Block doppelbar.

ARCnet–Datenpaket–Aufbau (kurzes Paket = 256 Bytes)

Das Datenpaket besteht aus:

- 3 Bytes Kontrolldaten: Abs., Empf., CP
- max. 253 Bytes Anwenderdaten bestehend aus:
1 Byte DIN–Kennung + 2 Bytes Job–Nummer + max. 125 Worte Nutzdaten



Position: 0 1 2

F9H FAH FBH FCH FDH FEH FFH

Bsp. projektierter Job:

N0: KW 01,00 = 5

#D0: #127 (7FH)

#J0: #10 (0AH)

#L0: #2

MW0: MW 10,00 = 258 (102H)

MW 10,01 = 772 (304H)

Abs.: Stations–Adresse des Senders (05H)

Empf.: Stations–Adresse des Empfängers

CP: Continuous Pointer; enthält die Position (F9H) des 1. Anwenderdaten–Bytes

DN: DIN–Kennung (7FH)

LB JN: Low Byte Job–Nummer (0AH)

HB JN: High Byte Job–Nummer (00H)

LB NW1: Low Byte Nutzwort 1 (02H)

HB NW1: High Byte Nutzwort 1 (01H)

LB NW2: Low Byte Nutzwort 2 (04H)

HB NW2: High Byte Nutzwort 2 (03H)

VE–Daten

Laufzeit:

07 KT 92 / 07 KT 93

Grundlaufzeit:

$(97 + 8 * \#JOB) \mu s$

Zusätzliche Laufzeit:

$((52...178 + 16 * \#JOB) * \#JOB) \mu s$

Aktualisierung der Ausgänge:

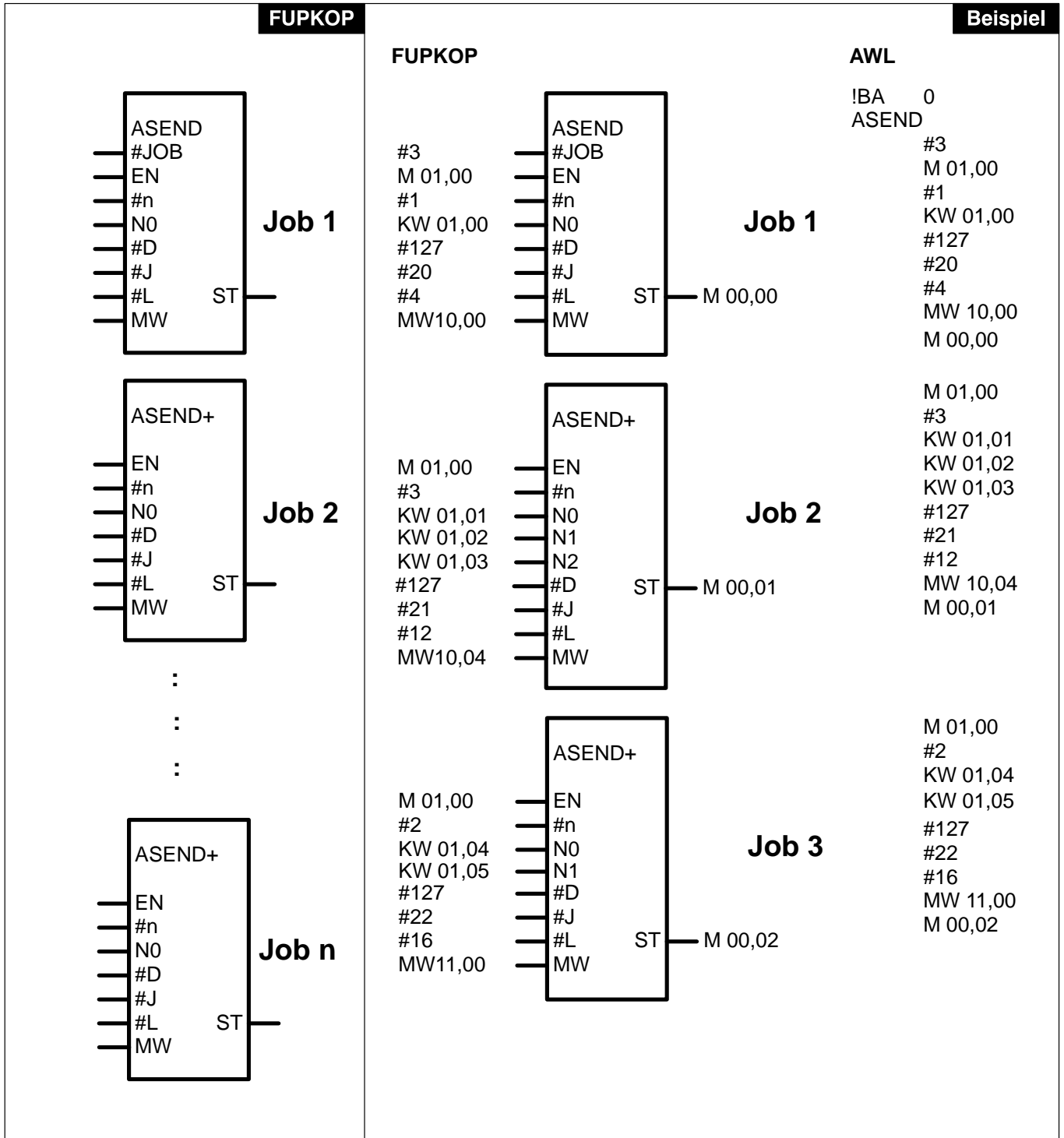
nein

Anzahl der Vergangenheitswerte:

0 Worte

Verfügbar ab:

ABB Procontic CS31, 07 KT 92 R262, 07 KT 93 R171



Parameter

#JOB	DIREKTE KONSTANTE	#, #H	Gesamtanzahl im Baustein projektierter Jobs
EN	BINÄR	E, M, A, K, S	Freigabe: Job senden 0: nicht senden 0/1–Flanke: senden 1: bei Änderung der Nutzdaten senden
#n	DIREKTE KONSTANTE	#, #H	Anzahl der Empfänger des Datenpaketes; es gilt: $1 \leq \#n0 \leq 15$
N0	WORT	EW, MW, AW, KW	Node–Nummer (Stations–Adresse) des Empfängers. Eingang ist doppelbar.
#D	DIREKTE KONSTANTE	#, #H	DIN–Kennung
#J	DIREKTE KONSTANTE	#, #H	Job–Nummer
#L	DIREKTE KONSTANTE	#, #H	Anzahl der zu sendenden Nutzdaten–Worte
MW	WORT	EW, MW, AW, KW	erste Wortvariable der Nutzworte
ST	BINÄR	A, M	Job im Zwischenbuffer abgelegt

Der Eingang N0 ist doppelbar;

Die Eingänge EN, #n, N0, #D, #J, #L und MW und der Ausgang ST sind die Parameter, die einen Job definieren. Diese Parameter erscheinen deshalb sowohl im Baustein ASEND, als auch in jedem Erweiterungs–Verknüpfungselement ASEND+.

Beschreibung

Der Funktionsbaustein ASEND dient zum Verschicken von Jobs (Datenpaketen) über das ARCnet–Netzwerk. Die projizierten Jobs werden in einem Zwischenbuffer abgelegt. Von dort werden sie dem ARCnet–Controller zum Senden übergeben. Der Transport vom Zwischenbuffer zum ARCnet–Controller erfolgt durch den Funktionsbaustein APOLL und zusätzlich in den Pausen zwischen zwei SPS–Zyklen durch das Betriebssystem.

Die Größe des Zwischenbuffers beträgt maximal 31 Datenpakete.

Der Ausgang ST signalisiert, daß der projizierte Job im Zwischenbuffer abgelegt wurde.

Wichtiger Hinweis:

Es darf **maximal 1** Bausteinaufruf ASEND im Anwenderprogramm projiziert sein. Soll mehr als *ein* Job verschickt werden, so wird der ASEND mit dem Verknüpfungselement ASEND+ erweitert. Für *jeden* weiteren Job wird im FUP *ein* ASEND+ *direkt im Anschluß* an den ASEND projiziert.

Der ASEND+ ist kein eigenständiger Baustein, sondern nur eine Erweiterung des ASEND Bausteins. Aus diesem Grund müssen alle benötigten ASEND+ direkt im Anschluß an den ASEND projiziert werden. Zwischen dem ASEND und dem ersten ASEND+ dürfen auf keinen Fall

andere VEs projiziert werden. Dasselbe Verbot gilt auch zwischen den einzelnen ASEND+ Bausteinen.

Aus dem gleichen Grund darf der Ausgang ST auch nicht durch eine Linie mit einem anderen VE verbunden werden. Am Ausgang ST ist deshalb stets eine Variable anzugeben.

Die VEs ASEND+ existieren nur im FUP. Beim Übersetzen werden der ASEND und alle nachfolgenden ASEND+ zu *einem* ASEND–AWL–Funktionsbaustein zusammengefaßt.

Bevor der Funktionsbaustein ASEND Datenpakete im Zwischenbuffer ablegen kann, muß der ARCnet–Controller initialisiert sein (Funktionsbaustein AINIT).

Wichtiger Hinweis:

Bei der Verwendung einer SPS mit ARCnet–Anschluß wird ein Teil des SPS–TURBO–Programmspeicher–2 für den ARCnet reserviert.

Bei Programmen mit mehr als 2 k Anweisungen kann es aufgrund des reduzierten TURBO–Speichers–2 u. U. beim Durchführen einer *Änderung* eines *laufenden* Programmes zu einer systembedingten Vergrößerung der Auslastung kommen.

Keine Probleme treten auf, wenn die Auslastung *vor* der *Änderung* eines laufenden Programmes *kleiner* als 80 % ist, oder wenn die Programmlänge kleiner als 2 k Anweisungen beträgt.

#JOB DIREKTE KONSTANTE

Am Eingang #JOB wird die *Gesamtanzahl* der Jobs angegeben, die im Baustein ASEND und in allen nachfolgenden Erweiterungs–Verknüpfungselementen ASEND+ projiziert sind.

Es gilt: $1 \leq \#JOB$

EN BINÄR

Der projizierte Job wird in Abhängigkeit vom Eingang EN im Zwischenbuffer abgelegt und anschließend versendet.

Es gilt:

EN = 0: Der projizierte Job wird *nicht* im Zwischenbuffer abgelegt und damit auch nicht versendet.

EN = 0/1–Flanke: Der projizierte Job wird als Datenpaket im Zwischenbuffer abgelegt und dann versendet.

EN = 1: Der projizierte Job wird *nur* dann im Zwischenbuffer abgelegt und versendet, falls sich die Nutzdaten des Jobs *geändert* haben.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der Empfänger für den projizierten Job angegeben. Die Node–Nummern der Empfänger werden an den Eingängen N0... Nn–1 projiziert.

Es gilt: $1 \leq \#n \leq 15$

N0...Nn–1 WORT

Soll ein und derselbe Job an mehrere Empfänger verschickt werden, so werden an den Eingängen N0 ... Nn–1 die Node–Nummern (Stations–Adressen) der Empfänger angegeben.

N0: Node–Nummer = 0:

Bei der Node–Nummer "0" am Eingang N0 wird der Job an alle Stationen gesendet (Broadcast).

N1...Nn–1: Node–Nummer = 0:

Bei der Node–Nummer "0" an den Eingängen N1...Nn–1 wird kein Broadcast gesendet.

Falls beim Broadcast ein Empfänger nicht zum Empfang bereit ist, ist für diesen Empfänger das Datenpaket verloren.

Es gilt: $0 \leq N0...Nn-1 \leq 255$

#D DIREKTE KONSTANTE

Am Eingang #D wird die DIN–Kennung für den projizierten Job angegeben. Der Wert 111 (6F_H) ist dabei nicht zulässig, weil dieser Wert für Programmier– und Testprogramme reserviert ist.

Es gilt: $0 \leq \#D \leq 127$

verboten: 111 (6F_H)

#J DIREKTE KONSTANTE

Am Eingang #J wird die Job–Nummer für den projizierten Job angegeben.

Es gilt: $-32767 \leq \#J \leq +32767$

#L DIREKTE KONSTANTE

Am Eingang #L wird die Anzahl der Nutzdaten–Worte des projizierten Jobs angegeben.

Es gilt: $0 \leq \#L \leq 125$

MW WORT

Am Eingang MW wird der Anfangswortmerker des zu versendenden Wort–Merkerbereichs angegeben. Die Daten dieses Merkerbereichs werden in diesem Job als Nutzdaten verschickt.

ST BINÄR

Am Ausgang ST (Status) wird signalisiert, daß der projizierte Job im Zwischenbuffer abgelegt wurde.

Es gilt:

ST = 0: Der projizierte Job wurde nicht im Zwischenbuffer abgelegt.

ST = 1: Der projizierte Job wurde im Zwischenbuffer abgelegt.

Hinweis

Die Ein– und Ausgänge sind nicht invertierbar.

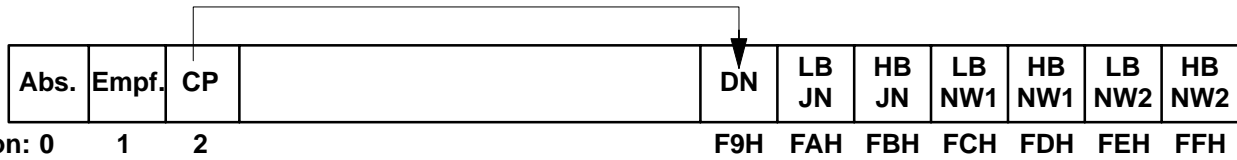
Der Eingang N0 ist doppelbar.

Am Ausgang ST muß eine Variable angegeben werden d. h. der Ausgang darf nicht durch eine Linie mit einem anderen VE verbunden werden.

ARCnet–Datenpaket–Aufbau (kurzes Paket = 256 Bytes)

Das Datenpaket besteht aus:

- 3 Bytes Kontrolldaten: Abs., Empf., CP
- max. 253 Bytes Anwenderdaten bestehend aus:
 1 Byte DIN–Kennung + 2 Bytes Job–Nummer + max. 125 Worte Nutzdaten



Bsp. projektiertes Job:

N0: KW 01,01 = 5
 #D: #127 (7FH)
 #J: #10 (0AH)
 #L: #2
 MW: MW 10,00 = 258 (102H)
 MW 10,01 = 772 (304H)

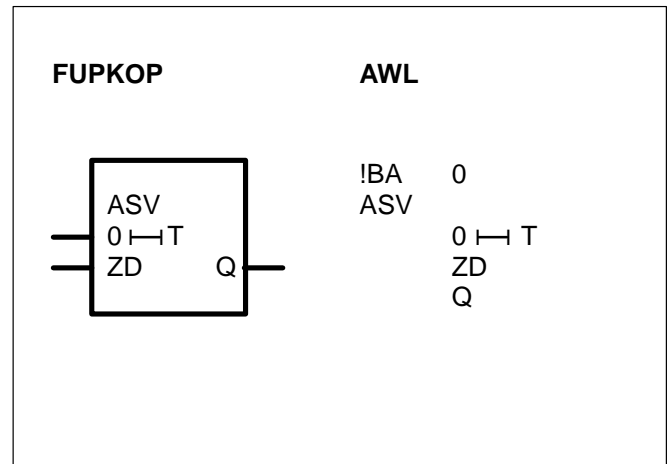
Abs.: Stations–Adresse des Senders (05H)
 Empf.: Stations–Adresse des Empfängers
 CP: Continuous Pointer; enthält die Position (F9H) des 1. Anwenderdaten–Bytes
 DN: DIN–Kennung (7FH)
 LB JN: Low Byte Job–Nummer (0AH)
 HB JN: High Byte Job–Nummer (00H)
 LB NW1: Low Byte Nutzwort 1 (02H)
 HB NW1: High Byte Nutzwort 1 (01H)
 LB NW2: Low Byte Nutzwort 2 (04H)
 HB NW2: High Byte Nutzwort 2 (03H)

VE–Daten

Laufzeit: 07 KT 92 / 07 KT 93
 Grundlaufzeit: (21 + 22 * #JOB) µs
 Zusätzliche Laufzeit: (106...360 * #JOB) µs
 Aktualisierung der Ausgänge: ja
 Anzahl der Vergangenheitswerte: #JOB Worte
 Verfügbar ab: ABB Procontic CS31, 07 KT 92 R262, 07 KT 93 R171

Die 1–0–Flanke des Eingangs 0 HT wird um die Zeitdauer ZD verzögert und als 1–0–Flanke am Ausgang Q ausgegeben.

Geht der Eingang 0 HT vor Ablauf der Zeit ZD wieder auf 1–Pegel, so bleibt der Ausgang Q auf 1–Pegel.



Parameter

0 HT	BINÄR	E, M, A, S, K	Eingangssignal
ZD	DOPPELWORT	MD, KD	Verzögerungszeit
Q	BINÄR	M, A	Verzögertes Signal

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	45 µs; Höchstlaufzeit 541 µs (max), bei Flanken am Eingang 0 HT	97 µs / 551 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	1 Wort	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Die 1–0–Flanke des Eingangs 0 HT wird um die Zeitdauer ZD verzögert und als 1–0–Flanke am Ausgang Q ausgegeben.

Geht der Eingang 0 HT vor Ablauf der Zeit ZD wieder auf 1–Pegel, so bleibt der Ausgang Q auf 1–Pegel.

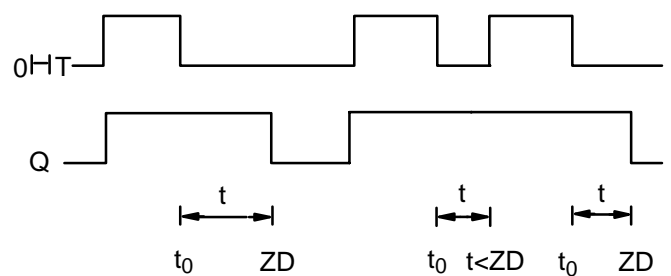
Die Zeitangabe erfolgt in Millisekunden, wobei nur ganzzahlige Vielfache von 5 ms zulässig sind (Bsp.: 5 ms, 10 ms, 500 ms, 100 000 ms ...). Angebbare Zeitbereich: 5 ms ... 24,8 Tage.

Bei der indirekten Konstanten KD wird der Wert der Verzögerungszeit im Langtext angegeben.

Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit

Sinnvoller Bereich für ZD: > 1 Zykluszeit.

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar.



Allgemeines Verhalten

- Gestartete Zeitwerke werden vom Betriebssystem der SPS bearbeitet und sind deshalb vollkommen unabhängig von der Bearbeitung des SPS-Programms. Erst nach Ablauf des Zeitwerks erfolgt eine entsprechende Meldung des Betriebssystems an den zugehörigen Zeitbaustein im SPS-Programm.
- Die Bearbeitung eines Zeitwerks im Betriebssystem der SPS wird durch folgende Befehle *nicht* beeinflusst:
 - Programm abbrechen
 - Programm starten
 - Programm stoppen
 - Programm fortsetzen

D.h. ist ein Zeitwerk gestartet, so wird es im Betriebssystem der SPS auch dann weiter bearbeitet, wenn das zugehörige SPS-Programm abgebrochen, wieder gestartet oder gestoppt und wieder fortgesetzt wird.

Initialisierung

Die Zeitwerke werden grundsätzlich bei jedem Kalt- oder Warmstart der SPS initialisiert. Ein laufendes Zeitwerk wird also stets durch einen Kalt- oder Warmstart der SPS abgebrochen.

Kaltstart:

- Befehl KALT <CR>
- erstmaliges Zuschalten der Spannung

Warmstart:

- Befehl WARM <CR>
- Zuschalten der Spannung
- RESET-Schalter

Wichtiger Hinweis

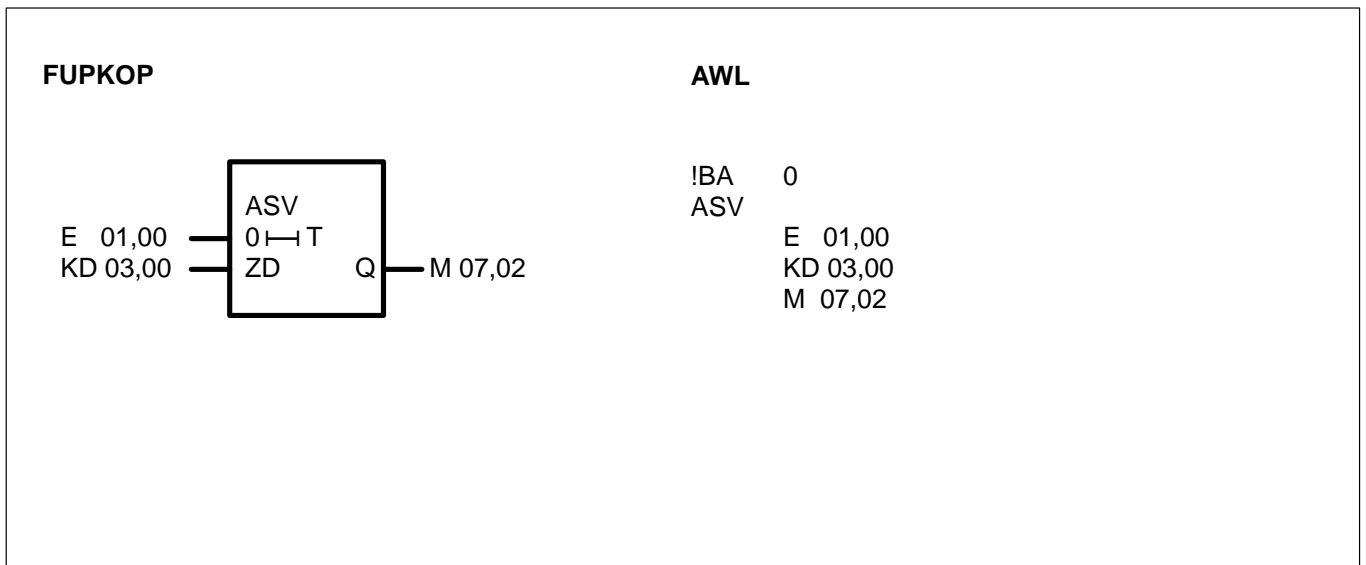
Das Verhalten der Zeitwerke ist geändert worden. Alle laufenden Zeitwerke werden bei folgenden Aktionen automatisch gestoppt:

- SPS-Programm abbrechen
- RUN/STOP-Schalter von RUN -> STOP

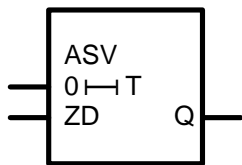
Diese Änderung gilt ab:

07 KR 91 Index g
07 KT 92 Index g
07 KT 93

Beispiel



VE-FUP-Definition



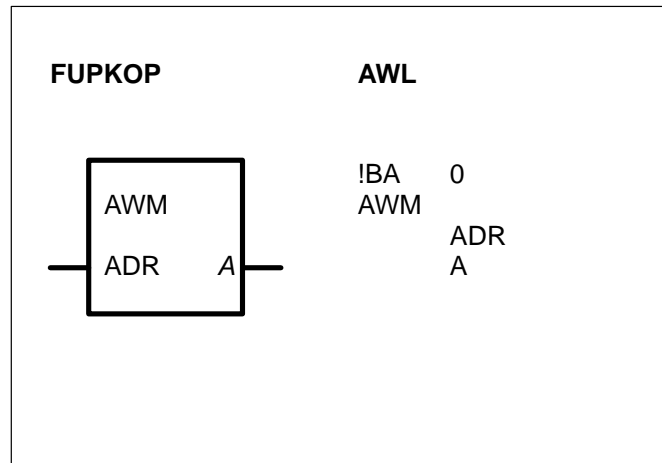
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0 I-T	E	L	N	P	J	0	0
ZD	E	D	N	P	J	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	ASV			
00002		PP 0	0 I-T	Eingang BINÄR (Starte Zeit)
00003		PP 0	ZD	Eingang DOPPELWORT (Zeitwert)
00004		PP 0	Q	Ausgang BINÄR (Zeit)

Der Funktionsbaustein liest den Wert eines Operanden, wobei das Verfahren der indirekten Adressierung benutzt wird. Der gelesene Wert wird dem Ausgang A zugewiesen.

Hinweis: Der Baustein AWM kann nur sinnvoll in Verbindung mit dem Baustein ADRWA benutzt werden.



Parameter

ADR	WORT	EW, AW, MW, KW	Indirekte Adresse des zu lesenden Operanden
A	WORT	AW, MW	Wert des gelesenen Operanden

VE-Daten

Laufzeit:		
Grundlaufzeit:		26 µs
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein liest den Wert eines Operanden, wobei das Verfahren der indirekten Adressierung benutzt wird. Der gelesene Wert wird dem Ausgang A zugewiesen.

Hinweis: Der Baustein AWM kann nur sinnvoll in Verbindung mit dem Baustein ADRWA benutzt werden.

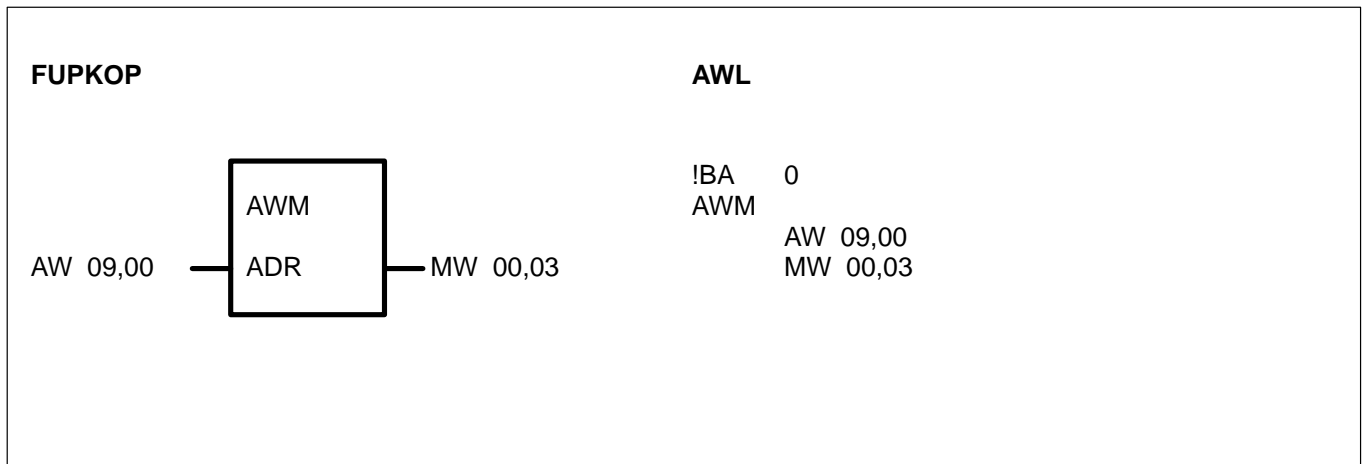
Der Wert des Operanden am Eingang ADR wird als Adresse des zu lesenden Operanden interpretiert (indirekte Adressierung).

Der Operand am Eingang ADR stellt also zusammen mit seinem Wert eine indirekte Adresse dar. Diese indirekte Adresse wird vom Funktionsbaustein ADRWA erzeugt.

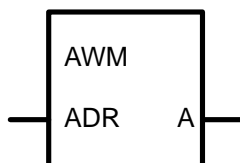
Der Ein- und der Ausgang ist weder doppelbar noch negierbar.

Hinweis: Die Erläuterung des Verfahrens der indirekten Adressierung und die Verwendungsmöglichkeit des Funktionsbausteins AWM ist beim Funktionsbaustein ADRWA beschrieben.

Beispiel



VE-FUP-Definition



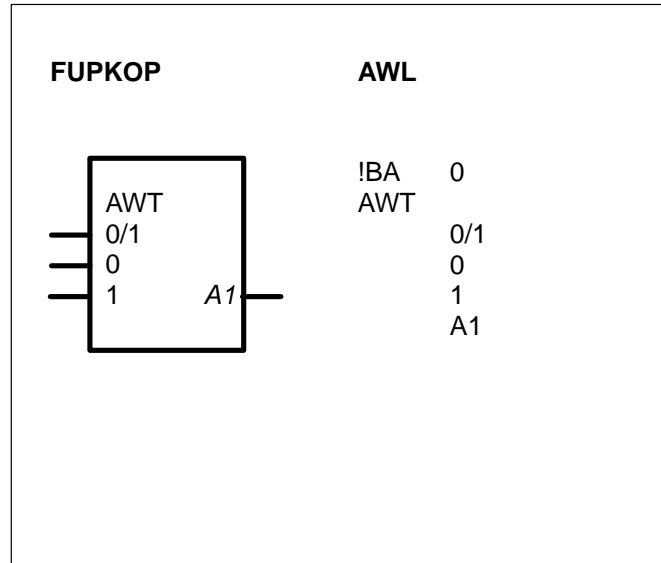
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
ADR	E	W	N	P	J	0	0
A	A	W	N	P	N	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	AWM			
00002		PP 0	ADR	Eingang WORT (Operandenadresse)
00003		PP 0	A	Ausgang WORT (Operandenwert)

0-Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang A1 den Wert des Wortoperanden am Eingang 0 zu.

1-Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang A1 den Wert des Wortoperanden am Eingang 1 zu.



Parameter

0/1	BINÄR	E, M, A, S, K	Umschalteingang
0	WORT	EW, MW, AW, KW	Worteingang für 0/1 = 0
1	WORT	EW, MW, AW, KW	Worteingang für 0/1 = 1
A1	WORT	AW, MW	Wortausgang

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	30 µs	40 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

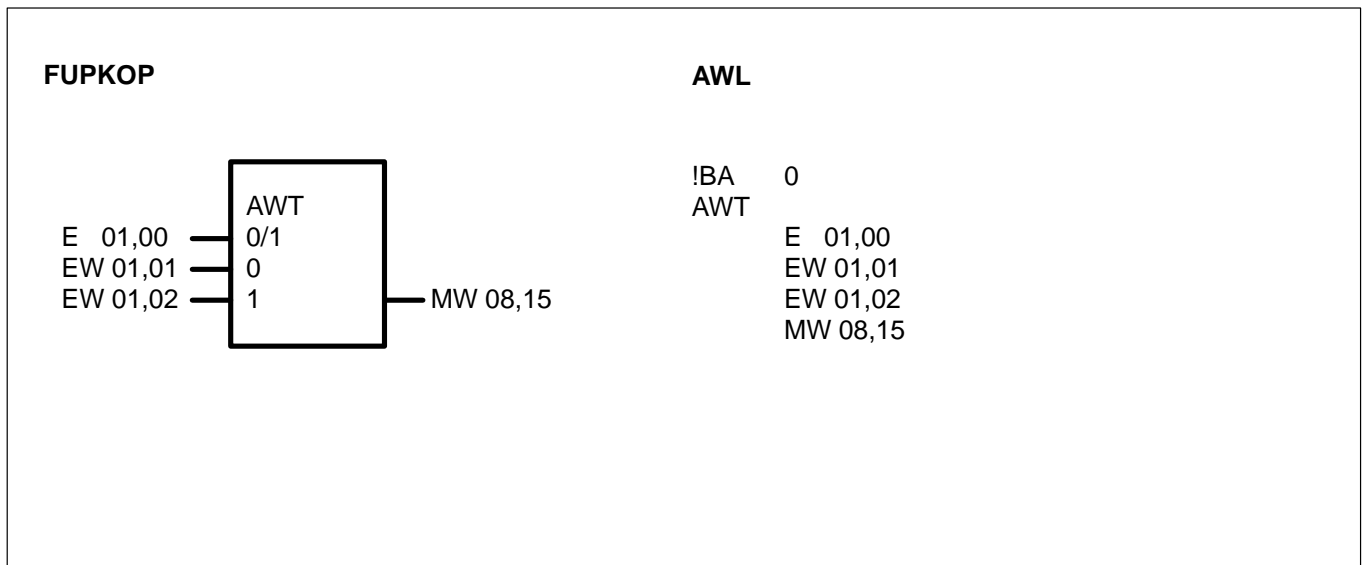
Beschreibung

0-Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang A1 den Wert des Wortoperanden am Eingang 0 zu.

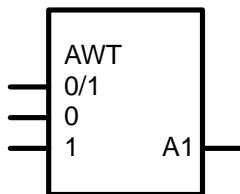
1-Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang A1 den Wert des Wortoperanden am Eingang 1 zu.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar/invertierbar.

Beispiel



VE-FUP-Definition



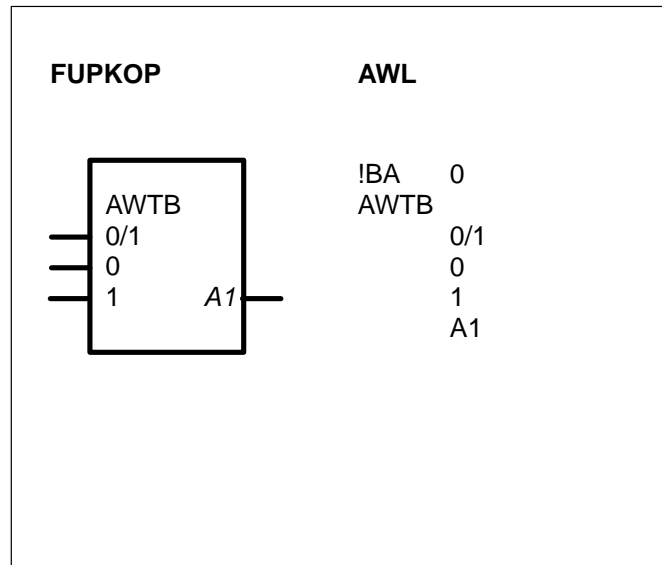
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0/1	E	L	N	P	J	0	0
0	E	W	N	P	J	0	0
1	E	W	N	P	J	0	0
A1	A	W	N	P	N	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	AWT			
00002		PP 0	0/1	Eingang BINÄR (Selektor)
00003		PP 0	0	Eingang WORT
00004		PP 0	1	Eingang WORT
00005		PP 0	A1	Ausgang WORT

0–Signal am binären Eingang 0/1 weist dem Operanden am Ausgang A1 den Zustand des Operanden am Eingang 0 zu.

1–Signal am binären Eingang 0/1 weist dem Operanden am Ausgang A1 den Zustand des Operanden am Eingang 1 zu.



Parameter

0/1	BINÄR	E, M, A, S, K	Umschalteingang
0	BINÄR	E, M, A, S, K	Binäreingang für 0/1 = 0
1	BINÄR	E, M, A, S, K	Binäreingang für 0/1 = 1
A1	BINÄR	A, M	Binärausgang

VE–Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

Zusätzliche Laufzeit:

Aktualisierung der Ausgänge:

Anzahl der Vergangenheitswerte:

Verfügbar ab:

32 µs

—

ja

keine

ABB Procontic CS31 / 907 PC 331

74 µs

—

ja

keine

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

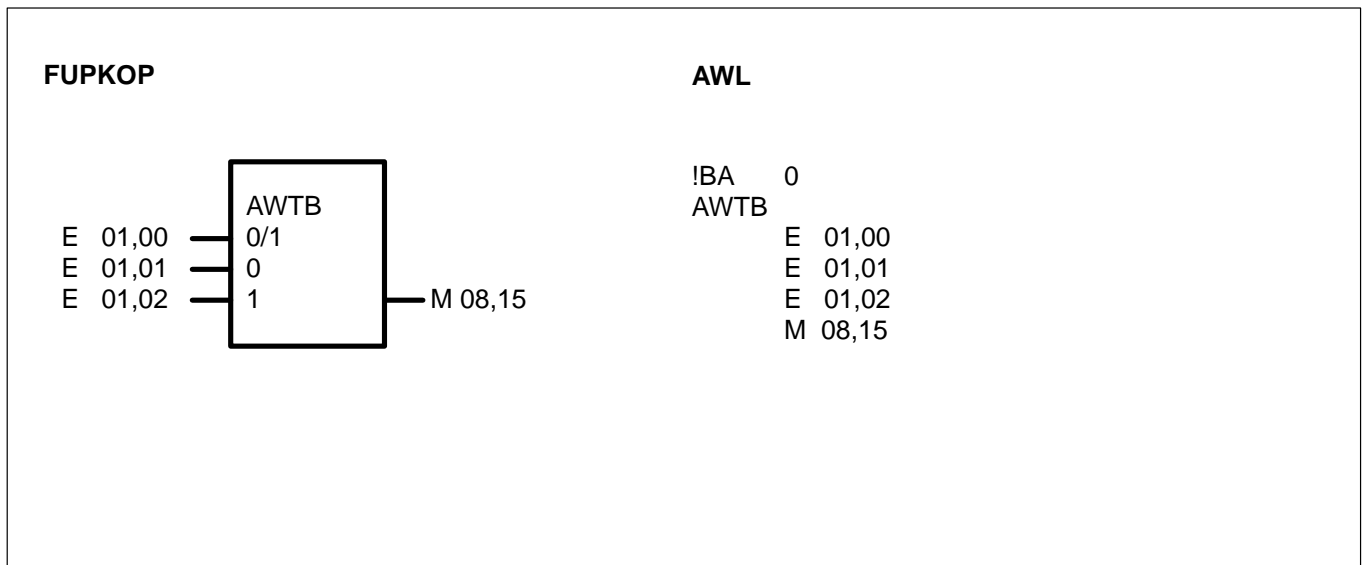
Beschreibung

0–Signal am binären Eingang 0/1 weist dem Operanden am Ausgang A1 den Zustand des Operanden am Eingang 0 zu.

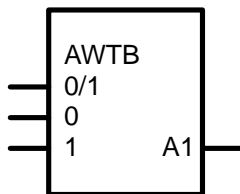
1–Signal am binären Eingang 0/1 weist dem Operanden am Ausgang A1 den Zustand des Operanden am Eingang 1 zu.

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar.

Beispiel



VE-FUP-Definition



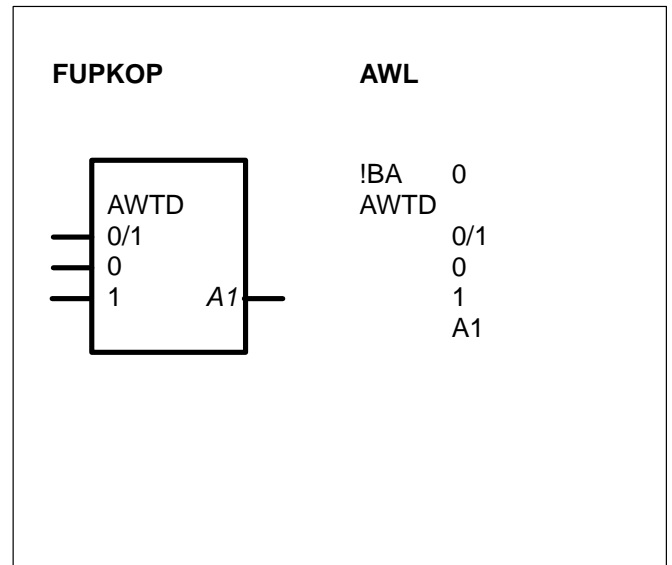
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0/1	E	L	N	P	J	0	0
0	E	L	N	P	J	0	0
1	E	L	N	P	J	0	0
A1	A	L	N	P	N	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	AWTB			
00002		PP 0	0/1	Eingang BINÄR (Selektor)
00003		PP 0	0	Eingang BINÄR
00004		PP 0	1	Eingang BINÄR
00005		PP 0	A1	Ausgang BINÄR

0–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang A1 den Wert des Doppelwortoperanden am Eingang 0 zu.

1–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang A1 den Wert des Doppelwortoperanden am Eingang 1 zu.



Parameter

0/1	BINÄR	E, M, A, S, K	Umschalteingang
0	DOPPELWORT	MD, KD	Doppelworteingang für 0/1 = 0
1	DOPPELWORT	MD, KD	Doppelworteingang für 0/1 = 1
A1	DOPPELWORT	MD	Doppelwortausgang

VE-DATEN

Laufzeit:		
Grundlaufzeit:		41 µs
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

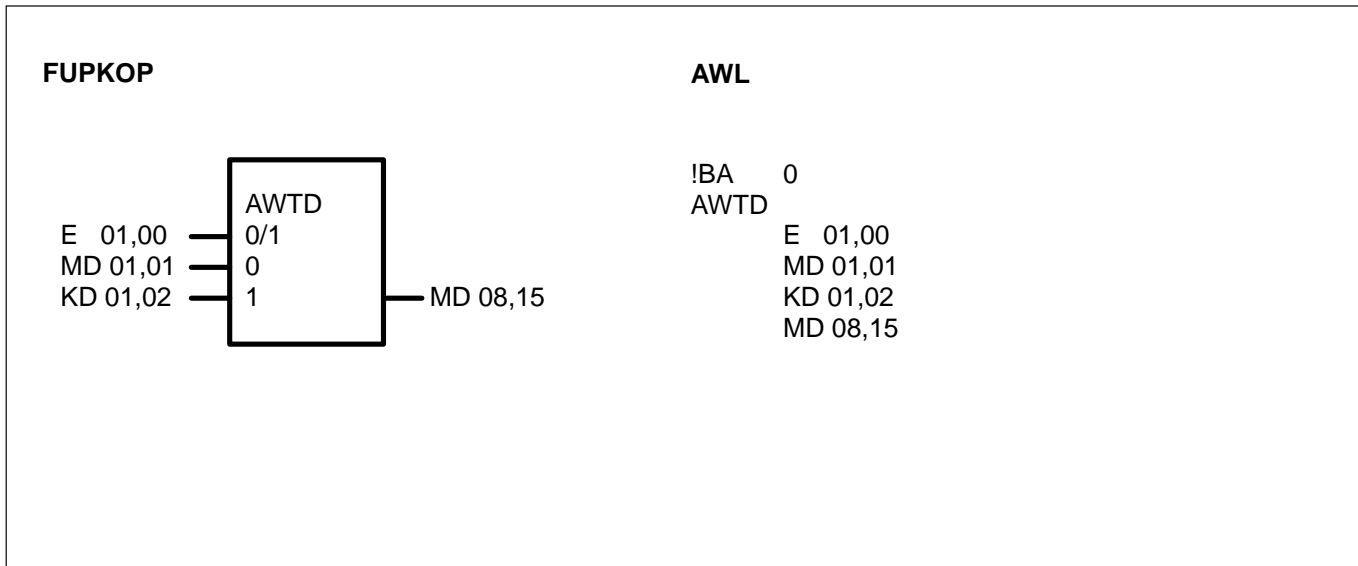
Beschreibung

0–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang A1 den Wert des Doppelwortoperanden am Eingang 0 zu.

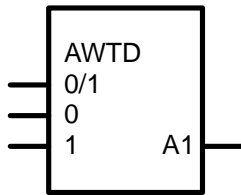
1–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang A1 den Wert des Doppelwortoperanden am Eingang 1 zu.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar/invertierbar.

Beispiel



VE-FUP-Definition

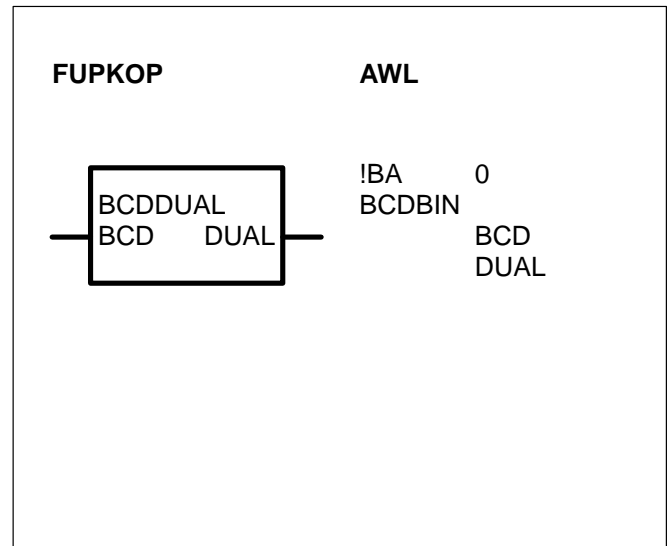


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0/1	E	L	N	P	J	0	0
0	E	D	N	P	J	0	0
1	E	D	N	P	J	0	0
A1	A	D	N	P	N	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	AWTD			
00002		PP 0	0/1	Eingang BINÄR (Selektor)
00003		PP 0	0	Eingang DOPPELWORT
00004		PP 0	1	Eingang DOPPELWORT
00005		PP 0	A1	Ausgang DOPPELWORT

Die positive, BCD-codierte Zahl am Eingang BCD wird in eine Dualzahl gewandelt und dem Operanden am Ausgang DUAL zugewiesen.



Parameter

BCD	WORT	EW, MW, AW, KW	BCD-codierte Zahl
DUAL	WORT	AW, MW	Dualzahl

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

88 µs

365 µs

Zusätzliche Laufzeit:

—

—

Aktualisierung der Ausgänge:

ja

ja

Anzahl der Vergangenheitswerte:

keine

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

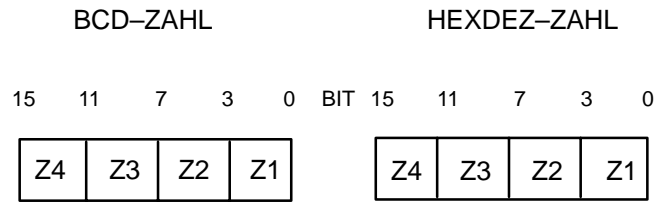
Beschreibung

Die positive, BCD-codierte Zahl am Eingang BCD wird in eine Dualzahl gewandelt und dem Operanden am Ausgang DUAL zugewiesen.

Der Ein- und Ausgang ist weder doppelbar noch negierbar.

Definition:

Die Wertigkeit der Ziffern einer BCD-codierten Zahl und einer Hexadezimal-Zahl sind wie folgt definiert:



Zahlenwert:

$$\begin{aligned} Z1 &* 1 \\ Z2 &* 10 \\ Z3 &* 100 \\ Z4 &* 1000 \end{aligned}$$

$$0 \leq Zi \leq 9$$

Zahlenwert:

$$\begin{aligned} Z1 &* 1 \\ Z2 &* 16 \\ Z3 &* 256 \\ Z4 &* 4096 \end{aligned}$$

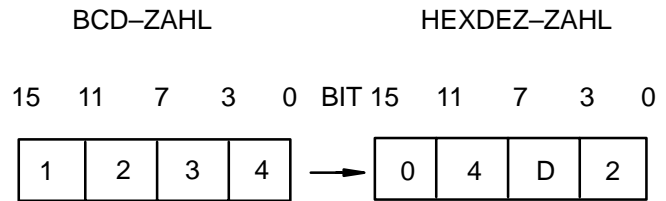
$$0 \leq Zi \leq F$$

Anmerkung:

Der Baustein akzeptiert zusätzlich am BCD-Eingang auch Ziffern für die gilt:

$$0 \leq Zi \leq F$$

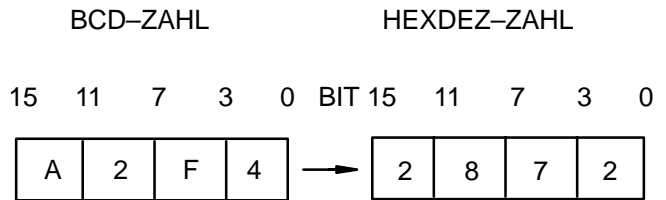
Beispiel 1



$$\begin{aligned} Z1 &= 4 * 1 = 4 \\ Z2 &= 3 * 10 = 30 \\ Z3 &= 2 * 100 = 200 \\ Z4 &= 1 * 1000 = 1000 \\ \hline &+ 1234 \end{aligned}$$

$$\begin{aligned} Z1 &= 2 * 1 = 2 \\ Z2 &= 13 * 16 = 208 \\ Z3 &= 4 * 256 = 1024 \\ Z4 &= - * - = - \\ \hline &+ 1234 \end{aligned}$$

Beispiel 2



$$\begin{aligned} Z1 &= 4 * 1 = 4 \\ Z2 &= 15 * 10 = 150 \\ Z3 &= 2 * 100 = 200 \\ Z4 &= 10 * 1000 = 10000 \\ \hline &+ 10354 \end{aligned}$$

$$\begin{aligned} Z1 &= 2 * 1 = 2 \\ Z2 &= 7 * 16 = 112 \\ Z3 &= 8 * 256 = 2048 \\ Z4 &= 2 * 4096 = 8192 \\ \hline &+ 10354 \end{aligned}$$

Darstellung einer negativen BCD-Zahl

Eine negative BCD-Zahl kann in der SPS durch *getrennte* Darstellung des Wertes und des Vorzeichens dargestellt werden. Dabei wird der Wert der BCD-Zahl in einer Wort-Variablen und die Information über das Vorzeichen in einer Binär-Variablen abgelegt.

Beispiel:

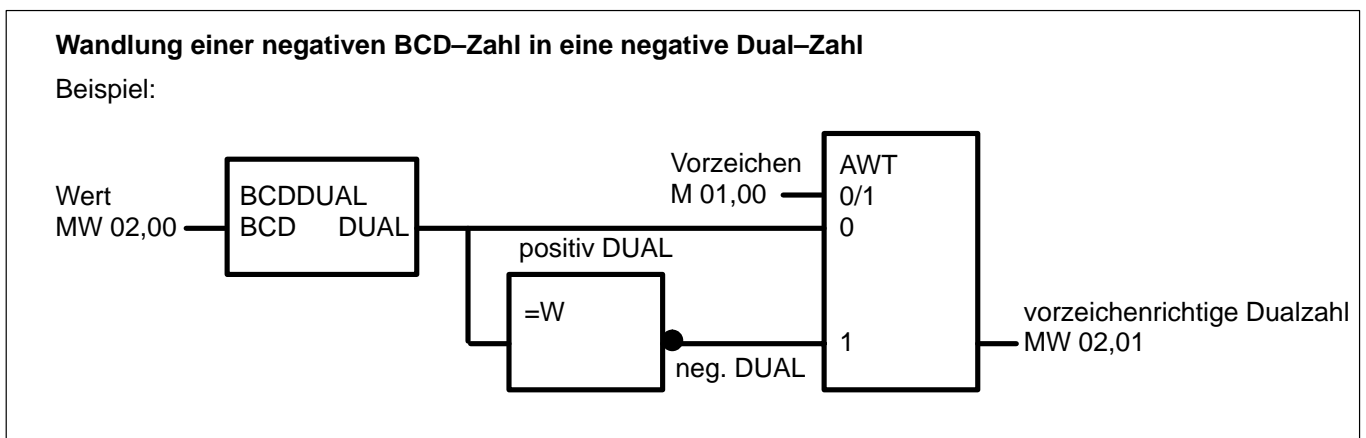
Darstellung der BCD-Zahl -7:

MW 02,00 = 7 Wert der Zahl BCD-codiert

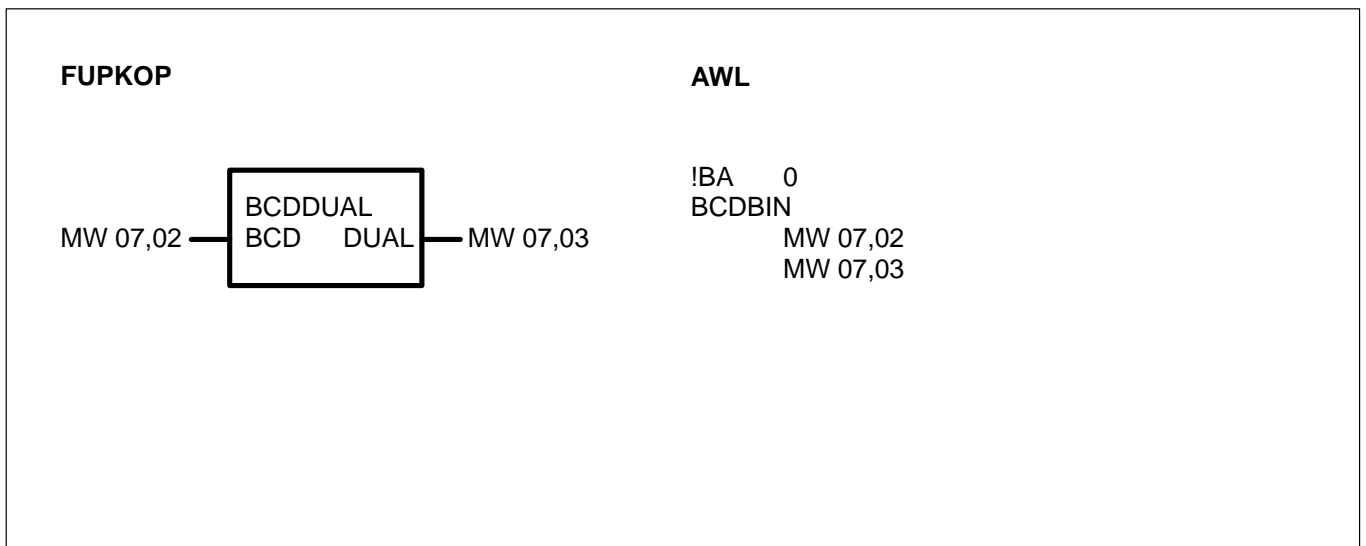
M 01,00 = 1 Vorzeichen der Zahl

- positiv: 0

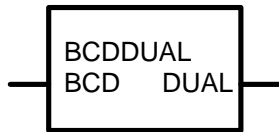
- negativ: 1



Beispiel



VE-FUP-Definition

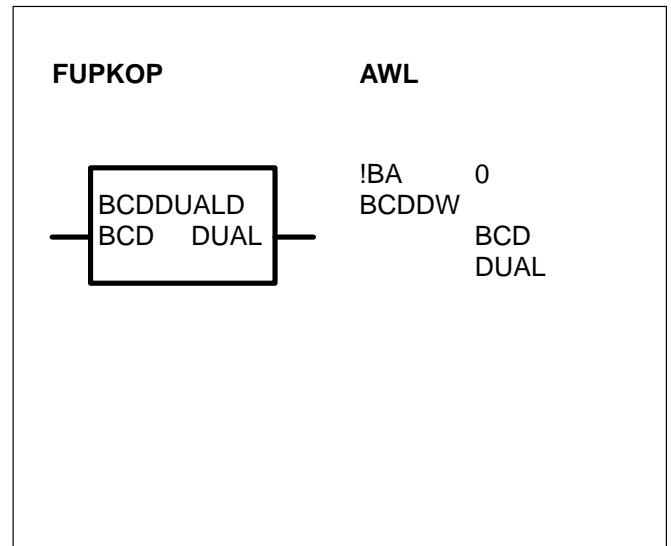


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
BCD	E	W	N	P	J	0	0
DUAL	A	W	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	BCDBIN			
00002		PP 0	BCD	Eingang WORT (BCD-Zahl)
00003		PP 0	DUAL	Ausgang WORT (Dualzahl)

Die positive, BCD-codierte Zahl am Eingang BCD wird in eine Dualzahl gewandelt und dem Operanden am Ausgang DUAL zugewiesen.



Parameter

BCD	DOPPELWORT MD, KD	BCD-codierte Zahl
DUAL	DOPPELWORT MD	Dualzahl

VE-Daten

Laufzeit:	
Grundlaufzeit:	310 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Die positive, BCD-codierte Zahl am Eingang BCD wird in eine Dualzahl gewandelt und dem Operanden am Ausgang DUAL zugewiesen.

Der Eingang und der Ausgang sind weder doppelbar noch negierbar.

Definition

Die Wertigkeit der Ziffern einer BCD-codierten Zahl und einer Hexadezimal-Zahl sind wie folgt definiert:

Anmerkung:

Der Baustein akzeptiert am BCD-Eingang auch Ziffern für die gilt:

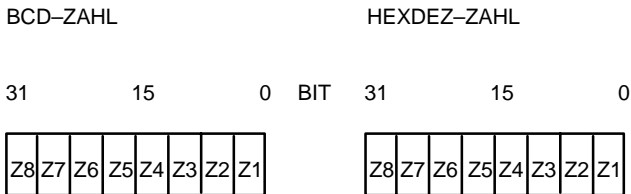
$$0 \leq Z_i \leq F$$

Darstellung einer negativen BCD-Zahl

siehe Funktionsbaustein BCDDUAL

Wandlung einer negativen BCD-Zahl in eine negative Dual-Zahl

siehe Funktionsbaustein BCDDUAL

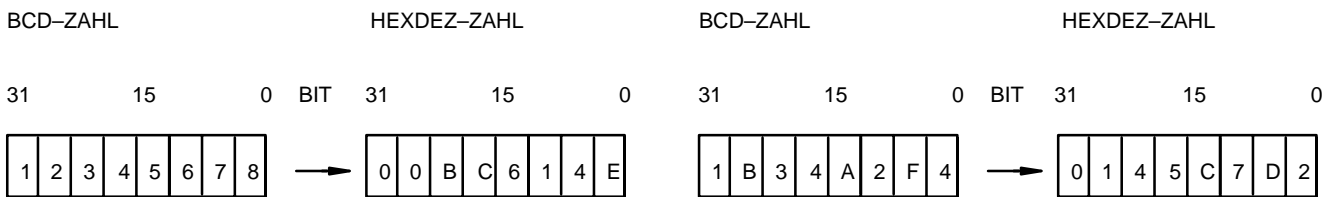


<p>Zahlenwert:</p> <p>Z1 * 1</p> <p>Z2 * 10</p> <p>Z3 * 100</p> <p>Z4 * 1000</p> <p>Z5 * 10000</p> <p>Z6 * 100000</p> <p>Z7 * 1000000</p> <p>Z8 * 10000000</p>	<p>Zahlenwert:</p> <p>Z1 * 1</p> <p>Z2 * 16</p> <p>Z3 * 256</p> <p>Z4 * 4096</p> <p>Z5 * 65536</p> <p>Z6 * 1048576</p> <p>Z7 * 16777216</p> <p>Z8 * 268435456</p>
--	---

$0 \leq Z_i \leq 9$	$0 \leq Z_i \leq F$
---------------------	---------------------

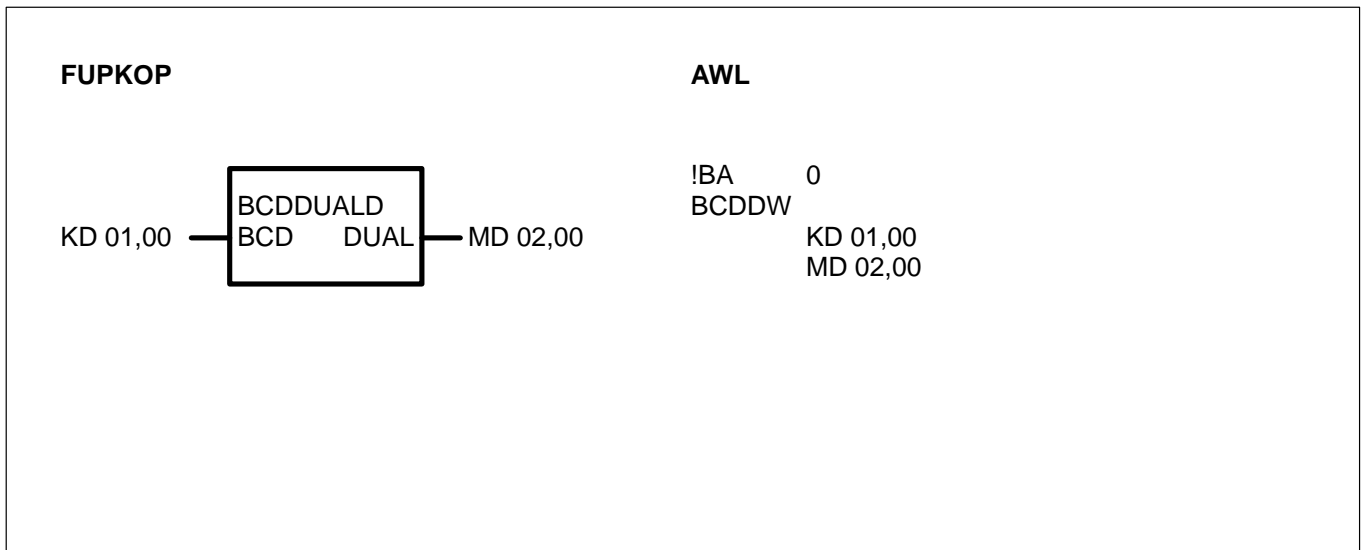
Beispiel 1:

Beispiel 2:

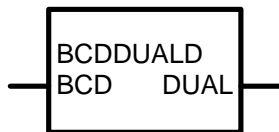


<p>Z1 = 8 * 1 = 8</p> <p>Z2 = 7 * 10 = 70</p> <p>Z3 = 6 * 100 = 600</p> <p>Z4 = 5 * 1000 = 5000</p> <p>Z5 = 4 * 10000 = 40000</p> <p>Z6 = 3 * 100000 = 300000</p> <p>Z7 = 2 * 1000000 = 2000000</p> <p>Z8 = 1 * 10000000 = 10000000</p>	<p>Z1 = 14 * 1 = 14</p> <p>Z2 = 4 * 16 = 64</p> <p>Z3 = 1 * 256 = 256</p> <p>Z4 = 6 * 4096 = 24576</p> <p>Z5 = 12 * 65536 = 786432</p> <p>Z6 = 11 * 1048576 = 11534336</p> <p>Z7 = 0 * 16777216 = 0</p> <p>Z8 = 0 * 268435456 = 0</p>	<p>Z1 = 4 * 1 = 4</p> <p>Z2 = 15 * 10 = 150</p> <p>Z3 = 2 * 100 = 200</p> <p>Z4 = 10 * 1000 = 10000</p> <p>Z5 = 4 * 10000 = 40000</p> <p>Z6 = 3 * 100000 = 300000</p> <p>Z7 = 11 * 1000000 = 11000000</p> <p>Z8 = 1 * 10000000 = 10000000</p>	<p>Z1 = 2 * 1 = 2</p> <p>Z2 = 13 * 16 = 208</p> <p>Z3 = 7 * 256 = 1792</p> <p>Z4 = 12 * 4096 = 49152</p> <p>Z5 = 5 * 65536 = 327680</p> <p>Z6 = 4 * 1048576 = 4194304</p> <p>Z7 = 1 * 16777216 = 16777216</p> <p>Z8 = 0 * 268435456 = 0</p>
+ 12345678	+ 12345678	+ 21350354	+ 21350354

Beispiel



VE-FUP-Definition

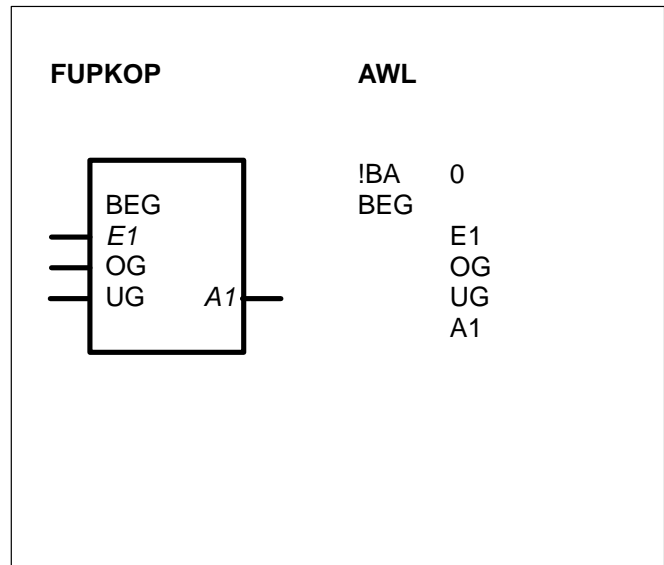


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
BCD	E	D	N	P	J	0	0
DUAL	A	D	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	BCDDW			
00002		PP 0	BCD	Eingang DOPPELWORT (BCD-Zahl)
00003		PP 0	DUAL	Ausgang DOPPELWORT (Dualzahl)

Der Wert des Operanden am Eingang E1 wird auf den Bereich zwischen der oberen und der unteren Grenze begrenzt.



Parameter

E1	WORT	EW, MW, AW, KW	Eingangswert
OG	WORT	EW, MW, AW, KW	Obere Grenze
UG	WORT	EW, MW, AW, KW	Untere Grenze
A1	WORT	AW, MW	Begrenzter Wert

VE-Daten

07 KR 91 / 07 KT 92 / 07 KT 93

07 KR 31

Laufzeit:		
Grundlaufzeit:	32...35 µs	147 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0

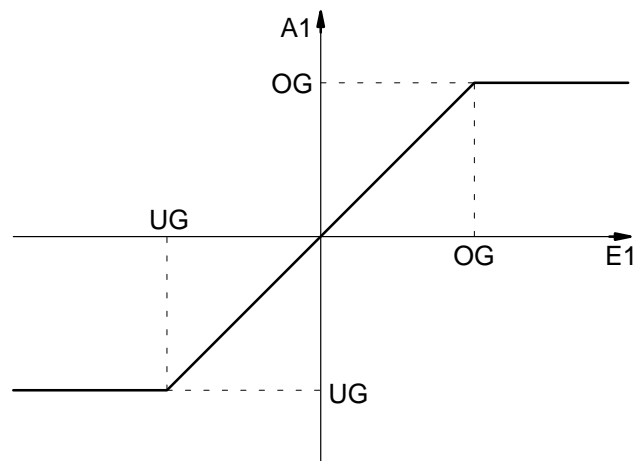
Beschreibung

Der Wert des Operanden am Eingang E1 wird auf den Bereich zwischen der oberen und der unteren Grenze begrenzt.

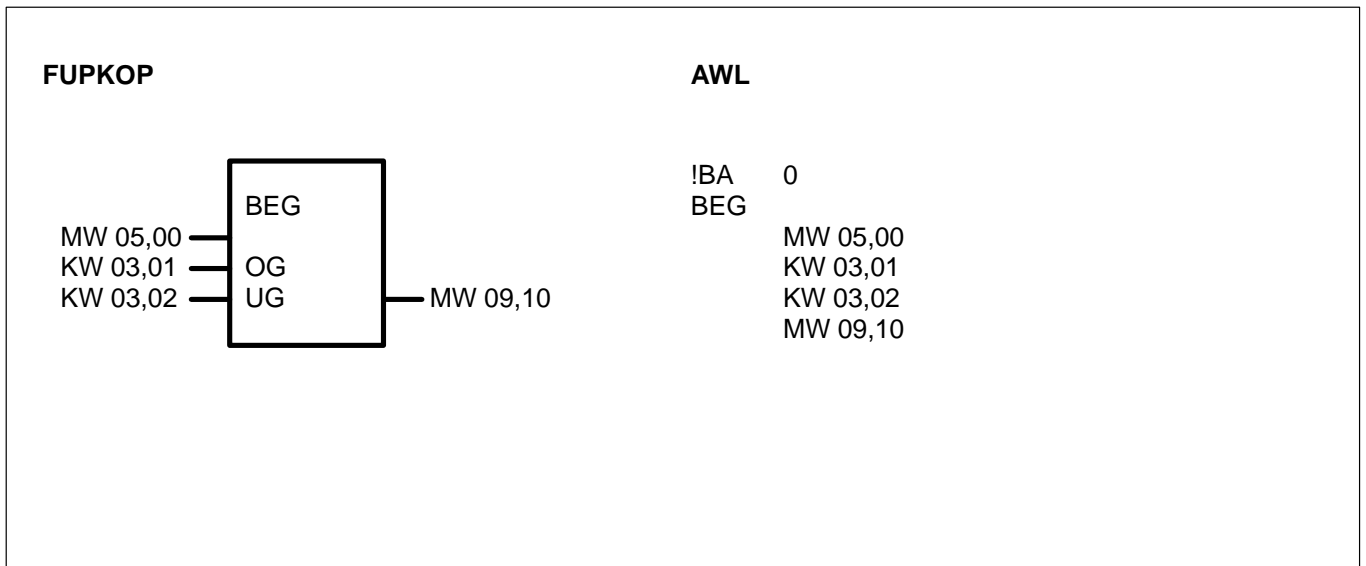
Die obere Grenze wird durch den Operanden am Eingang OG, die untere Grenze durch den Operanden am Eingang UG vorgegeben.

Es gilt: $A1 = UG$ für $E1 \leq UG$
 $A1 = E1$ für $UG \leq E1 \leq OG$
 $A1 = OG$ für $E1 \geq OG$

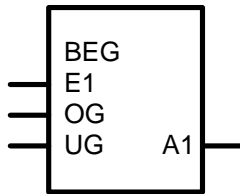
Die Eingänge und der Ausgang sind weder doppelbar noch negierbar.



Beispiel



VE-FUP-Definition

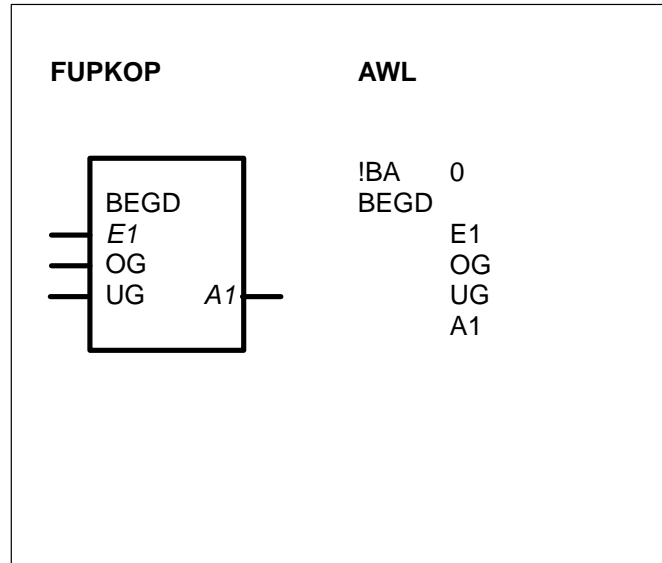


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
OG	E	W	N	P	J	0	0
UG	E	W	N	P	J	0	0
A1	A	W	N	P	N	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	BEG			
00002		PP 0	E1	Eingang WORT (Eingangswert)
00003		PP 0	OG	Eingang WORT (Obere Grenze)
00004		PP 0	UG	Eingang WORT (Untere Grenze)
00005		PP 0	A1	Ausgang WORT(Begrenzter Wert)

Der Wert des Operanden am Eingang E1 wird auf den Bereich zwischen der oberen und der unteren Grenze begrenzt.



Parameter

E1	DOPPELWORT	MD,	Eingangswert
OG	DOPPELWORT	MD, KD	Obere Grenze
UG	DOPPELWORT	MD, KD	Untere Grenze
A1	DOPPELWORT	MD	Begrenzter Wert

VE-DATEN

Laufzeit:		
Grundlaufzeit:	30 ... 38 μ s	
Zusätzliche Laufzeit:	—	
Aktualisierung der Ausgänge:	ja	
Anzahl der Vergangenheitswerte:	keine	
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	

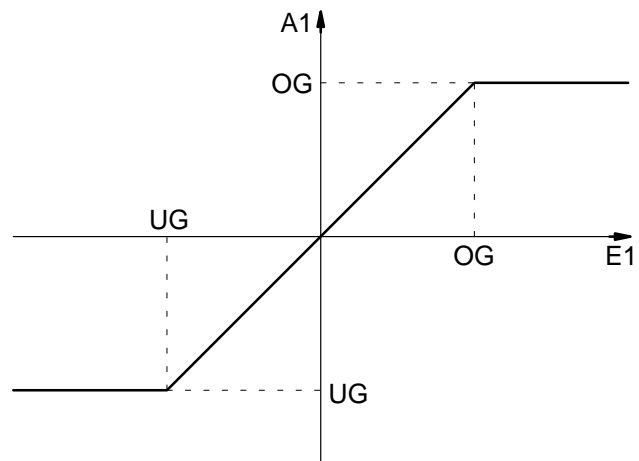
Beschreibung

Der Wert des Operanden am Eingang E1 wird auf den Bereich zwischen der oberen und der unteren Grenze begrenzt.

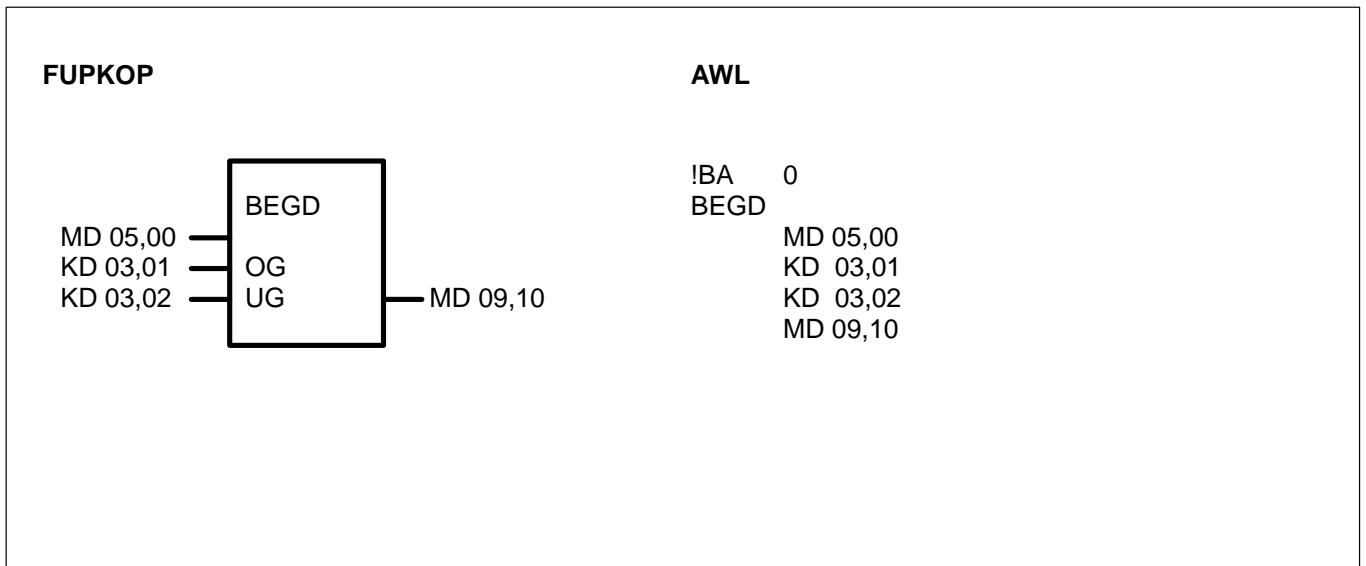
Die obere Grenze wird durch den Operanden am Eingang OG, die untere Grenze durch den Operanden am Eingang UG vorgegeben.

Es gilt: $A1 = UG$ für $E1 \leq UG$
 $A1 = E1$ für $UG \leq E1 \leq OG$
 $A1 = OG$ für $E1 \geq OG$

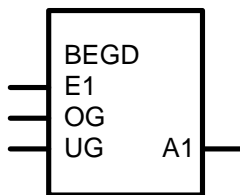
Die Eingänge und der Ausgang sind weder doppelbar noch negierbar.



Beispiel



VE-FUP-Definition

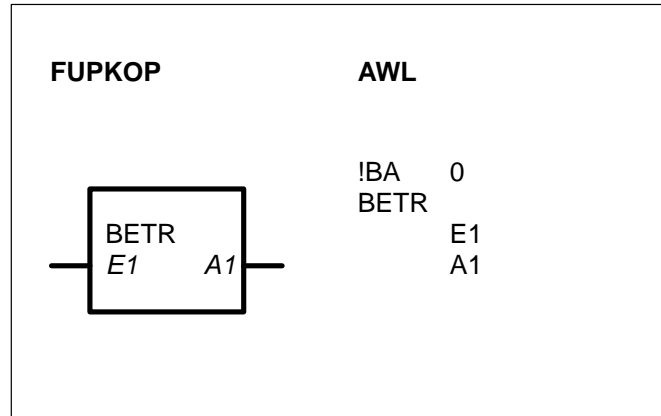


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
OG	E	D	N	P	J	0	0
UG	E	D	N	P	J	0	0
A1	A	D	N	P	N	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	BEGD			
00002		PP 0	E1	Eingang DOPPELWORT (Eingangswert)
00003		PP 0	OG	Eingang DOPPELWORT (Obere Grenze)
00004		PP 0	UG	Eingang DOPPELWORT (Untere Grenze)
00005		PP 0	A1	Ausgang DOPPELWORT (Begrenzter Wert)

Vom Wert des Wortoperanden am Eingang E1 wird der Betrag gebildet und dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	WORT	EW, MW, AW, KW	Eingangswert
A1	WORT	MW, AW	Betrag des Eingangswerts

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	28 µs	42 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Vom Wert des Wortoperanden am Eingang E1 wird der Betrag gebildet und dem Operanden am Ausgang A1 zugewiesen.

Der Eingang und der Ausgang sind weder doppelbar noch negierbar.

Zahlenbereich

Integer Wort (16 Bit).

Speziell gilt hier für den Eingang E1:

Liegt am Eingang E1 der unzulässige Wert 8000_H (−32768) an, so wird dem Ausgang A1 der maximal mögliche Wert 7FFF_H (+32767) zugewiesen.

Allgemein gilt:

- untere Grenze: 8001_H −32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (−32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

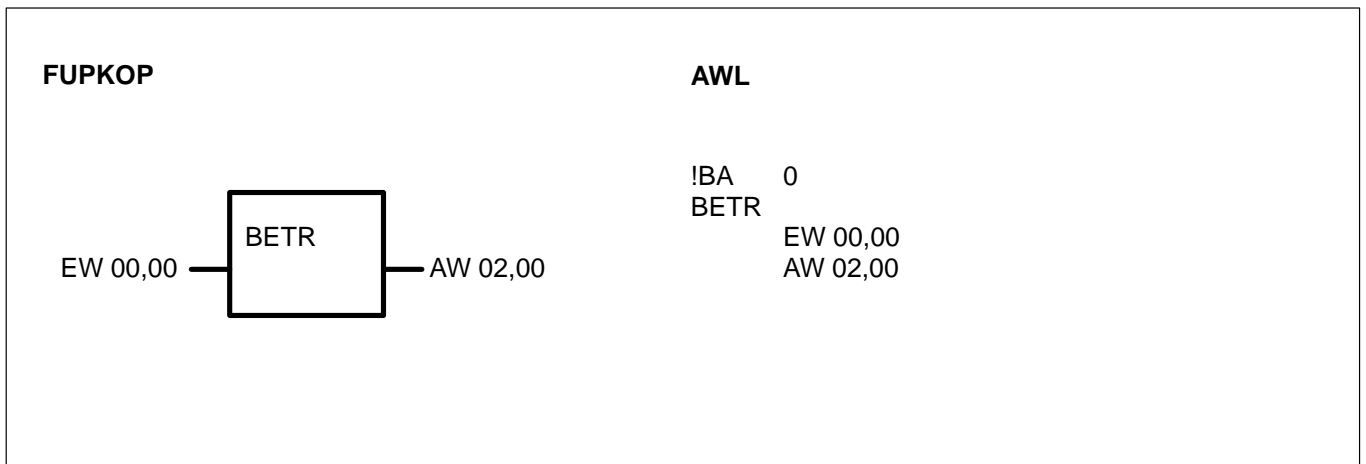
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf keinen Fall eine Negation oder Subtraktion angewendet werden.

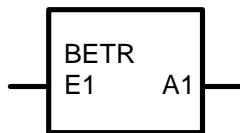
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der verbotene Wert 8000_H (−32768) auf den erlaubten Wert 8001_H (−32767) korrigiert.

Beispiel



VE-FUP-Definition

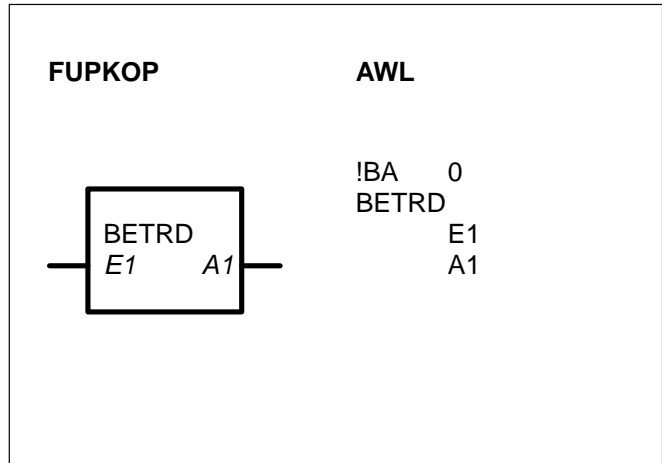


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
A1	A	W	N	P	N	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	BETR			
00002		PP 0	E1	Eingang WORT
00003		PP 0	A1	Ausgang WORT

Vom Wert des Operanden am Eingang E1 wird der Betrag gebildet und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	DOPPELWORT MD, KD	Eingangswert
A1	DOPPELWORT MD	Betrag des Eingangswerts

VE-Daten

Laufzeit:		
Grundlaufzeit:	32 ... 36 µs	
Zusätzliche Laufzeit:	—	
Aktualisierung der Ausgänge:	ja	
Anzahl der Vergangenheitswerte:	keine	
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	

Beschreibung

Vom Wert des Operanden am Eingang E1 wird der Betrag gebildet und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Liegt am Eingang E1 aus irgendeinem Grund der unzulässige Wert 8000 0000 H (−2 147 483 648) an, so wird dem Ausgang A1 der Wert 7FFF FFFF_H (+2 147 483 697) zugewiesen. Der unzulässige Wert 8000 0000_H wird also zunächst auf den zulässigen Wert 8000 0001_H korrigiert und dann erst wird der Betrag gebildet.

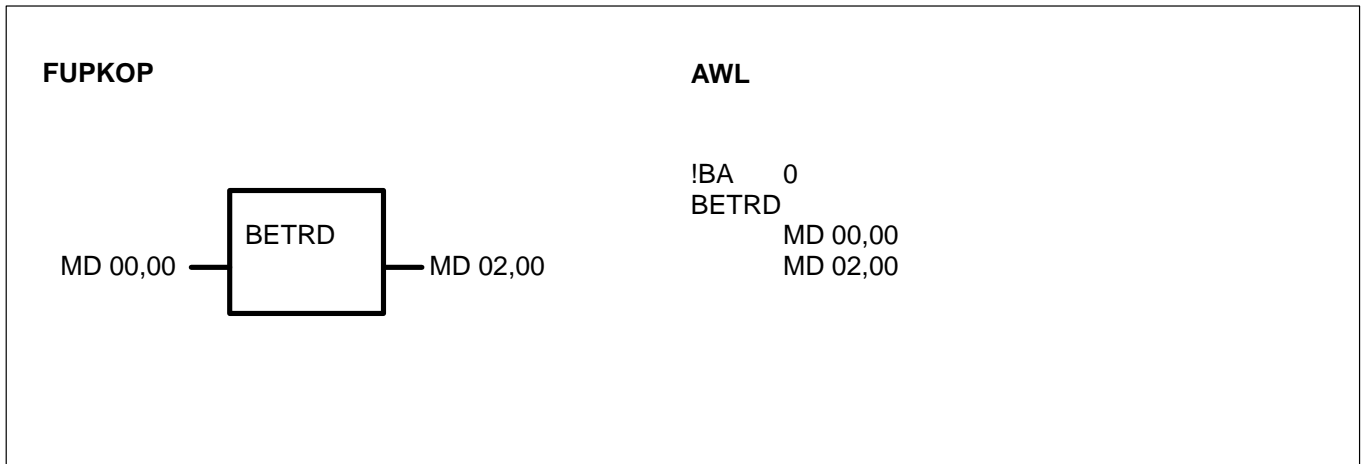
Der Eingang und der Ausgang sind weder doppelbar noch negierbar.

Zahlenbereich

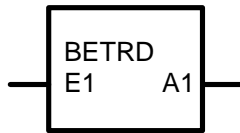
Integer Doppelwort (32 Bit)

– untere Grenze:	8000 0001 _H	−2 147 483 647
– obere Grenze:	7FFF FFFF _H	+2 147 483 647
– unzulässiger Wert:	8000 0000 _H	—

Beispiel



VE-FUP-Definition

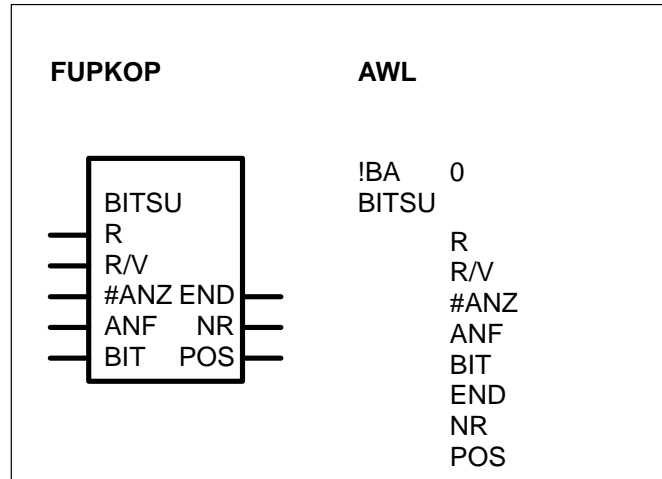


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
A1	A	D	N	P	N	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	BETRD			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	A1	Ausgang DOPPELWORT

Der Funktionsbaustein durchsucht ein Bitfeld nach einem gesetzten Bit. Das Bitfeld besteht aus aufeinanderfolgenden Wortoperanden. Wird ein gesetztes Bit gefunden, so wird dies an den Bausteinausgängen angezeigt.



Parameter

R	BINÄR	A, E, M, S, K	Reset des Bausteins und neuer Suchlauf
R/V	BINÄR	A, E, M, S, K	Suchrichtung, rückwärts/vorwärts
#ANZ	DIREKTE KONSTANTE	#, #H	Feldlänge, Anzahl der Wortoperanden
ANF	WORT	EW, AW, MW, KW	Feldanfang, 1. Wortoperand
BIT	WORT	EW, AW, MW, KW	Anzahl der zu überspringenden gesetzten Bits
END	BINÄR	A, M	Feldende erreicht
NR	WORT	AW, MW	Nummer des Wortoperanden, der das gesetzte Bit enthält
POS	WORT	AW, MW	Bitposition innerhalb des Wortoperanden

VE-Daten

Laufzeit:		
Grundlaufzeit:		136 µs
Zusätzliche Laufzeit:		26 µs pro Wortvariable
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		9 Worte
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein durchsucht ein Bitfeld nach einem gesetzten Bit. Das Bitfeld besteht aus aufeinanderfolgenden Wortoperanden. Wird ein gesetztes Bit gefunden, so wird dies an den Bausteinausgängen angezeigt.

Die Ein- und Ausgänge sind weder doppelbar noch negierbar/invertierbar.

Das Bitfeld besteht aus aufeinanderfolgenden Wortoperanden. Am Eingang ANF wird der erste Wortoperand des Bitfeldes angegeben.

Nach dem Start des SPS-Programms (1. Programmzyklus) setzt der Baustein alle Ausgänge auf den Wert 0 und beginnt sofort mit dem Suchvorgang nach gesetzten Bits. Der Suchvorgang ist also bereits im 1. Programmzyklus aktiv.

Findet der Baustein ein gesetztes Bit, so wird dessen Position an den Bausteinausgängen angezeigt und der Suchvorgang beendet. Beim erneuten Aufruf im nächsten Programmzyklus setzt der Baustein den Suchlauf fort und zwar an der Bitposition, die direkt auf das zuletzt gefundene Bit folgt. Ist beim Suchvorgang das Ende des Bitfeldes erreicht, so wird dies angezeigt, und durch ein Reset-Signal kann ein neuer Suchvorgang wieder gestartet werden. Der neue Suchvorgang beginnt wieder am Anfang des Bitfeldes.

R BINÄR

Der Eingang R dient zum Rücksetzen (Reset) des Bausteins und Starten eines neuen Suchvorgangs vom Anfang des Bitfeldes an.

R = 0 → kein Reset

R = 1 → Reset wird ausgelöst

Ein 1-Signal am Eingang R bewirkt ein Rücksetzen aller Bausteinausgänge auf den Wert 0 *und* den sofortigen Start eines neuen Suchlaufs vom Anfang des Bitfeldes an. Der neue Suchvorgang beginnt also noch im selben Bearbeitungszyklus, in dem das Rücksetzen der Bausteinausgänge erfolgt.

Der Suchlauf innerhalb des Reset-Zyklus endet entweder beim ersten gesetzten Bit, oder, wenn kein Bit gesetzt ist, am Feldende. Der Reset-Eingang muß beim nächsten Aufruf auf 0 gesetzt sein, damit der Baustein das Bitfeld weiter nach gesetzten Bits untersuchen kann. Der Reset-Eingang ist dominant gegenüber den anderen Eingängen.

R/V BINÄR

Am Eingang R/V wird die Suchrichtung vorgegeben.

R/V = 0 → suchen vorwärts

R/V = 1 → suchen rückwärts

R/V = 0 bedeutet: Der *Anfang* des Bitfeldes im Sinne des Suchvorgangs ist identisch mit dem *physikalischen Anfang* des Bitfeldes.

R/V = 1 bedeutet: Der *Anfang* des Bitfeldes im Sinne des Suchvorgangs ist identisch mit dem *physikalischen Ende* des Bitfeldes.

#ANZ DIREKTE KONSTANTE

Am Eingang #ANZ wird die Anzahl der Wortoperanden, aus denen das Bitfeld besteht, angegeben. Die Angabe erfolgt als direkte Konstante.

ANF WORT

Am Eingang ANF wird der Wortoperand angegeben, mit dem das Bitfeld physikalisch beginnt. Das gesamte Bitfeld besteht aus dem Operand am Eingang ANF und den entsprechend der Operandennumerierung nachfolgen-

den Operanden. Die Gesamtzahl der Wortoperanden wird am Eingang #ANZ vorgegeben.

BIT WORT

Am Eingang BIT wird angegeben, auf welche Art und Weise der Funktionsbaustein die im Bitfeld gesetzten Bits anzeigen soll.

BIT=1 → *jedes* gesetzte Bit im Bitfeld wird an den Ausgängen NR und POS angezeigt.

BIT=2 → das *erste* gesetzte Bit im Bitfeld wird angezeigt; von den weiteren gesetzten Bits im Bitfeld wird dann nur noch jedes *zweite* gesetzte Bit angezeigt.

BIT=3 → das *erste* gesetzte Bit im Bitfeld wird angezeigt; von den weiteren gesetzten Bits im Bitfeld wird dann nur noch jedes *dritte* gesetzte Bit angezeigt.

BIT=n → das *erste* gesetzte Bit im Bitfeld wird angezeigt; von den weiteren gesetzten Bits im Bitfeld wird dann nur noch jedes *n-te* gesetzte Bit angezeigt.

– Sonderfunktion: BIT = 0

Findet der Baustein ein gesetztes Bit, so wird im nächsten Programmzyklus der Suchvorgang *nicht automatisch* an der darauf folgenden Bitposition fortgesetzt. Der Suchvorgang wird an dieser Stelle solange unterbrochen, bis das gefundene Bit die Wertigkeit 0 angenommen hat. Während der Unterbrechung des Suchvorgangs wird an den Ausgängen weiterhin die Position des zuletzt gefundenen Bits angezeigt. Nimmt das zuletzt gefundene Bit den Wert 0 an, so wird der Suchlauf an der nächsten Bitposition fortgesetzt. Ist in der gewählten Suchrichtung noch ein weiteres Bit gesetzt, so wird die Position dieses Bits angezeigt und der Suchvorgang wiederum unterbrochen.

Sind keine Bits mehr in Suchrichtung gesetzt, so wird die Position des zuletzt gefundenen Bits angezeigt. Der Ausgang END (Feldende erreicht) wird 1 gesetzt.

– Sind nach einem Reset keine Bits gesetzt, so durchläuft der Baustein das Bitfeld bis zum Ende und bleibt dort stehen. Der Ausgang END (Feldende erreicht) wird 1 gesetzt. Die Ausgänge NR und POS werden 0 gesetzt.

– Sind nach einem Reset Bits gesetzt, so zeigt der Baustein das 1. gesetzte Bit in Suchrichtung an und bleibt auf diesem Bit stehen.

END BINÄR

Am Ausgang END wird angezeigt, ob beim Suchvorgang das Ende des Bitfeldes erreicht wurde.

END = 0 → Ende des Bitfeldes nicht erreicht

END = 1 → Ende des Bitfeldes erreicht

Suchen vorwärts: Das Ende des Bitfeldes ist definiert durch die Bitposition 15 im *letzten* Wortoperanden des physikalischen Bitfeldes.

Suchen rückwärts: Das Ende des Bitfeldes ist definiert durch die Bitposition 15 im *ersten* Wortoperanden des physikalischen Bitfeldes.

Ist das Bit an der letzten Bitposition des Feldes gesetzt und wird es bei einem Suchvorgang gefunden und angezeigt, so wird der Ausgang END noch nicht auf den Wert 1 gesetzt. Dies geschieht erst beim nächsten Bearbeitungszyklus des Bausteins. Voraussetzung für das Setzen des Ausganges END ist also, daß beim aktuellen Suchvorgang *kein* gesetztes Bit gefunden wurde.

NR, POS WORT

Findet der Baustein im projektierten Bitfeld ein gesetztes Bit, so wird dessen Position an den Ausgängen NR und POS angezeigt.

Dabei bedeutet:

NR: laufende Nummer des Wortoperanden, in dem das gesetzte Bit gefunden wurde.

POS: Position des gesetzten Bits innerhalb des Wortoperanden.

Laufende Nummer des Wortoperanden

NR = 0 → 1. Wortoperand des Bitfeldes

NR = 1 → 2. Wortoperand des Bitfeldes

⋮

NR = n-1 → n. Wortoperand des Bitfeldes

Die Numerierung der Wortoperanden erfolgt im Sinne des Suchvorgangs.

Beim Vorwärtssuchen gilt also:

erster Wortoperand $\hat{=}$ erstem Wortoperand des physikalischen Bitfeldes

letzter Wortoperand $\hat{=}$ letztem Wortoperand des physikalischen Bitfeldes

Beim Rückwärtssuchen gilt also:

erster Wortoperand $\hat{=}$ letztem Wortoperand des physikalischen Bitfeldes

letzter Wortoperand $\hat{=}$ erstem Wortoperand des physikalischen Bitfeldes

Position des gesetzten Bits innerhalb des Wortoperanden

Die Numerierung innerhalb eines Wortoperanden läuft von 0 ... 15. Dabei entspricht die Position 0 dem niederwertigsten Bit und Position 15 dem höchstwertigsten Bit innerhalb des Wortoperanden.

Wird bei einem Suchvorgang das Feldende erreicht, ohne daß dabei ein gesetztes Bit gefunden wurde, so wird am Ausgang NR und am Ausgang POS weiterhin die Position des zuletzt gefundenen Bits ausgegeben. Dies geschieht solange, bis durch ein Reset am Eingang R ein neuer Suchlauf vom Anfang des Bitfeldes an erzwungen wird.

Sind im gesamten Bitfeld überhaupt keine Bits gesetzt, so nehmen die Ausgänge am Ende des ersten Suchlaufs folgende Werte an:

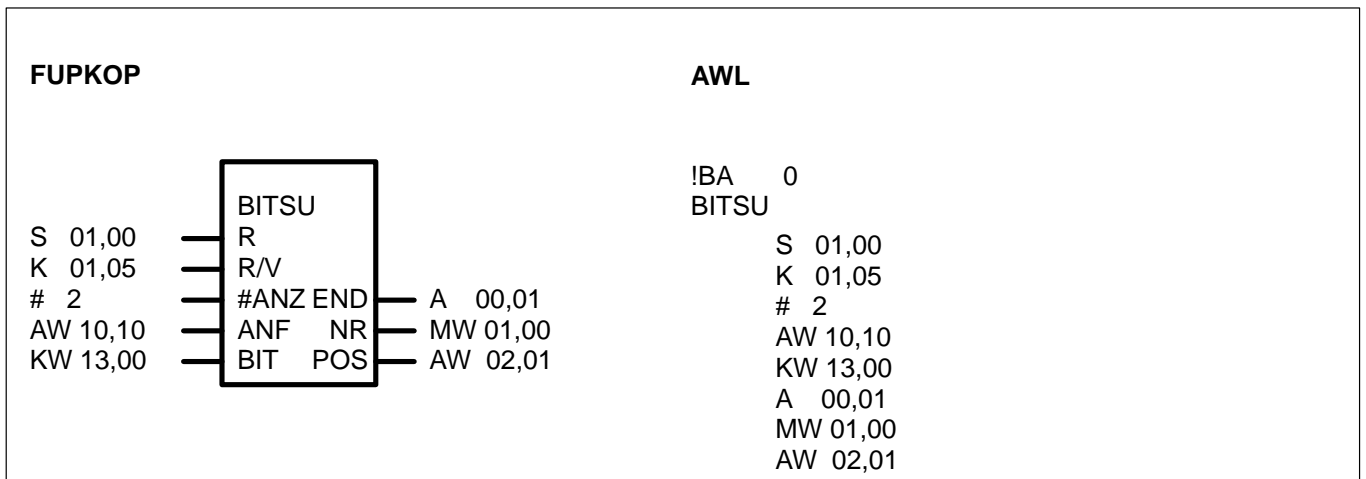
END = 1

NR = 0

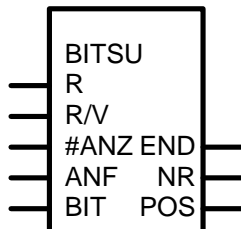
POS = 0

Dieser Zustand kann wieder durch ein 1-Signal am Eingang R (Reset) verlassen werden.

Beispiel



VE-FUP-Definition

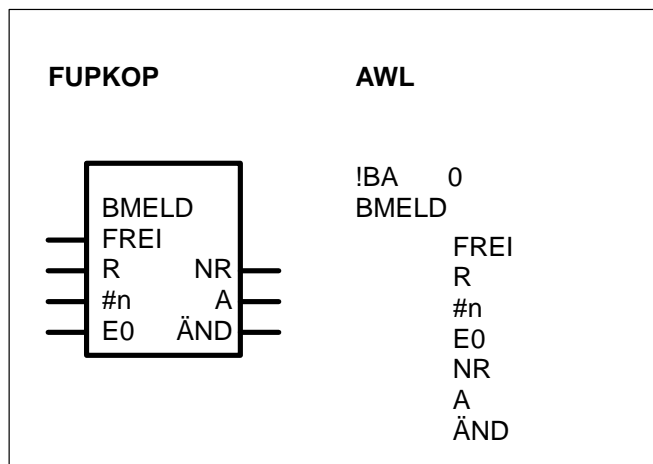


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
R	E	L	N	P	J	0	0
R/V	E	L	N	P	J	0	0
#ANZ	K	W	N	P	J	0	0
ANF	E	W	N	P	J	0	0
BIT	E	W	N	P	J	0	0
END	A	L	N	P	J	0	0
NR	A	W	N	P	J	0	0
POS	A	W	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	BITSU			
00002		PP 0	R	Eingang BINÄR (Reset)
00003		PP 0	R/V	Eingang BINÄR (Suchrichtung)
00004		PP 0	#ANZ	DIREKTE KONSTANTE (Feldlänge #Worte)
00005		PP 0	ANF	Eingang WORT (Feldanfang)
00006		PP 0	BIT	Eingang WORT (Anzahl Bitüberspringer)
00007		PP 0	END	Ausgang BINÄR (Feldende erreicht)
00008		PP 0	NR	Ausgang WORT (Nummer)
00009		PP 0	POS	Ausgang WORT (Bitposition)

Der Funktionsbaustein überwacht die am doppelbaren Eingang E0 anliegenden Binärwerte auf Änderung.



Parameter

FREI	BINÄR	A, E, M, S, K	Baustein-Freigabe
R	BINÄR	A, E, M, S, K	Reset
#n	DIREKTE KONSTANTE	#, #H	Anzahl Eingangswerte
E0	BINÄR	A, E, M, S, K	Eingangswerte; Eingang ist doppelbar
NR	WORT	AW, MW	Nummer des Eingangswertes
A	BINÄR	A, M	aktueller Eingangswert
ÄND	BINÄR	A, M	Änderung erkannt

VE-Daten

Laufzeit (Serie 90):	
Grundlaufzeit:	61 µs
Zusätzliche Laufzeit:	31 µs pro Eingang E0 ... E _{n-1}
Aktualisierung der Ausgänge:	ja, falls FREI=1
Anzahl der Vergangenheitswerte:	gerade Anzahl von Binärwerten: (2 + #n/2) Worte ungerade Anzahl von Binärwerten: (2 + (#n + 1)/2) Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331 Advant Controller 31, Serie 40...50

Beschreibung

Der Funktionsbaustein überwacht die am doppelbaren Eingang E0 anliegenden Binärwerte auf Änderung.

Die Eingänge und Ausgänge sind nicht negierbar/invertierbar. Der Eingang E0 ist doppelbar.

Erkennung einer Änderung

Bei jeder Bearbeitung des Bausteins werden der Reihe nach die aktuellen Eingangswerte an den Eingängen E0 ... E_{n-1} mit den Vergangenheitswerten (Eingangswerte der vorherigen Bausteinbearbeitung) verglichen. Wird an einem der Eingänge E0 ... E_{n-1} eine Änderung erkannt, so wird:

- dies am Ausgang ÄND signalisiert
- die Nummer des Eingangs, an dem die Änderung festgestellt wurde, am Ausgang NR ausgegeben
- der sich geänderte Eingangswert am Ausgang A ausgegeben

Pro Bearbeitung des Bausteins wird nur die Änderung an *einem* Eingang erkannt. Wird eine Änderung erkannt, so werden bei der nächsten Bearbeitung des Bausteins die Eingänge überwacht, die auf den Eingang folgen, an dem zuvor die Änderung festgestellt wurde.

Initialisierung der Vergangenheitswerte

Bei der ersten Bearbeitung nach der SPS-Initialisierung (Frei = 1) bzw. Freigabe der Bearbeitung nachdem sie gesperrt war (FREI wechselt von 0 nach 1), werden alle aktuellen Eingangswerte einmalig als Vergangenheitswerte übernommen, und alle Ausgänge werden auf den Wert 0 gesetzt. Diese initialisierten Vergangenheitswerte stellen nun die Ausgangsbasis zur Erkennung von Änderungen dar.

FREI BINÄR

Mit dem Eingang FREI wird die Bearbeitung des Baustein freigegeben.

FREI = 0 → Baustein wird nicht bearbeitet

FREI = 1 → Bearbeitung des Bausteins ist freigegeben

Ist FREI = 0, dann werden auch die Ausgänge des Bausteins nicht mehr aktualisiert.

R BINÄR

Mit dem Eingang R kann der Baustein zurückgesetzt werden (Reset)

R = 0 → kein Reset

R = 1 → Reset des Bausteins

Reset bedeutet:

- Übernahme der aktuellen Werte an den Eingängen $E_0 \dots E_{n-1}$ als Vergangenheitswerte.
- Alle Ausgänge werden auf den Wert 0 gesetzt.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der zu überwachenden Werte an den Eingängen $E_0 \dots E_{n-1}$ angegeben. Die Angabe erfolgt als direkte Konstante.

Bereich für #n: $1 \leq \#n \leq 127$

E0...E_{n-1} BINÄR

Der Eingang E0 ist doppelbar ($E_0 \dots E_{n-1}$).

An den Eingängen $E_0 \dots E_{n-1}$ werden die auf Änderung zu überwachenden Operanden angegeben.

NR WORT

Am Ausgang NR wird die laufende Nummer des Eingangs $E_0 \dots E_{n-1}$ ausgegeben, an dem eine Änderung festgestellt wurde.

Wird bei der Bearbeitung des Bausteins *keine* Änderung festgestellt, so wird weiterhin am Ausgang NR die Nummer des sich zuletzt geänderten Eingangs ausgegeben.

Es gilt die Zuordnung:

Änderung festgestellt an $E_0 \rightarrow NR = 0$

Änderung festgestellt an $E_1 \rightarrow NR = 1$

Änderung festgestellt an $E_{n-1} \rightarrow NR = n-1$

A BINÄR

Wird an einem der Eingänge $E_0 \dots E_{n-1}$ eine Änderung erkannt, so wird der sich geänderte Eingangswert dem Ausgang A zugewiesen.

Wird bei der Bearbeitung des Bausteins *keine* Änderung an den Eingängen $E_0 \dots E_{n-1}$ festgestellt, so wird am Ausgang A weiterhin der Wert des sich zuletzt geänderten Eingangs ausgegeben.

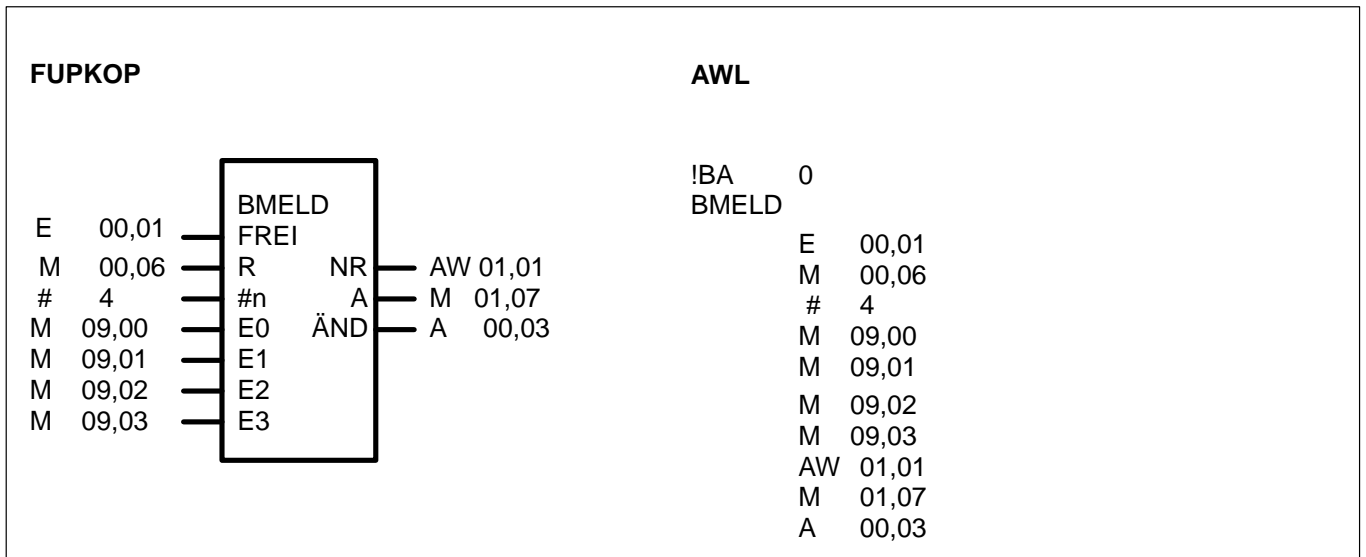
ÄND BINÄR

Am Ausgang ÄND wird angezeigt, ob eine Änderung an den Eingängen $E_0 \dots E_{n-1}$ erkannt wurde.

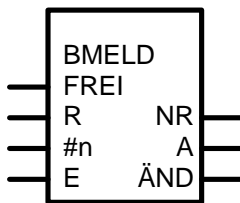
ÄND = 0 → keine Änderung festgestellt

ÄND = 1 → Änderung festgestellt

Beispiel



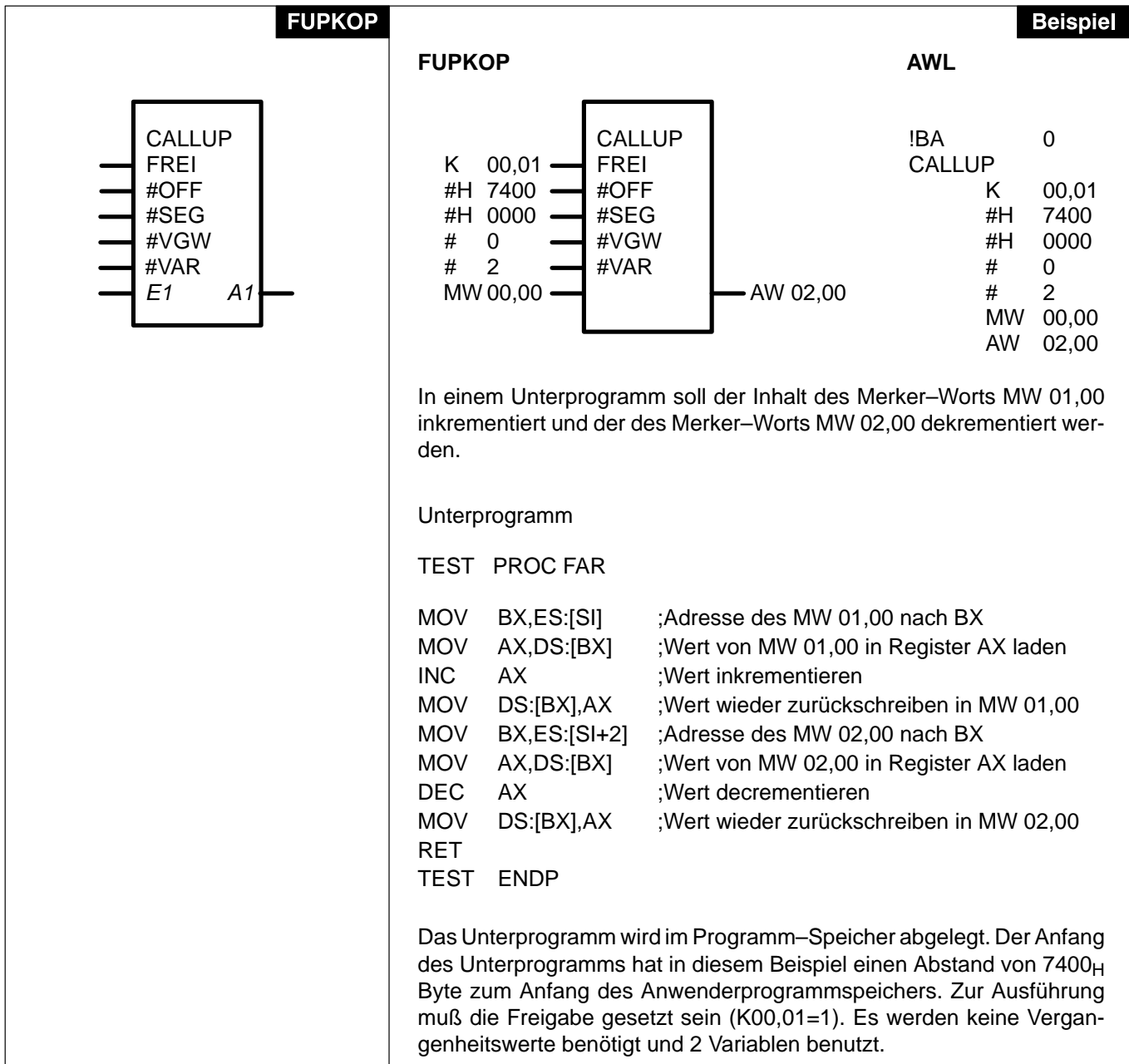
VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
R	E	L	N	P	J	0	0
#n	K	W	N	P	J	0	0
E	E	L	N	P	J	1	0
NR	A	W	N	P	J	0	0
A	A	L	N	P	J	0	0
ÄND	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	BMELD			
00002		PP 0	FREI	Eingang BINÄR (Baustein-Freigabe)
00003		PP 0	R	Eingang BINÄR (Reset)
00004		PP 0	#n	# KONSTANTE (Anzahl Eingangswerte)
	[1		
00005		PP 1	E	Eingang BINÄR (Eingangswerte)
]	1		
00006		PP 0	NR	Ausgang WORT (Nummer)
00007		PP 0	A	Ausgang BINÄR (aktueller Eingangswert)
00008		PP 0	ÄND	Ausgang BINÄR (Änderung erkannt)



Parameter

FREI	BINÄR	E, M, A, K, S	Bausteinfreigabe
#OFF	DIREKTE KONSTANTE	#, #H	Offset–Adresse des Unterprogramms
#SEG	DIREKTE KONSTANTE	#, #H	Segment–Adresse des Unterprogramms
#VGW	DIREKTE KONSTANTE	#, #H	Anzahl der Vergangenheitswerte des Unterprogramms
#VAR	DIREKTE KONSTANTE	#, #H	Anzahl der Ein/Ausgangsvariablen des Unterprogramms
E1	X	E, M, A, K, S, EW, MW, AW, KW, MD, KD	Eingangsvariable (doppelbar)
A1	X	M, A, MW, AW, MD	Ausgangsvariable (doppelbar)

Beschreibung

Der Funktionsbaustein ruft ein in INTEL–Maschinencode auf der SPS abgelegtes Unterprogramm auf. Der Maschinencode muß mit dem Prozessor der jeweiligen SPS kompatibel sein.

Nach Abarbeitung des Unterprogrammes wird wieder in den Baustein zurückgesprungen. Vor dem Aufruf des Unterprogramms werden alle Register automatisch gesichert und nach dem Rücksprung wieder hergestellt.

Der Eingang E1 und der Ausgang A1 sind doppelbar. Die Ein– und Ausgänge sind weder negierbar noch invertierbar.

FREI BINÄR

Freigabe für die Bearbeitung des Funktionsbausteins

FREI = 0: Baustein wird nicht bearbeitet

FREI = 1: Baustein wird bearbeitet

#OFF DIREKTE KONSTANTE

#SEG DIREKTE KONSTANTE

An diesen beiden Eingängen wird die Startadresse des Assembler–Unterprogramms vorgegeben. Die Startadresse besteht aus einer Offset– und einer Segmentadresse. Die Angabe der Offset– und der Segmentadresse erfolgt als direkte Konstante.

Wird als Segmentadresse der Wert "0" angegeben, so erwartet der Baustein, daß das Unterprogramm im Anwenderprogrammspeicher abgelegt ist. Die Offset–adresse stellt dann den Byte–Abstand zwischen Anwenderprogramm anfang und Unterprogramm anfang dar.

Bei Sicherung des SPS–Programms im EPROM bzw. Flash–EPROM wird das Unterprogramm automatisch mitgesichert.

#VGW DIREKTE KONSTANTE

Am Eingang #VGW wird die Anzahl der im Unterprogramm verwendeten Vergangenheitswerte angegeben. Die Angabe erfolgt als direkte Konstante.

#VAR DIREKTE KONSTANTE

Am Eingang #VAR wird die Gesamtzahl der Eingangs– und Ausgangsvariablen des Unterprogramms angegeben. Es handelt sich dabei also um die Summe aller E1 ... En und A1 ... An. Die Angabe erfolgt als direkte Konstante.

E1 BINÄR, WORT, DOPPELWORT

Der Eingang E1 ist doppelbar (E1 ... En).

An den Eingängen E1 ... En werden die Eingangsvariablen für das Unterprogramm angegeben. Das Unterprogramm macht auf diese Variablen einen "lesenden" Zugriff. Beim Zugriff des Unterprogramms auf diese Variablen muß der Assemblerprogrammierer das definierte Datenformat beachten. Direkte Konstanten (Wertübergabe) sind an den Eingängen E1 ... En nicht zulässig.

A1 BINÄR, WORT, DOPPELWORT

Der Ausgang A1 ist doppelbar (A1 ... An).

An den Ausgängen A1 ... An werden die Ausgangsvariablen für das Unterprogramm angegeben. Das Unterprogramm macht auf diese Variablen einen "schreibenden" Zugriff, d. h. im Unterprogramm errechnete Werte werden dem Hauptprogramm mit Hilfe der Ausgangsvariablen A1 ... An zur Verfügung gestellt. Der Assemblerprogrammierer muß beim Zugriff auf diese Variablen das definierte Datenformat beachten.

Anwendungshinweise

Für 07 KR 31 siehe Dokumentation zu dieser Zentraleinheit.

Unterprogramm ablegen

Die beste Möglichkeit für die Ablage des Unterprogrammes ist der Anwenderprogrammspeicher (nach PE). Das Laden des Unterprogramms kann z. B. mit einem HEX–File–Lader oder dem SPS–Monitor erfolgen.

Wird als Segmentadresse der Wert "0" angegeben, so erwartet der Baustein, daß das Unterprogramm im Anwenderprogrammspeicher abgelegt ist. Die Offset–adresse stellt dann den Byte–Abstand zwischen Anwenderprogramm–anfang und Unterprogramm–anfang dar. Bei Sicherung des SPS–Programms im EPROM bzw. Flash–EPROM wird das Unterprogramm automatisch mitgesichert.

Unterprogramm laden

Das Unterprogramm kann mit dem Monitor–Befehl "INTEL–HEX–FILE lesen" über die serielle Schnittstelle COM2 in die SPS eingelesen werden (siehe dazu Beschreibung der Monitorfunktionen, Befehl "R"). Dies ist nur bei Steuerungen möglich, die zwei serielle Schnittstellen haben (z. B. 07 KT 92).

Unterprogramm erstellen

1. Register

Bei der Erstellung des Unterprogramms können folgende Register beliebig verwendet werden:

AX, BX, CX, DX, BP

Für den Zugriff auf Variablen sind folgende Register vorgesehen:

DS, ES, DI, SI

Diese Register können aber auch verändert werden, da sie nach der Ausführung des Unterprogramms im Baustein CALLUP wieder korrigiert werden.

2. Startadresse

Die Anfangsadresse des Unterprogramms muß im Bausteinaufruf an den Eingängen #OFF und #SEG angegeben werden. Die Angabe von Offset und Segment erfolgt jeweils als direkte Konstante.

3. Zugriff auf Ein– und Ausgangsvariablen

Der Zugriff auf die Variablen erfolgt über die Pointer ES:SI und DS:BX.

Dabei befindet sich beim Eintritt in das Unterprogramm in (ES:SI) der Pointer auf die Adresse der ersten, in (ES:SI+2) der Pointer auf die Adresse der zweiten Variablen im Bausteinaufruf, usw. Der Zugriff auf den Inhalt dieser Variablen erfolgt dann mit

DS:BX

Beispiel 1:

Die Werte der ersten beiden *Wort–Variablen* im Bausteinaufruf sollen in die Register AX und CX geholt werden.

```
MOV  BX,ES:[SI]    ;Adresse für 1.Variable
                        ;holen und in BX ablegen
MOV  AX,DS:[BX]    ;Wert nach AX holen
MOV  BX,ES:[SI+2]  ;Adresse für 2.Variable
                        ;holen und in BX ablegen
MOV  CX,DS:[BX]    ;Wert nach CX holen
```

Beispiel 2:

Die Werte der ersten beiden *Binär–Variablen* im Bausteinaufruf sollen in die Register AL und CL geholt werden.

```
MOV  BX,ES:[SI]    ;Adresse für 1.Variable
                        ;holen und in BX ablegen
MOV  AL,DS:[BX]    ;Wert nach AL holen
MOV  BX,ES:[SI+2]  ;Adresse für 2.Variable
                        ;holen und in BX ablegen
MOV  CL,DS:[BX]    ;Wert nach CL holen
```

Binärvariablen belegen ein ganzes Byte. Dieses Byte darf als Inhalt nur 0 oder 1 haben.

4. Zugriff auf Vergangenheitswerte/RAM

Wird im Unterprogramm RAM–Speicherplatz für Zwischenwerte benötigt, so kann dafür der Vergangenheitswert–Speicher verwendet werden.

Die Zwischenwerte bleiben erhalten und stehen im nächsten Programmzyklus zur weiteren Bearbeitung

wieder zur Verfügung. Jedem Aufruf des Unterprogramms stehen seine eigenen Speicherplätze im Vergangenheitswert–Speicher zur Verfügung. Wird ein Unterprogramm z. B. dreimal aufgerufen, so werden also im Vergangenheitswert–Speicher insgesamt dreimal so viele Vergangenheitswerte belegt, wie im Aufruf angegeben.

Die Anzahl der benutzten Speicher–Worte (1 Wort = 16 Bit) im Vergangenheitswert–Speicher ist im Bausteinaufruf als direkte Konstante anzugeben. Werden zu viele Vergangenheitswerte belegt, so gibt die SPS beim Kommando PA (Programmaufbereitung) bzw. beim Programmstart eine Fehlermeldung aus.

Die Adressierung der Vergangenheitswerte erfolgt über die Pointer

DS:DI

Dabei zeigt DS:DI beim Eintritt in das Unterprogramm auf den ersten zur Verfügung stehenden Vergangenheitswert.

Beispiel:

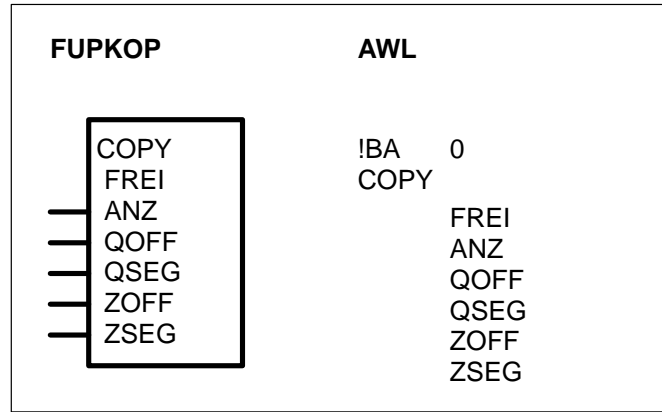
Der unter Punkt 3. nach Register AX geholte Wert soll mit dem ersten Wort im Vergangenheitswert–Speicher verglichen werden.

```
CMP  AX,DS:[DI]    ;Wert mit VW vergleichen
```

5. Die Stacktiefe beträgt 40 Worte. Damit können max. 40 PUSH–Befehle nacheinander durchgeführt werden.
6. Rücksprung aus dem Unterprogramm
Der Rücksprung erfolgt durch ein **RET FAR** (8086–Maschinencode: CB). Ein **RET FAR** wird durch den 8086–Assembler dadurch erzeugt, daß das Unterprogramm als **PROC FAR** vereinbart wird.

VE–Daten	Serie 90	Serie 30, 40, 50
Laufzeit:		
Grundlaufzeit:	29 µs	12 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	hängt vom Unterprogramm des Anwenders ab	
Anzahl der Vergangenheitswerte:	Wert an #VGW Worte	Wert an #VGW Worte
Verfügbar ab:	07 KR 91 R0101, Index c, SW-Vers. 1.1 07 KR 91 R0151, Index c, SW-Vers. 1.1 07 KT 92 R0101, SW-Version 1.0	907 PC 331 / 07 KR 31 V 2.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Der Funktionsbaustein kopiert n Worte aus einem Quell-Speicherbereich in einen Ziel-Speicherbereich.



Parameter

FREI	BINÄR	E,A,M,K,S	Freigabe für den Baustein
ANZ	WORT	EW,AW,MW,KW	Anzahl (n) der zu kopierenden Worte
QOFF	WORT	EW,AW,MW,KW	Offset-Adresse des Anfangs des Quellbereichs
QSEG	WORT	EW,AW,MW,KW	Segment-Adresse des Anfangs des Quellbereichs
ZOFF	WORT	EW,AW,MW,KW	Offset-Adresse des Anfangs des Zielbereichs
ZSEG	WORT	EW,AW,MW,KW	Segment-Adresse des Anfangs des Zielbereichs

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	59 µs	146 µs
Zusätzliche Laufzeit:	2 µs pro kopiertes Wort	30 µs pro kopiertes Wort
Aktualisierung der Ausgänge:	entfällt	entfällt
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 2.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein kopiert n Worte aus einem Quell-Speicherbereich in einen Ziel-Speicherbereich.

Der Inhalt des Quell-Speicherbereichs wird dabei nicht verändert.

Der Anfang des Quell- und des Ziel-Speicherbereichs wird an den Bausteineingängen jeweils durch die Offset- und Segment-Adresse angegeben. Die Offset- und Segment-Adressen werden als Variablen oder als indirekte Konstanten projiziert.

Die Ein- und Ausgänge sind weder doppelbar noch invertierbar/negierbar.

FREI BINÄR
Freigabe für den Baustein
FREI = 0 -> der Baustein wird nicht bearbeitet
FREI = 1 -> der Baustein wird bearbeitet

ANZ WORT
Anzahl n der zu kopierenden Worte.
Es gilt: $0 \leq n \leq + 8000_H$
n = 0: Kein Kopiervorgang
n = 8000_H: Ein ganzes Segment (64 kByte) wird kopiert

QOFF WORT
Offset-Adresse des Anfangs des Quellbereichs

QSEG WORT
Segment-Adresse des Anfangs des Quellbereichs

ZOFF WORT
Offset-Adresse des Anfangs des Zielbereichs

ZSEG WORT
Segment-Adresse des Anfangs des Zielbereichs

Beispiel:
Es sind 128 Worte ab Adresse 4000_H:6 nach Adresse 5000_H:4 zu kopieren.
Anzahl der zu kopierenden Worte: 128
Offset-Adr. des Quellspeichers: 6
Segment-Adr. des Quellspeichers: 4000_H
Offset-Adr. des Zielspeichers: 4
Segment-Adr. des Zielspeichers: 5000_H

Hinweis:

Steht der Funktionsbaustein COPY im SPS-Programm mit der *niedrigeren* Priorität, so kann der Programmier- vorgang vom Programm mit der höheren Priorität unter-

brochen werden. Eine Unterbrechung kann verhindert werden durch: Projektierung des Bausteins COPY im SPS-Programm mit der höheren Priorität.

Beispiel

FUPKOP

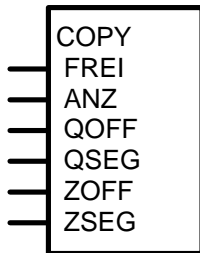
M	00,00	—	COPY
MW	09,07	—	FREI
EW	09,08	—	ANZ
MW	04,02	—	QOFF
KW	00,00	—	QSEG
EW	09,09	—	ZOFF
		—	ZSEG

AWL

```

!BA 0
COPY
M 00,00
MW 09,07
EW 09,08
MW 04,02
KW 00,00
EW 09,09
                    
```

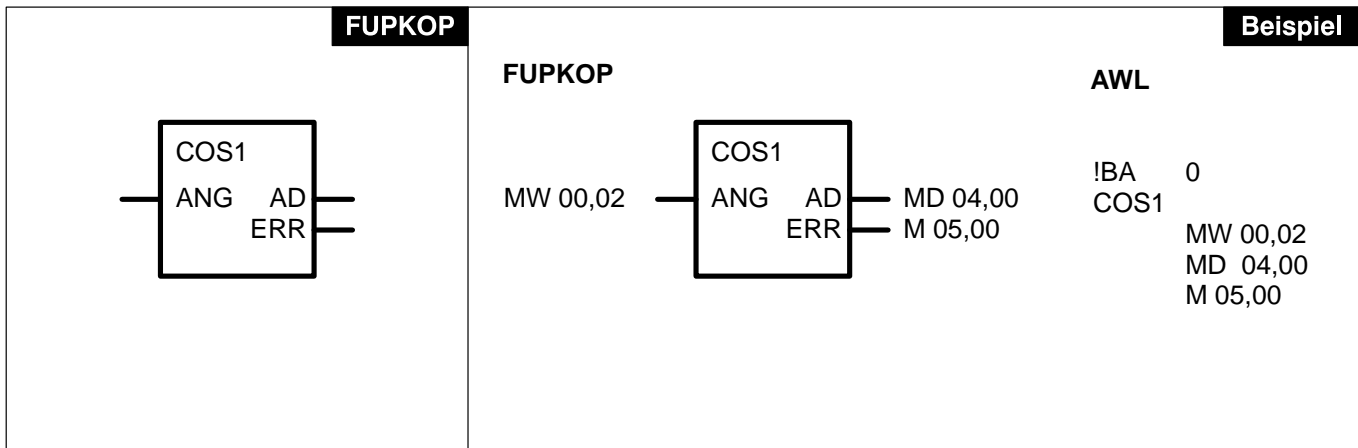
VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
ANZ	E	W	N	P	J	0	0
QOFF	E	W	N	P	J	0	0
QSEG	E	W	N	P	J	0	0
ZOFF	E	W	N	P	J	0	0
ZSEG	E	W	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)	
00001	COPY				
00002		PP 0	FREI	Freigabe	BINÄR
00003		PP 0	ANZ	Anzahl Worte	WORT
00004		PP 0	QOFF	OFFSET Quelle	WORT
00005		PP 0	QSEG	SEGMENT Quelle	WORT
00006		PP 0	ZOFF	OFFSET Ziel	WORT
00007		PP 0	ZSEG	SEGMENT Ziel	WORT



Parameter

ANG WORT EW, AW, MW
 AD DOPPELWORT AD, MD
 ERR BINÄR M, A

Winkel 0 ... 3600 (entspricht 0,0°...360,0°)
 Kosinus des Eingangswertes
 Fehler, falls Eingangswert negativ oder größer 3600

Beschreibung

Der Funktionsbaustein berechnet den Kosinuswert vom Eingang ANG und weist diesen dem Ausgang AD zu. Das Ergebnis liegt im Bereich -100 000 ... 100 000. Ist der Wert am Eingang ANG negativ oder größer 3600 (360°), wird dem Ausgang AD der Wert 0 und dem Ausgang ERR der Wert 1 zugewiesen. Der max. Fehler des Ergebnisses ist ±0,5.

ANG WORT

Es wird der Kosinus des Wertes des Eingangsoperanden ANG berechnet. Das Ergebnis ist als Wert des Ausgangsoperanden AD verfügbar.

Eingabe:

- 0000 für 0 Grad
- 0001 für 0,1 Grad
- 0010 für 1,0 Grad
- ..
- 3600 für 360,0 Grad

AD DOPPELWORT

Der Kosinus des Eingangswertes ist am Ausgang AD verfügbar.

ERR BINÄR

Der Ausgang ERR zeigt an, ob der Eingangswert im korrekten Bereich liegt ($0 \leq ANG \leq 3600$).

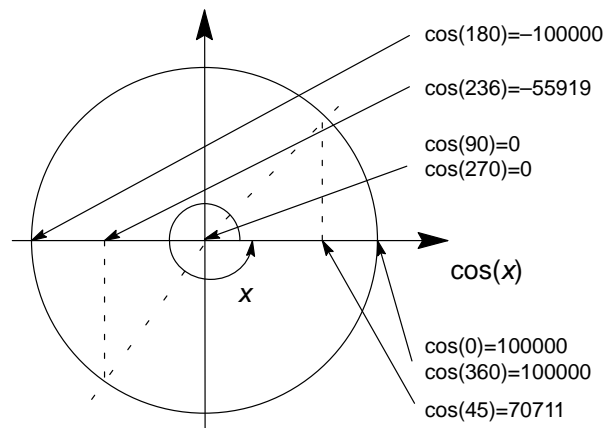
Eingang $0 \leq ANG \leq 3600$

→ ERR = 0 und AD = COS(ANG)

Eingang $ANG < 0$ oder $ANG > 3600$

→ ERR = 1 und AD = 0

Beispiele von Kosinuswerten



VE-Daten

Laufzeit:

Grundlaufzeit:

Zusätzliche Laufzeit:

Aktualisierung der Ausgänge:

Anzahl der Vergangenheitswerte:

Verfügbar ab:

07 KR 91 / 07 KT 92 / 07 KT 93 / 07 KP 62

42 – 71 µs

—

ja

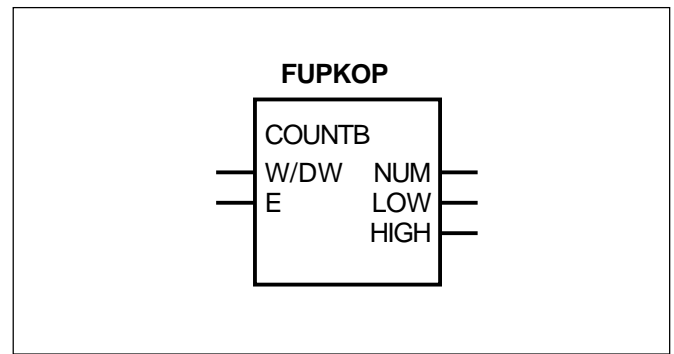
0 Worte

ABB Procontic CS31 07KT92 R202/R262, 07KT93 R171

ABB Procontic T200 07KP62 R202

Die gesetzten Bits in E (Binäreingang W/DW = 0 → E = Wort, W/DW = 1 → E = Doppelwort) werden gezählt und am Ausgang NUM ausgegeben. Zusätzlich wird die Position des kleinsten gesetzten Bits am Ausgang LOW und des größten gesetzten Bits am Ausgang HIGH ausgegeben.

Für E = 0 gilt: NUM = 0, LOW = 0, HIGH = 0



Parameter

W/DW	BINÄR	E, K, M, S	Schalter Wort/Doppelwort W/DW = 0 → E = Wort, W/DW = 1 → E = Doppelwort
E	WORT DOPPELWORT	EW, MW MD	Eingang
NUM	WORT	AW, MW	Anzahl der gesetzten Bits
LOW	WORT	AW, MW	Position des kleinsten gesetzten Bits
HIGH	WORT	AW, MW	Position des größten gesetzten Bits

VE-Daten

Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine

Verfügbar ab: 07 KR 91 R2 Index c, 07 KT 92 R2 Index c, 07 KT 93 Index f, 07 KT 94

Beschreibung

Die gesetzten Bits in E (Binäreingang W/DW = 0 → E = Wort, W/DW = 1 → E = Doppelwort) werden gezählt und am Ausgang NUM ausgegeben. Zusätzlich wird die Position des kleinsten gesetzten Bits am Ausgang LOW und des größten gesetzten Bits am Ausgang HIGH ausgegeben.

Für E = 0 gilt: NUM = 0, LOW = 0, HIGH = 0.

W/DW BINÄR

Bei W/DW = 0 ist der Eingang E ein Wortoperand, bei W/DW = 1 ist der Eingang E ein Doppelwortoperand.

E WORT/DOPPELWORT

Die gesetzten Bits in E werden gezählt und am Ausgang NUM ausgegeben.

Zusätzlich wird die Position des kleinsten gesetzten Bits am Ausgang LOW und des größten gesetzten Bits am Ausgang HIGH ausgegeben.

Für E = 0 gilt: NUM = 0, LOW = 0, HIGH = 0.

NUM WORT

Am Ausgang NUM wird die Anzahl der gesetzten Bits von E ausgegeben.

Ist E ein Wortoperand (W/DW = 0) gilt $0 \leq \text{NUM} \leq 15$.

Ist E ein Doppelwortoperand (W/DW = 1) gilt $0 \leq \text{NUM} \leq 31$.

LOW WORT

Am Ausgang LOW wird die Position des kleinsten gesetzten Bits von E ausgegeben.

Ist E ein Wortoperand (W/DW = 0) gilt: $0 \leq \text{LOW} \leq 15$.
 Ist E ein Doppelwortoperand (W/DW = 1) gilt:
 $0 \leq \text{LOW} \leq 31$.

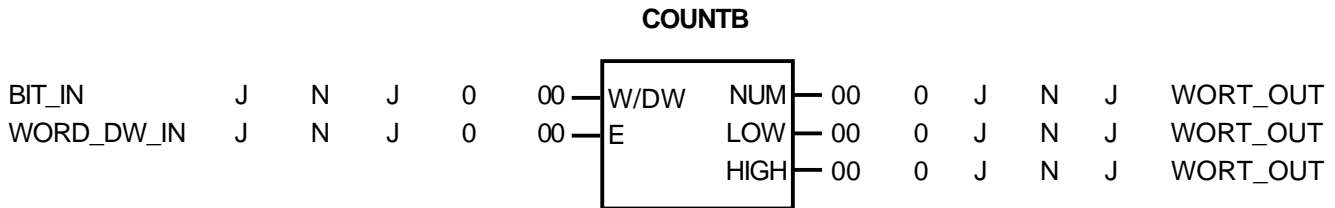
HIGH WORT

Am Ausgang HIGH wird die Position des größten gesetzten Bits von E ausgegeben.

Ist E ein Wortoperand (W/DW = 0) gilt: $0 \leq \text{HIGH} \leq 15$.
 Ist E ein Doppelwortoperand (W/DW = 1) gilt:
 $0 \leq \text{HIGH} \leq 31$.

Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr	Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	----	-----	-----	-----	-----	--------------

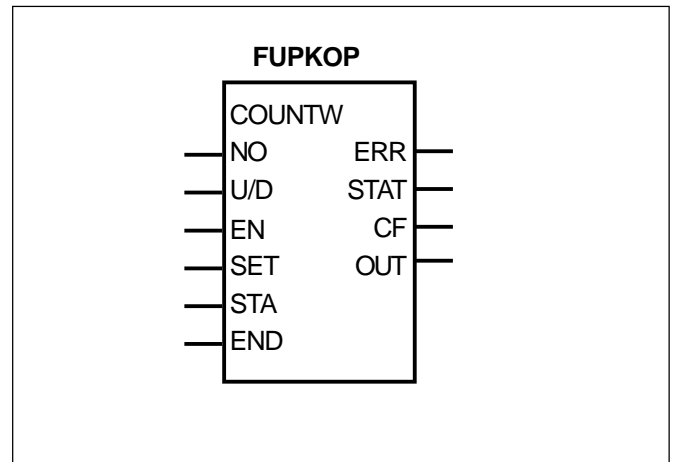


Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	COUNTB		
	00001	PP 00	W/DW	
	00002	PP 00	E	
	00003	PP 00	NUM	
	00004	PP 00	LOW	
	00005	PP 00	HIGH	
	00006			

Der schnelle Zähler in der Zentraleinheit 07 KT 94 arbeitet unabhängig von Anwenderprogrammen und kann damit schnell auf externe Signale reagieren. Er kann in sieben verschiedenen konfigurierbaren Betriebsarten verwendet werden. Die Betriebsart wird in der Systemkonstanten KW 85,02 konfiguriert.

Die projektierte Betriebsart wird nur in der Initialisierungsphase (Power-on, Kaltstart, Warmstart) aktiviert. Für alle Betriebsarten wird der gleiche Funktionsbaustein COUNTW eingesetzt.



Parameter

NO	WORT	EW, MW	Zählernummer 0 oder 1
U/D	BINÄR	E, K, M, S	Richtung 0 = Vorwärts, 1 = Rückwärts
EN	BINÄR	E, K, M, S	Freigabe des Zählvorgangs mit EN = 1
SET	BINÄR	E, K, M, S	Setzen auf Anfangswert mit SET = 1
STA	WORT	EW, MW	Startwert
END	WORT	EW, MW	Endwert
ERR	BINÄR	A, M	Fehler
STAT	WORT	AW, MW	Status
CF	BINÄR	A, M	Carry Flag (1 = Endwert erreicht)
OUT	WORT	AW, MW	Zählerstand (Istwert)

VE - Daten

Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	1 Wort (2 Bytes)
Verfügbar ab:	07 KT 94 Index e

Beschreibung

Der schnelle Zähler in der Zentraleinheit 07 KT 94 arbeitet unabhängig von Anwenderprogrammen und kann damit schnell auf externe Signale reagieren. Er kann in sieben verschiedenen konfigurierbaren Betriebsarten verwendet werden. Die Betriebsart wird in der Systemkonstanten KW 85,02 konfiguriert.

Die projektierte Betriebsart wird nur in der Initialisierungsphase (Power-On, Kaltstart, Warmstart) aktiviert. Für alle Betriebsarten wird der gleiche Funktionsbaustein COUNTW eingesetzt.

Unabhängig von der Betriebsart gelten die folgenden Eigenschaften:

- Gezählt werden die Impulse des Zähler-Eingangssignals bzw. die ausgewerteten Signale Spur A und Spur B bei Anschluß von inkrementalen Drehgebern.
- Die maximale Zählfrequenz beträgt 50 kHz.
- Der Zähler belegt die Klemmen 2 (E 62,00) und 3 (E 62,01) als schnelle Eingänge sowie in einer Betriebsart auch die Ausgangsklemme 46 (A 62,00). Es ist möglich, die Zählfunktion der 07 KT 94 auszuschalten, um alle Ein- und Ausgänge für andere Zwecke zur Verfügung zu haben.
- In allen Betriebsarten kann der Zähler vorwärts zählen. Er zählt dann vom Startwert (Setzwert) bis zum Endwert (max. von -32768 bis +32767 bzw. von 8000_H bis 7FFF_H). Beim Überschreiten von +32767 springt der Zähler nach -32768. Erreicht der Zähler den im Funktionsbaustein projektierten Endwert (Endvalue END), wird im Funktionsbaustein CF = 1 (Endwert erreicht) gespeichert. Beim Setzen des Zählers wird CF auf 0 zurückgesetzt.

NO WORT

In bestimmten Betriebsarten lassen sich gleichzeitig 2 Zähler unabhängig voneinander betreiben. In diesem Fall wird der Funktionsbaustein zweimal projektiert. An den Eingang NO muß dann bei dem ersten Funktionsbaustein eine 0 (für Zähler 0) und an dem zweiten Funktionsbaustein eine 1 (für Zähler 1) angelegt werden. Normalerweise wird NO mit 0 projektiert.

U/D BINÄR

In einigen Betriebsarten kann der Zähler auch rückwärts zählen. Ist dies gewünscht, so ist der Eingang U/D (Up/Down) des Funktionsbausteins auf 1 zu projektieren. Der Zähler zählt dann vom Startwert (Setzwert) bis zum Endwert (max. von +32767 bis -32768 bzw. von $7FFF_H$ bis 8000_H). Beim Unterschreiten von -32768 springt der Zähler nach +32767.

EN BINÄR

Das Verarbeiten der Zählsignale muß freigegeben werden. Dies geschieht je nach Betriebsart über eine Klemme oder über den Eingang EN (Enable) des Funktionsbausteins mit EN = 1.

STA WORT

Der Zähler kann auf einen Anfangswert gesetzt werden. Dieser Anfangswert muß an den Eingang STA (Start value) des Funktionsbausteins angelegt werden. Mit einem Setzsignal (je nach Betriebsart über eine Klemme oder über Eingang SET des Funktionsbausteins) wird der Wert der Wortvariablen in den Zähler übernommen.

Hinweis: Wenn das Setzsignal bei vorhandener Freigabe über mehrere Verarbeitungszyklen ansteht, dann setzt der Verarbeitungsprozessor den Zähler immer wieder am Programmende. In der übrigen Zeit des Verarbeitungszyklus zählt der Zähler Impulse.

END WORT

Erreicht der Zähler den projektierten Endwert END, wird der Binärausgang CF = 1 speichernd gesetzt.

ERR BINÄR

Der Binärausgang ERR (Error) zeigt an, daß im Ausgabewort STAT (Fehler-Status) ein Fehlercode vorliegt.

STAT WORT

Das Ausgabewort in STAT (Fehler-Status) hat folgende Bedeutung:

- 1...255 = ausgewählte Betriebsart in KW 85,02
- 256 = keine Betriebsart ausgewählt (KW 85,02=0)
- 257 = keine zugelassene Betriebsart in KW 85,02 projektiert

CF BINÄR

Erreicht der Zähler den projektierten Endwert END, wird CF = 1 speichernd gesetzt. Beim Setzen des Zählers wird CF auf 0 zurückgesetzt.

OUT WORT

Der aktuelle Zählerstand (Istwert) kann jederzeit über den Ausgang OUT des Funktionsbausteins abgefragt werden.

Im folgenden wird beschrieben, welche Betriebsarten konfiguriert werden können, wie sie projektiert werden und wie sie sich voneinander unterscheiden. Die Systemkonstante KW 85,02 konfiguriert die Zählerbetriebsart. Hexwert im Low-Byte (Bit 0 bis Bit 7) von KW 85,02 bedeuten:

- 00_H = kein Zähler (Standardeinstellung)
- 01_H = Mode 1, ein Vorwärtszähler
- 02_H = Mode 2, ein Vorwärtszähler mit Freigabeeingang
- 03_H = Mode 3, zwei Vorwärts-/Rückwärtszähler
- 04_H = Mode 4, zwei Vor-/Rückzähler, der zweite Zähler zählt die Impulse auf der fallenden Flanke.
- 05_H = Mode 5, ein V/R-Zähler mit dynamischem Setzeingang, Setzen auf der steigendem Flanke
- 06_H = Mode 6, ein V/R-Zähler mit dynamischen Setzeingang, Setzen auf der fallenden Flanke
- 07_H = Mode 7, ein V/R-Zähler mit Richtungsdiskriminator

Das High-Byte wird mit 00_H konfiguriert.

Achtung: Die projektierte Betriebsart wird nur in der Initialisierungsphase (Power-on, Kaltstart, Warmstart) aktiviert.

– **Betriebsart 0:** Kein Zähler

Diese Betriebsart ist zu wählen, wenn der interne schnelle Zähler nicht gebraucht wird. Alle digitalen Ein- und Ausgänge stehen dann für andere Zwecke zur Verfügung.

– **Betriebsart 1:** 1 Vorwärtszähler

Die folgenden zwei Klemmen sind für den Zähler reserviert und können anderweitig nicht benutzt werden:

- Klemme 2 (E 62,00): Zähleingang
- Klemme 46 (A 62,00): Ausgang „Endwert erreicht“

Die Freigabe des Zähleingangs und des Ausgangs „Endwert erreicht“ wird über den Funktionsbaustein mit EN = 1 erreicht.

- **Betriebsart 2:** 1 Vorwärtszähler mit Freigabeeingang über Klemme

Die folgenden drei Klemmen sind für den Zähler reserviert und können anderweitig nicht benutzt werden:

Klemme 2 (E 62,00): Zählereingang
 Klemme 3 (E 62,01): Freigabeeingang
 Klemme 46 (A 62,00): Ausgang „Endwert erreicht“

Der Freigabeeingang gibt sowohl den Zählereingang als auch den Ausgang „Endwert erreicht“ frei. Der Zähler wird nur freigegeben, wenn der Freigabeeingang = 1 und beim Funktionsbaustein EN = 1 ist.

- **Betriebsart 3:** 2 Vorwärts-/Rückwärtszähler

In dieser Betriebsart existieren zwei voneinander unabhängige Zähler. Die Meldung „Endwert erreicht“ ist nur über die Funktionsbausteine abfragbar, nicht über Klemmen.

Die folgenden zwei Klemmen sind für den Zähler reserviert und können anderweitig nicht benutzt werden:

Klemme 2 (E 62,00): Zählereingang Zähler 0
 Klemme 3 (E 62,01): Zählereingang Zähler 1

Der Funktionsbaustein COUNTW wird zweimal benötigt.

Hinweis:

Ein Wechsel der Zählrichtung (über U/D) ist im laufenden Betrieb nicht vorgesehen. Wird U/D umgeschaltet, bewirkt dies gleichzeitig ein Setzen des Zählers auf den Anfangswert, danach wird in der neuen Zählrichtung weitergezählt.

- **Betriebsart 4:** 2 Zähler (1 Zählereingang invertiert)

Diese Betriebsart entspricht vollständig der Betriebsart 3, mit der einzigen Ausnahme, daß der zweite Zählereingang (Zähler 1) invertiert ist. Er zählt die 1/0-Flanken an Klemme 3 (E 62,01).

- **Betriebsart 5:** 1 Vor-/Rückzähler mit dynamischem Setzeingang über Klemme

In dieser Betriebsart steht ein Vorwärts-/Rückwärtszähler mit einem dynamischen Setzeingang zur Verfügung. Dynamisch heißt hier, daß das Setzen nur auf der Signalfanke wirkt und nicht, solange das Signal ansteht.

Die folgenden zwei Klemmen sind für den Zähler reserviert und können anderweitig nicht benutzt werden:

Klemme 2 (E 62,00): Zählereingang
 Klemme 3 (E 62,01): Dynamischer Setzeingang

Der dynamische Setzeingang an der Klemme 3 wirkt auf der 0/1 -Flanke.

Die Meldung „Endwert erreicht“ ist nur über den Funktionsbaustein abfragbar, nicht über eine Klemme.

Hinweis:

Ein Wechsel der Zählrichtung (über U/D) ist im laufenden Betrieb nicht vorgesehen. Wird U/D umgeschaltet, bewirkt dies gleichzeitig ein Setzen des Zählers auf den Anfangswert, danach wird in der neuen Zählrichtung weitergezählt.

- **Betriebsart 6:** 1 V/R-Zähler mit dynamischem Setzeingang über Klemme

Diese Betriebsart entspricht vollständig der Betriebsart 5, mit der einzigen Ausnahme, daß der dynamische Setzeingang auf der 1/0-Flanke wirkt.

- **Betriebsart 7:** Zähler mit Richtungsdiskriminator

In dieser Betriebsart lassen sich Drehgeber mit zwei um 90 Grad versetzten Zählimpulsen anschließen. Der Zähler zählt je nach Lage der Impulse vorwärts oder rückwärts. Eine Impulsvervielfachung (x2 oder x4) findet nicht statt.

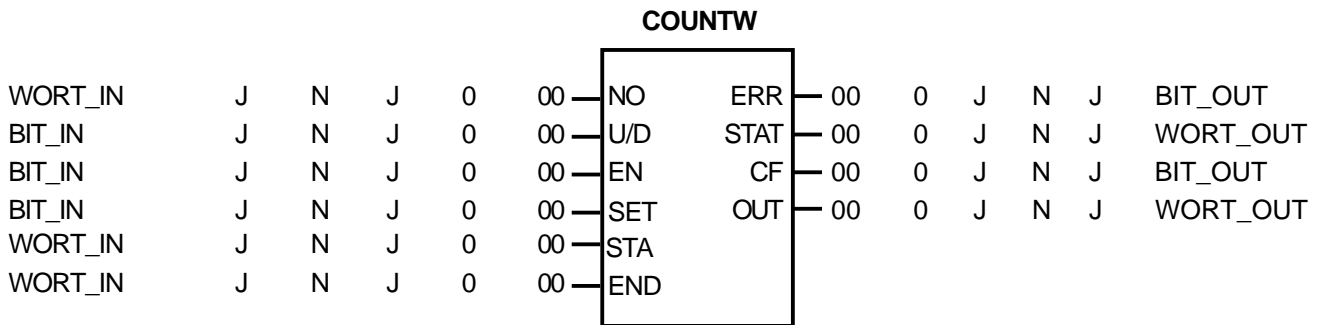
Es eignen sich Drehgeber mit Signalen von 24 V. Signale von 5-V-Drehgebern müssen hochgesetzt werden. Eine Drehgeber-Nullspur wird nicht verarbeitet. Die Meldung „Endwert erreicht“ ist nur über den Funktionsbaustein abfragbar, nicht über eine Klemme.

Die folgenden zwei Klemmen sind für den Zähler reserviert und können anderweitig nicht benutzt werden:

Klemme 2 (E62,00): Spur A des Drehgebers
 Klemme 3 (E62,01): Spur B des Drehgebers

Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr	Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	----	-----	-----	-----	-----	--------------



Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	;Baustein Count_W Anfang		
	00001	IBA	000	
	00002	COUNTW		
	00003	PP	00	NO
	00004	PP	00	U/D
	00005	PP	00	EN
	00006	PP	00	SET
	00007	PP	00	STA
	00008	PP	00	END
	00009	PP	00	ERR
	00010	PP	00	STAT
	00011	PP	00	CF
	00012	PP	00	OUT
	00013	;ENDE Baustein COUNT_W		
	00014			

FUPKOP		Beispiel
	<p>FUPKOP</p> <pre> M 00,00 --- FREI RDY MW 10,00 --- GRN OK MW 10,01 --- CODE ERR MW 10,02 --- D1 A1 MW 10,03 --- D2 A2 MW 10,04 --- D3 A3 MW 10,05 --- D4 A4 MW 10,06 --- D5 A5 MW 10,07 --- D6 A6 MW 10,08 --- D7 A7 MW 10,09 --- D8 </pre>	<p>AWL</p> <pre> IBA 0 CS31CO M 00,00 MW 10,00 MW 10,01 MW 10,02 MW 10,03 MW 10,04 MW 10,05 MW 10,06 MW 10,07 MW 10,08 MW 10,09 A 00,00 M 00,01 MW 10,10 MW 10,11 MW 10,12 MW 10,13 MW 10,14 MW 10,15 MW 11,00 MW 11,01 </pre>
	<p>Beispiel für die Konfiguration des analogen Eingangs EW 00,03 auf Strombereich 4 – 20 mA:</p> <pre> MW 10,00 = 0 Gruppennummer 0 MW 10,01 = 170 Auftragskennung für "Analogeingabe-Kanal konfigurieren" MW 10,02 = 3 Kanalnummer 3 MW 10,03 = 49 Kennung für Strombereich 4 – 20 mA </pre> <p>Diese Merker können z. B. mit der Funktion "Überschreiben" im Automatisierungsgerät auf diese Werte gesetzt werden.</p>	

Parameter

FREI	BINÄR	E, A, M, S	Freigabe (0→1-Flanke) zur Bearbeitung des Bausteins
GRN	WORT	EW, AW, MW, KW	Gruppennummer des Vorortmoduls, auf den sich der Auftrag bezieht
CODE	WORT	EW, AW, MW, KW	Kennung des durchzuführenden Auftrages
D1	WORT	EW, AW, MW, KW	1. Parameter des Auftrages
:	:	:	:
D8	WORT	EW, AW, MW, KW	8. Parameter des Auftrages
RDY	BINÄR	A, M	Bearbeitung des Auftrages ist abgeschlossen
OK	BINÄR	A, M	Auftrag konnte korrekt bearbeitet werden
ERR	WORT	AW, MW	Fehlermeldung/Statusmeldung
A1	WORT	AW, MW	1. Parameter der Antwort
:	:	:	:
A7	WORT	AW, MW	7. Parameter der Antwort

Beschreibung

Der Funktionsbaustein dient zum Konfigurieren der CS31–Vorortmodule. Der Baustein kann sowohl Konfigurationsparameter an die Vorortmodule schicken, als auch deren aktuell eingestellte Konfiguration abfragen.

Außer der Konfiguration der CS31–Vorortmodule kann der Funktionsbaustein noch weitere Aufträge bearbeiten (siehe Liste der Aufträge).

Hinweis:

Der Funktionsbaustein CS31CO ersetzt das Verknüpfungselement CS31CONF.

Die Freigabe zur *einmaligen* Bearbeitung eines Auftrages erfolgt durch eine 0→1–Flanke am Eingang FREI.

Am Eingang CODE wird die gewünschte Auftragskennung angegeben.

Die für den Auftrag erforderlichen Parameter werden an den Eingängen D1 ... D8 projiziert.

An den Ausgängen RDY, OK und ERR werden Statusmeldungen signalisiert.

Die Antwortdaten des Auftrages stehen an den Ausgängen A1 ... A7 zur Verfügung.

Die Bearbeitung des Auftrages kann mehrere SPS–Zyklen dauern.

FREI BINÄR

Über den Eingang FREI wird die Bearbeitung des Bausteins gesteuert.

FREI = 0: *alle* Bausteinausgänge werden auf den Wert "0" gesetzt. Dies gilt aber nicht *während* einer gerade laufenden Auftragsabwicklung, d. h. die Bearbeitung eines gerade laufenden Auftrages wird durch FREI = 0 nicht beeinflusst.

FREI = 0-1–Flanke: Die Bearbeitung des Auftrages wird freigegeben. *Während* der Bearbeitung des Auftrages wird der Eingang FREI *nicht* mehr ausgewertet.

FREI = 1: Der Baustein wird *nicht* bearbeitet, d. h. der Baustein verändert seine Ausgänge nicht mehr. Dies gilt aber nicht *während* einer gerade laufenden Auftragsabwicklung.

GRN WORT

Gruppennummer, mit der das Vorortmodul vom Automatisierungsprogramm aus angesprochen wird.

Bereich: 0 ... 63

Beispiel:

Beim Binäreingang E 12,08 ist "12" die Gruppennummer und "08" die Kanalnummer.

CODE WORT

Am Eingang CODE wird die Kennung des auszuführenden Auftrages angegeben (siehe Liste der Aufträge auf der nächsten Seite).

D1...D8 WORT

An den Eingängen D1 ... D8 werden die für den Auftrag erforderlichen Parameter vorgegeben. Die Anzahl der Parameter hängt vom durchzuführenden Auftrag ab. Es gibt auch Aufträge, die keine Parameter benötigen (siehe dazu Liste der Aufträge auf der nächsten Seite).

RDY BINÄR

Der Ausgang RDY signalisiert, daß die Bearbeitung des laufenden Auftrages abgeschlossen ist. Der Ausgang RDY macht keine Aussage darüber, ob die Bearbeitung des Auftrages *mit Erfolg* durchgeführt werden konnte oder nicht. Der Ausgang RDY ist deshalb immer zusammen mit dem Ausgang OK zu betrachten.

RDY = 1 und OK = 1

Die Bearbeitung des Auftrages wurde *fehlerfrei* abgewickelt. Ein neuer Auftrag kann mit einer 0→1–Flanke am Eingang FREI gestartet werden.

RDY = 1 und OK = 0

Bei der Bearbeitung des Auftrages wurde *ein Fehler* festgestellt. Am Ausgang ERR steht die zugehörige Fehlerkennung zur Verfügung. Ein neuer Auftrag kann mit einer 0→1–Flanke am Eingang FREI gestartet werden.

RDY = 0:

Die Bearbeitung eines freigegebenen Auftrages ist noch nicht abgeschlossen (Auftrag läuft noch) *oder* mit FREI = 0 wurde der Ausgang RDY zurückgesetzt

OK BINÄR

Der Ausgang OK signalisiert, ob der Auftrag erfolgreich abgewickelt wurde, oder ob bei der Bearbeitung ein Fehler festgestellt wurde. Im Fehlerfall wird am Ausgang ERR eine Fehlernummer angezeigt.

Der Ausgang OK ist erst gültig, wenn der Auftrag abgeschlossen ist d.h. wenn RDY = 1 ist.

es gilt:

wenn RDY = 1 *und*

OK = 1: Der Auftrag wurde erfolgreich bearbeitet

OK = 0: Bei der Bearbeitung des Auftrages wurde ein Fehler festgestellt.

ERR WORT

Am Ausgang ERR werden Status- und Fehlerkennungen ausgegeben.

Die Statuskennungen werden *während der Bearbeitung* eines Auftrages ausgegeben, um zu signalisieren in welchem Bearbeitungsstadium sich der Auftrag gerade befindet. Nach der Freigabe eines Auftrages werden Statuskennungen also nur solange signalisiert, wie RDY = 0 ist.

Die Fehlerkennungen werden *nach Abschluß* der Auftragsbearbeitung ausgegeben, falls ein Fehler aufgetreten ist. Fehlerkennungen werden also erst dann signalisiert, wenn

RDY = 1 und

OK = 0

sind.

Fehlerkennungen

ERR = 1: Es wurde am Eingang CODE eine unzulässige Auftragskennung angegeben.

ERR = 2: Es wurden an den Eingängen D1 ... D8 falsche Parameter angegeben (z. B. eine Gruppennummer, zu der es kein Vorortmodul am CS31–Bus gibt).

ERR = 3: Das angesprochene CS31–Vorortmodul akzeptiert den Auftrag nicht.

Statuskennungen

ERR = 8: Der Funktionsbaustein wartet, weil momentan gerade ein Auftrag eines anderen Auftraggebers bearbeitet wird.

ERR = 10: Der Auftrag wurde an den Empfänger abgeschickt und der Baustein wartet auf dessen Antwort.

A1...A7 WORT

An den Ausgängen A1 ... A7 steht nach Abschluß der Auftragsbearbeitung die Antwort zur Verfügung.

Die Anzahl der Antwort–Parameter hängt vom ausgeführten Auftrag ab (siehe Liste der Aufträge).

Liste der Aufträge

Die Abwicklung eines Auftrages besteht aus:

- dem Verschicken des Auftrages und
- der Bereitstellung der OK–Antwort bzw. Nicht–OK–Antwort

Die OK–Antwort wird beim jeweiligen Auftrag beschrieben.

Die Nicht–OK–Antwort der einzelnen Aufträge sieht immer wie folgt aus:

- **Nicht–OK–Antwort**

für die Nicht–OK–Antwort gilt prinzipiell:

RDY: 1

OK: 0

ERR: 1 ☞ unzulässige Auftragskennung
 2 ☞ falscher Parameter, z. B. Gruppennummer, zu der es kein Vorortmodul gibt
 3 ☞ Vorortmodul akzeptiert den Auftrag nicht

A1 ... A7: 0

- **Aktualisierung der maximalen Anzahl erkannter Vorortmodule**

Im Eingangswort EW 07,15 steht u. a. die maximale Anzahl der in der Vergangenheit erkannten Vorortmodule. Die momentane wirkliche Anzahl der vorhandenen Vorortmodule kann geringer sein.

Mit diesem Kommando wird dieser Wert aktualisiert. Die vorhandenen Module werden gezählt und der Wert wird abgeleitet.

Der Anwender kann diesen Wert im SPS–Programm (EW 07,15, Bit 8 ... 15) abfragen.

- **Auftrag**

GRN: 255 (Master–SPS mit Bus)

CODE: 132

D1 ... D8: nicht benutzt

- **OK–Antwort**

RDY: 1

OK: 1

A1 ... A7: 0

○ **Abfrage, ob bei einem Eingang die Drahtbruchüberwachung ein- oder ausgeschaltet ist**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 32
 D1: Kanalnummer
 D2 ... D8: nicht benutzt

● **OK-Antwort**

RDY: 1
 OK: 1
 A1: 47 ⇐ Drahtbruchüberwachung EIN
 32 ⇐ Drahtbruchüberwachung AUS
 A2 ... A7: 0

○ **Abfrage, ob bei einem Ausgang die Drahtbruchüberwachung ein- oder ausgeschaltet ist**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 33
 D1: Kanalnummer
 D2 ... D8: nicht benutzt

● **OK-Antwort**

RDY: 1
 OK: 1
 A1: 47 ⇐ Drahtbruchüberwachung EIN
 32 ⇐ Drahtbruchüberwachung AUS
 A2 ... A7: 0

○ **Drahtbruchüberwachung eines Einganges ein- bzw. ausschalten**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 224 ⇐ Drahtbruchüberwachung EIN
 160 ⇐ Drahtbruchüberwachung AUS
 D1: Kanalnummer
 D2 ... D8: nicht benutzt

● **OK-Antwort**

RDY: 1
 OK: 1
 A1 ... A7: 0

○ **Drahtbruchüberwachung eines Ausganges ein- bzw. ausschalten**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 225 ⇐ Drahtbruchüberwachung EIN
 161 ⇐ Drahtbruchüberwachung AUS
 D1: Kanalnummer
 D2 ... D8: nicht benutzt

● **OK-Antwort**

RDY: 1
 OK: 1
 A1 ... A7: 0

○ **Abfrage, ob ein Kanal als Eingang oder als Eingang/Ausgang konfiguriert ist**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 34
 D1: Kanalnummer
 D2 ... D8: nicht benutzt

● **OK-Antwort**

RDY: 1
 OK: 1
 A1: 34 ⇐ Eingang
 35 ⇐ Eingang/Ausgang
 A2 ... A7: 0

○ **Konfiguration eines Kanals als Eingang oder Eingang/Ausgang**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 162 ⇐ Eingang
 163 ⇐ Eingang/Ausgang
 D1: Kanalnummer
 D2 ... D8: nicht benutzt

● **OK-Antwort**

RDY: 1
 OK: 1
 A1 ... A7: 0

○ **Abfragen der Eingangsverzögerung eines Kanals**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 38
 D1: Kanalnummer
 D2 ... D8: nicht benutzt

● **OK-Antwort**

RDY: 1
 OK: 1
 A1: Eingangsverzögerung:
 2 ⇐ 2 ms
 4 ⇐ 4 ms
 .
 .
 .
 30 ⇐ 30 ms
 32 ⇐ 32 ms
 A2 ... A7: 0

○ **Einstellen der Eingangsverzögerung eines Kanals**

● **Auftrag**

GRN: Gruppennummer 0...63
 CODE: 166
 D1: Kanalnummer
 D2: Eingangsverzögerung
 2 ⇨ 2 ms
 4 ⇨ 4 ms
 .
 .
 .
 30 ⇨ 30 ms
 32 ⇨ 32 ms

● **OK-Antwort**

RDY: 1
 OK: 1
 A1 ... A7: 0

○ **Fehler auf Vorortmodul quittieren**

Mit diesem Kommando werden die auf dem gewählten Vorortmodul registrierten Fehlermeldungen zurückgesetzt. Ein Rücksetzen ist nur möglich, wenn die Fehlerursache nicht mehr wirksam ist.

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 232
 D1: kleinste Kanalnummer auf dem Modul:
 0 ⇨ kleinste Kanalnummer auf dem Modul ist 0 (≤ 7)
 8 ⇨ kleinste Kanalnummer auf dem Modul ist 8 (> 7)
 D2: Modultyp:
 0 ⇨ Binäre Eingabe
 1 ⇨ Analoge Eingabe
 2 ⇨ Binäre Ausgabe
 3 ⇨ Analoge Ausgabe
 4 ⇨ Binäre Ein-/Ausgabe
 5 ⇨ Analoge Ein-/Ausgabe

Hinweis:
 Bit: gerade Zahl (0, 2, 4)
 Wort: ungerade Zahl (1, 3, 5)

D3 ... D8: nicht benutzt

● **OK-Antwort**

RDY: 1
 OK: 1
 A1 ... A7: 0

○ **Fehler auf Vorortmodul quittieren und Konfigurationswerte auf Standardeinstellung (Default) rücksetzen**

Zusätzlich zum Auftrag "Fehler auf Vorortmodul quittieren" werden alle konfigurierbaren Einstellungen auf die Standardeinstellung rückgesetzt.

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 233
 D1: erste Kanalnummer auf dem Modul:
 0 ⇨ erste Kanalnummer auf dem Modul ist 0 (≤ 7)
 8 ⇨ erste Kanalnummer auf dem Modul ist 8 (> 7)

D2: Modultyp:
 0 ⇨ Binäre Eingabe
 1 ⇨ Analoge Eingabe
 2 ⇨ Binäre Ausgabe
 3 ⇨ Analoge Ausgabe
 4 ⇨ Binäre Ein-/Ausgabe
 5 ⇨ Analoge Ein-/Ausgabe

Hinweis:
 Bit: gerade Zahl (0, 2, 4)
 Wort: ungerade Zahl (1, 3, 5)

D3 ... D8: nicht benutzt

● **OK-Antwort**

RDY: 1
 OK: 1
 A1 ... A7: 0

○ **Abfragen der Konfiguration eines analogen Einganges**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 42
 D1: Kanalnummer
 D2 ... D8: nicht benutzt

● **OK-Antwort**

RDY: 1
 OK: 1
 A1: 50 ⇨ Eingang 0 ... 20 mA
 49 ⇨ Eingang 4 ... 20 mA
 A2 ... A7: 0

○ **Abfragen der Konfiguration eines analogen Ausgangs**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 43
 D1: Kanalnummer
 D2 ... D8: nicht benutzt

● **OK–Antwort**

RDY: 1
 OK: 1
 A1: 50 ☞ Ausgang 0 ... 20 mA
 49 ☞ Ausgang 4 ... 20 mA
 51 ☞ Ausgang $\pm 10V$
 A2 ... A7: 0

○ **Konfiguration eines analogen Einganges**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 170
 D1: Kanalnummer
 D2: 50 ☞ Eingang 0 ... 20 mA
 49 ☞ Eingang 4 ... 20 mA
 D3 ... D8: nicht benutzt

● **OK–Antwort**

RDY: 1
 OK: 1
 A1 ... A7: 0

○ **Konfiguration eines analogen Ausgangs**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 171
 D1: Kanalnummer
 D2: 50 ☞ Ausgang 0 ... 20 mA
 49 ☞ Ausgang 4 ... 20 mA
 51 ☞ Ausgang $\pm 10V$
 D3 ... D8: nicht benutzt

● **OK–Antwort**

RDY: 1
 OK: 1
 A1 ... A7: 0

○ **Abfragen der Bus–Konfiguration**

Die Busanschaltung der Master–SPS verfügt über eine Liste, in der bestimmte Daten der Vorortmodule abgelegt sind. Die Vorortmodule sind in dieser Liste in der Reihenfolge numeriert, in welcher sie am CS31–Bus aufgefunden wurden. Bei diesem Kommando wird die interne Nummer der Module angegeben. Als Antwort erhält man die unter dieser Nummer gespeicherte Gruppennummer sowie Statusinformationen zu dem entsprechenden Modul.

● **Auftrag**

GRN: wird nicht ausgewertet
 CODE: 80
 D1: Nummer aus der Modulliste (1 ... 31)
 D2 ... D8: nicht benutzt

● **OK–Antwort**

RDY: 1
 OK: 1
 A1: Status des Vorortmoduls:
 Bit 0 ... 3: Anzahl der Prozeßdaten–Bytes (Binärmodul) bzw. Worte (Wortmodul), die das Modul dem Master sendet
 Bit 4 ... 7: Anzahl der Prozeßdaten–Bytes (Binärmodul) bzw. Worte (Wortmodul), die der Master dem Modul sendet
 A2: Gruppennummer (0 ... 63)
 A3: Bit 0: 0 ☞ kleinste Kanalnummer ≤ 7
 1 ☞ kleinste Kanalnummer > 7
 Bit 1: 0 ☞ Binärmodul
 1 ☞ Wortmodul

A4 ... A7: 0

○ **1 ... 6 Bytes lesen**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 49 ☞ 1 Byte lesen
 50 ☞ 2 Bytes lesen
 51 ☞ 3 Bytes lesen
 52 ☞ 4 Bytes lesen
 53 ☞ 5 Bytes lesen
 54 ☞ 6 Bytes lesen

D1: erste Kanalnummer auf dem Modul:
 0 ☞ erste Kanalnummer auf dem Modul ist 0 (≤ 7)
 8 ☞ erste Kanalnummer auf dem Modul ist 8 (> 7)

D2: Modultyp:
 0 ⇨ Binäre Eingabe
 1 ⇨ Analoge Eingabe
 2 ⇨ Binäre Ausgabe
 3 ⇨ Analoge Ausgabe
 4 ⇨ Binäre Ein-/Ausgabe
 5 ⇨ Analoge Ein-/Ausgabe

Hinweis:
 Bit: gerade Zahl (0, 2, 4)
 Wort: ungerade Zahl (1, 3, 5)

D3: Byte–Anfangsadresse (Low Byte)
 D4: Byte–Anfangsadresse (High Byte)
 D5 ... D8: nicht benutzt

● **OK–Antwort**

RDY: 1
 OK: 1
 A1: Wert des 1. Bytes
 A2: Wert des 2. Bytes oder 0
 A3: Wert des 3. Bytes oder 0
 A4: Wert des 4. Bytes oder 0
 A5: Wert des 5. Bytes oder 0
 A6: Wert des 6. Bytes oder 0
 A7: 0

○ **1 Bit von einem Byte lesen**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 63
 D1: erste Kanalnummer auf dem Modul:
 0 ⇨ erste Kanalnummer auf dem Modul ist 0 (≤ 7)
 8 ⇨ erste Kanalnummer auf dem Modul ist 8 (> 7)

D2: Modultyp:
 0 ⇨ Binäre Eingabe
 1 ⇨ Analoge Eingabe
 2 ⇨ Binäre Ausgabe
 3 ⇨ Analoge Ausgabe
 4 ⇨ Binäre Ein-/Ausgabe
 5 ⇨ Analoge Ein-/Ausgabe

Hinweis:
 Bit: gerade Zahl (0, 2, 4)
 Wort: ungerade Zahl (1, 3, 5)

D3: Byte–Anfangsadresse (Low Byte)
 D4: Byte–Anfangsadresse (High Byte)
 D5: Bitposition innerhalb des Bytes 0 ... 7
 D6 ... D8: nicht benutzt

● **OK–Antwort**

RDY: 1
 OK: 1
 A1: Bitwert (0 oder 1)
 A2 ... A7: 0

○ **1 ... 4 Bytes schreiben**

● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 65 ⇨ 1 Byte schreiben
 66 ⇨ 2 Bytes schreiben
 67 ⇨ 3 Bytes schreiben
 68 ⇨ 4 Bytes schreiben

D1: erste Kanalnummer auf dem Modul:
 0 ⇨ erste Kanalnummer auf dem Modul ist 0 (≤ 7)
 8 ⇨ erste Kanalnummer auf dem Modul ist 8 (> 7)

D2: Modultyp:
 0 ⇨ Binäre Eingabe
 1 ⇨ Analoge Eingabe
 2 ⇨ Binäre Ausgabe
 3 ⇨ Analoge Ausgabe
 4 ⇨ Binäre Ein-/Ausgabe
 5 ⇨ Analoge Ein-/Ausgabe

Hinweis:
 Bit: gerade Zahl (0, 2, 4)
 Wort: ungerade Zahl (1, 3, 5)

D3: Byte–Anfangsadresse (Low Byte)
 D4: Byte–Anfangsadresse (High Byte)
 D5: Wert des 1. Bytes
 D6: Wert des 2. Bytes oder nicht benutzt
 D7: Wert des 3. Bytes oder nicht benutzt
 D8: Wert des 4. Bytes oder nicht benutzt

● **OK–Antwort**

RDY: 1
 OK: 1
 A1 ... A7: 0

○ **1 Bit von einem Byte schreiben**● **Auftrag**

GRN: Gruppennummer 0 ... 63
 CODE: 79
 D1: erste Kanalnummer auf dem Modul:
 0 ☞ erste Kanalnummer auf dem Modul ist 0 (≤ 7)
 8 ☞ erste Kanalnummer auf dem Modul ist 8 (> 7)
 D2: Modultyp:
 0 ☞ Binäre Eingabe
 1 ☞ Analoge Eingabe
 2 ☞ Binäre Ausgabe
 3 ☞ Analoge Ausgabe
 4 ☞ Binäre Ein-/Ausgabe
 5 ☞ Analoge Ein-/Ausgabe

Hinweis:

Bit: gerade Zahl (0, 2, 4)

Wort: ungerade Zahl (1, 3, 5)

D3: Byte–Anfangsadresse (Low Byte)
 D4: Byte–Anfangsadresse (High Byte)
 D5: Bitposition innerhalb des Bytes 0 ... 7
 D6: Bitwert (0 oder 1)
 D7 ... D8: nicht benutzt

● **OK–Antwort**

RDY: 1
 OK: 1
 A1 ... A7: 0

VE–Daten**Serie 90****Serie 30, 40, 50**

Laufzeit:

Grundlaufzeit:

100 μ s

nicht verfügbar

Zusätzliche Laufzeit:

—

—

Aktualisierung der Ausgänge:

nein; Empfehlung: Ausgang direkt an Merker zuweisen oder Zwischenmerker global vergeben; siehe Griff 3, Kapitel 2.5.3

Anzahl der Vergangenheitswerte:

1 Wort

1 Wort

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

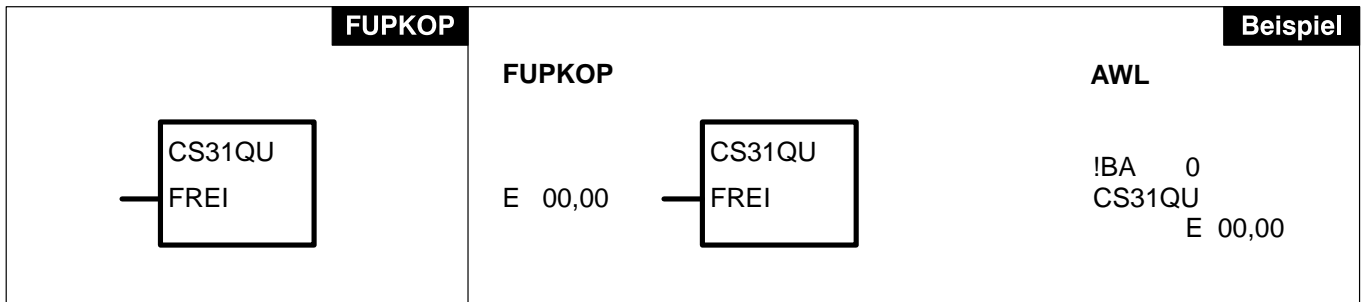
907 PC 331 / 07 KR 31 V 1.0,

07 KR 91 R 101, Index c, SW–Version V 1.1

07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

07 KR 91 R 151, Index c, SW–Version V 1.1

07 KT 92 R 101, SW–Version V 1.0, 07 KT 93



Parameter

FREI BINÄR E, A, M, S Freigabe der Bausteinbearbeitung

Beschreibung

Mit diesem Funktionsbaustein können Fehlermeldungen von CS31-Vorort-Modulen automatisch quittiert werden

Fehlermeldungen werden auf den CS31-Vorort-Modulen solange gespeichert, bis sie quittiert werden. Selbst wenn der Fehler beseitigt ist, steht die Fehlermeldung auf dem Modul bis zur Quittierung an und wird auch solange an das Automatisierungsgerät gemeldet.

Hinweis:
Der Funktionsbaustein CS31QU ersetzt das Verknüpfungselement CS31QUIT.

Mit einem 1-Signal am Eingang FREI wird die Bearbeitung des Bausteins freigegeben, und dieser führt dann fortwährend Quittierungen von CS31-Fehlern durch. Die Quittierung eines Fehlers auf einem CS31-Modul kann mehrere SPS-Zyklen dauern.

Ist der Funktionsbaustein freigegeben, so prüft er ständig, ob ein CS31-Fehler der Klasse 3 oder 4 vorliegt und quittiert diesen.

- 1. Es liegt ein CS31-Fehler der Klasse 3 vor:**
Der Baustein quittiert den Fehler auf dem CS31-Vorort-Modul, das den Fehler meldet *und* löscht auch die Fehlermeldung auf dem Automatisierungsgerät, d. h. der Fehlermerker M 255,13 wird zurückgesetzt, und die Leuchtdiode FK3 wird ausgeschaltet.

Beispiel für einen FK3-Fehler:
– ein Vorortmodul wird vom CS31-Bus abgetrennt

Hinweis:
Damit Fehler der Klasse 3 bearbeitet werden können, muß die Systemkonstante KW 00,07 (FK3_REAK) = 0 sein! Ist KW 00,07 ungleich 0, so bricht die SPS beim Erkennen eines FK3-Fehlers automatisch die Bearbeitung des Automatisierungsprogrammes ab.

- 2. Es liegt ein CS31-Fehler der Klasse 4 vor:**
Der Baustein quittiert den Fehler auf dem CS31-Vorort-Modul, das den Fehler meldet *und* löscht auch die Fehlermeldung auf der SPS, d. h. der Fehlermerker M 255,14 wird zurückgesetzt,

Beispiel für einen FK4-Fehler:
– ein Vorortmodul meldet Drahtbruch

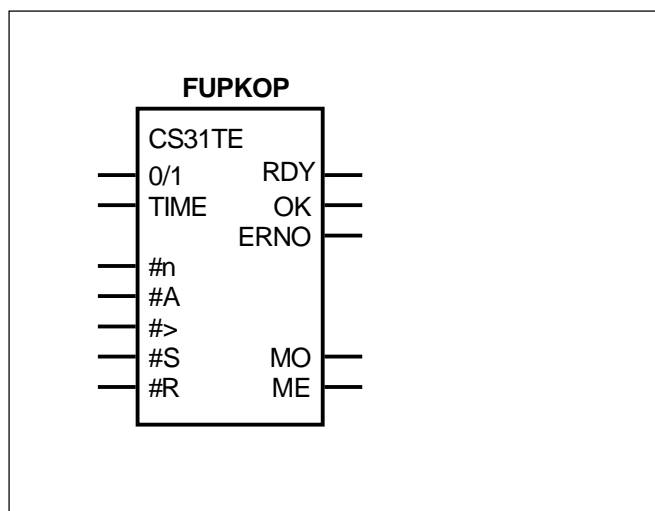
VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	60 µs	nicht verfügbar
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	—	—
Anzahl der Vergangenheitswerte:	1 Wort	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331 07 KR 91 R 101, Index c, SW-Version V 1.1 07 KR 91 R 151, Index c, SW-Version V 1.1 07 KT 92 R 101, SW-Version V 1.0, 07 KT 93	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Diesen Baustein kann nur der Master am CS31-Bus (KW 00,00 = -1) verwenden. Bei der Initialisierung meldet sich jedes Vor-Ort-Modul mit folgenden Daten beim Master an: Adresse, kleinste Kanalnummer und Anzahl der Prozeßdatenbytes bzw. -worte, die das Modul dem Master sendet und/oder von ihm empfängt. Diese Daten vergleicht der Baustein mit der Beschreibung der Vor-Ort-Module an den Bausteineingängen.



Parameter

0/1	BINÄR	E, K, M, S,	Freigabe für einmalige Bearbeitung des Bausteins mit 0/1-Flanke
TIME	DOPPELWORT	KD, MD	Zeit für Änderungskennung der Anzahl der Module in EW 07,15
RDY	BINÄR	A, M	Bearbeitung abgeschlossen
OK	BINÄR	A, M	Modul am CS31-Bus ok.
ERNO	WORT	AW, MW	Fehlernummer 255: kein Master >0: Vor-Ort-Modul kann nicht zugeordnet werden (Listenpointer Mail Abfrage der Buskonfiguration)
#n	DIREKTE KONSTANTE	#, #H	Anzahl der Module
#A	DIREKTE KONSTANTE	#, #H	Adresse (Gruppennummer) des Vor-Ort-Moduls
#>	DIREKTE KONSTANTE	#, #H	0: kleinste Kanalnummer ≤ 7 1: kleinste Kanalnummer > 7
#S	DIREKTE KONSTANTE	#, #H	bei Binärmodulen: Anzahl der Prozeßdatenbytes, die das Modul dem Master sendet (0...15 entspricht 0 bis 15 Bytes) bei Wortmodulen: Anzahl der Prozeßdatenworte, die das Modul vom Master empfängt (100...108 entspricht 0 bis 8 Worte)
#R	DIREKTE KONSTANTE	#, #H	bei Binärmodulen: Anzahl der Prozeßdatenbytes, die das Modul vom Master empfängt (0...15 entspricht 0 bis 15 Bytes) bei Wortmodulen: Anzahl der Prozeßdatenworte, die das Modul vom Master empfängt (100...108 entspricht 0...8 Worte)
MO	BINÄR	A, M	Modul ok. (Modulangaben stimmen mit den Bausteingaben überein)
ME	BINÄR	A, M	Daten senden/empfangen von Modul und Baustein sind unterschiedlich

Die Ein/Ausgänge von #A bis ME sind für jedes Modul doppelbar.

VE-Daten

Aktualisierung der Ausgänge:	nein
Anzahl der Verganheitswerte:	3 Worte (6 Bytes)

Verfügbar ab:	07 KR 91 R2 Index c, 07 KT 92 R2 Index c, 07 KT 93 R2 Index f, 07 KT 94
---------------	--

Beschreibung

Diesen Baustein kann nur der Master am CS31-Bus (KW 00,00 = -1) verwenden. Bei der Initialisierung meldet sich jedes Vor-Ort-Modul mit folgenden Daten beim Master an: Adresse, kleinste Kanalnummer und Anzahl der Prozeßdatenbytes bzw. -worte, die das Modul dem Master sendet und/oder von ihm empfängt. Nach einer 0/1-Flanke am Eingang 0/1 vergleicht der Baustein diese Daten mit der Beschreibung der Vor-Ort-Module an den Bausteineingängen. Bei Übereinstimmung wird für dieses Vor-Ort-Modul der Ausgang MO = 1 und ME = 0 gesetzt. Ändert sich die Anzahl der Vor-Ort-Module (EW 07,15) innerhalb der Zeit am Eingang TIME nicht, wird die Bearbeitung des Bausteins abgeschlossen (RDY = 1). Bei OK = 1 stimmen alle Module am Bus mit der Beschreibung am Baustein überein. Bei OK = 0 kann an dem Ausgang ERNO und an den Ausgängen MO und ME der einzelnen Module der entsprechende Fehler abgelesen werden.

0/1 BINÄR

Mit einer 0/1-Flanke am Baustein 0/1 wird die einmalige Bearbeitung des Bausteins freigegeben. Bei RDY = 1 ist die Bearbeitung abgeschlossen. An den Ausgängen OK und ERNO (bzw. MO und ME der einzelnen Module) kann abgelesen werden, ob bei der Bearbeitung ein Fehler aufgetreten ist.

TIME DOPPELWORT

Am Eingang TIME wird die Zeit für die Änderungserkennung der Anzahl der Module in EW 07,15 angegeben. Ändert sich die Anzahl der Vor-Ort-Module (EW 07,15) nach einer 0/1-Flanke am Eingang 0/1 innerhalb dieser Zeit nicht, wird die Bearbeitung des Bausteins abgeschlossen (RDY = 1 bedeutet: Ausgänge OK, ERNO, MO, ME sind gültig).

Empfohlen: 2000 bis 4000 (entspricht 2 bis 4 Sekunden)

RDY BINÄR

Am Ausgang RDY wird nach einer 0/1-Flanke am Eingang 0/1 angezeigt, daß die Bearbeitung des Bausteins abgeschlossen ist (RDY = 1 bedeutet: Ausgänge OK, ERNO, MO, ME sind gültig).

OK BINÄR

Am Ausgang OK wird angezeigt, daß die Vor-Ort-Module am CS31-Bus mit den Bausteinparametern übereinstimmen. Der Ausgang muß immer im Zusammenhang mit dem Ausgang RDY betrachtet werden. Es gilt:

RDY = 1 und OK = 1: Die Vor-Ort-Module am CS31-Bus stimmen mit den Bausteinparametern überein.

RDY = 1 und OK = 0: Es ist ein Fehler aufgetreten. Der Fehler kann mit den Ausgängen ERNO und MO und ME der einzelnen Module ausgewertet werden.

ERNO WORT

Am Ausgang ERNO wird eine Fehlernummer ausgegeben. Der Ausgang muß immer in Zusammenhang mit den Ausgängen RDY und OK (RDY = 1 und OK = 0) betrachtet werden. Es gilt:

ERNO = 255 -> Gerät ist kein Master

ERNO > 0 -> Vor-Ort-Modul am CS31-Bus kann nicht zugeordnet werden (Der Wert von ERNO ist der Pointer auf die Liste beim Mail „Abfrage der Bus-Konfiguration“, siehe 907 PC-331 Bedien- und Testfunktionen).

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der Module am CS3-Bus angegeben. Die Eingänge #A, #>, #S, #R, MO und ME sind für jedes Modul doppelbar.

#A DIREKTE KONSTANTE

Am Eingang #A wird die Adresse (Gruppennummer) des Vor-Ort-Moduls angegeben.

#> DIREKTE KONSTANTE

Am Eingang #> wird die kleinste Kanalnummer des Vor-Ort-Moduls angegeben.

Es gilt: 0: kleinste Kanalnummer \leq 7
1: kleinste Kanalnummer $>$ 7

#S DIREKTE KONSTANTE

Am Eingang #S wird die Anzahl der Prozeßdatenbytes (bei Binärmodulen) bzw. Prozeßdatenworte (bei Wortmodulen) angegeben, die das Modul dem Master sendet. Es gilt:

bei Binärmodulen: 0...15 (entspricht 0 bis 15 Bytes), Anzahl der Prozeßdatenbytes, die das Modul dem Master sendet.

bei Wortmodulen: 100...108 (entspricht 0 bis 8 Worte), Anzahl der Prozeßdatenworte, die das Modul dem Master sendet.

Gültige Werte für #S und #R:

- Binärmodul: #S und #R: 0...15
- Wortmodul: #S und #R: 100...108

#R DIREKTE KONSTANTE

Am Eingang #R wird die Anzahl der Prozeßdatenbytes (bei Binärmodulen) bzw. Prozeßdatenworte (bei Wortmodulen) angegeben, die das Modul vom Master empfängt. Es gilt:

bei Binärmodulen: 0...15 (entspricht 0 bis 15 Bytes), Anzahl der Prozeßdatenbytes, die das Modul vom Master empfängt.

bei Wortmodulen: 100...108 (entspricht 0 bis 8 Worte), Anzahl der Prozeßdatenworte, die das Modul vom Master empfängt.

Gültige Werte für #S und #R:

- Binärmodul: #S und #R: 0...15
- Wortmodul: #S und #R: 100...108

MO BINÄR

Am Ausgang MO (Modul ok.) wird angegeben, daß die Modulangaben mit den Bausteinangaben für dieses Vor-Ort-Modul übereinstimmen (MO = 1 und ME = 0).

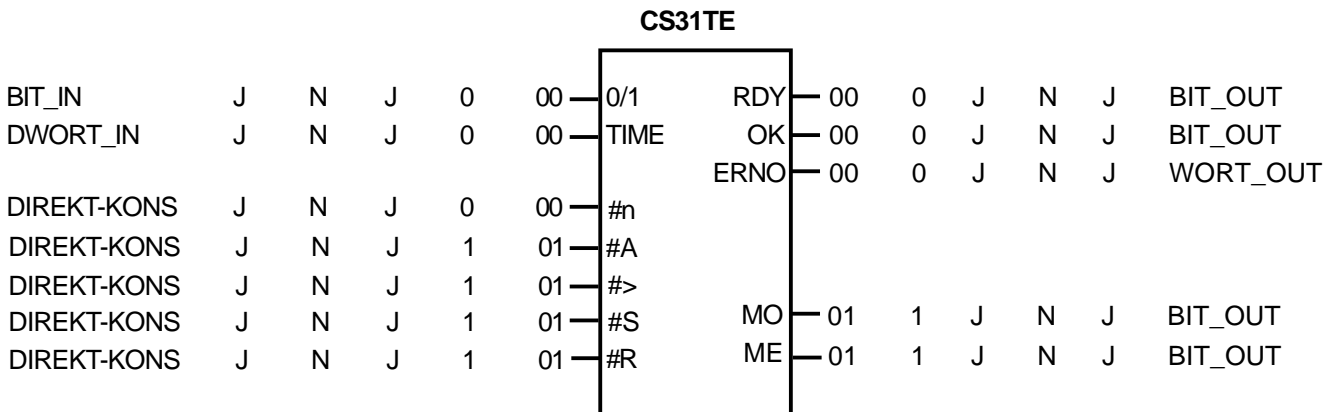
ME BINÄR

Am Ausgang ME (Modul Error) wird angegeben, dass die Modulangaben Daten senden/empfangen und die Bausteinangaben für dieses Vor-Ort-Modul nicht übereinstimmen (MO = 0 und ME = 1).

Die Ein/Ausgänge von #A bis ME sind für jedes Modul doppelbar.

Grafik

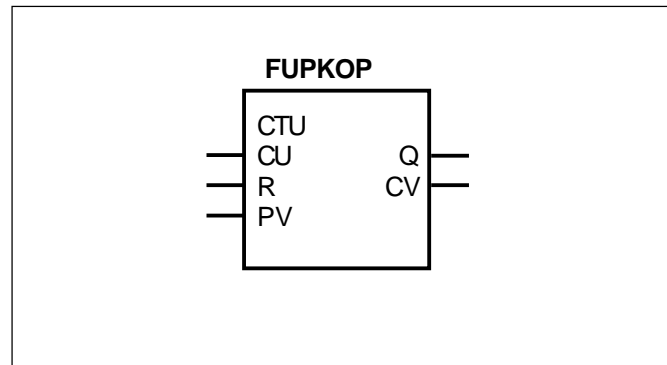
Plaus-gruppe	Pf1	Inv	Anz	Art	Nr		Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	--	----	-----	-----	-----	-----	--------------



Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	!BA	000	
	00001	CS31TE		
	00002	PP 00	0/1	
	00003	PP 00	TIME	
	00004	PP 00	RDY	
	00005	PP 00	OK	
	00006	PP 00	ERNO	
	00007	PP 00	#n	
	00008	[01	
	00009	PP 01	#A	
	00010	PP 01	#>	
	00011	PP 01	#S	
	00012	PP 01	#R	
	00013	PP 01	MO	
	00014	PP 01	ME	
	00015]	01	
	00016			

Der Funktionsbaustein dient zum Zählen von Impulsen. Beim Zählvorgang wird jeweils die positive Flanke des Impulses am Eingang CU ausgewertet. Sie erhöht den Wert am Ausgang CV um "1".



Parameter

CU	BINÄR	E, M, A, K, S	Impulseingang zählen
R	BINÄR	E, M, A, K, S	Rücksetzen des Zählers
PV	WORT	EW, MW, AW, KW	Sollwert des Zählers
Q	BINÄR	M, A	Ausgang Sollwert erreicht
CV	WORT	MW, AW	Ausgang für Zählerstand

Verfügbar ab:

Advant Controller 31, Serie 40 und 50

Beschreibung

Der Funktionsbaustein dient zum Zählen von Impulsen. Beim Zählvorgang wird jeweils die positive Flanke des Impulses am Eingang CU ausgewertet. Sie erhöht den Wert am Ausgang CV um "1".

Erreicht der Zähler den am Eingang PV definierten Sollwert, nimmt der Ausgang Q den Wert 1 an.

Mit R lässt sich der Zähler auf 0 zurücksetzen.

CU BINÄR

Eine positive Flanke erhöht den Zählerstand um 1.

R BINÄR

Ein 1-Signal setzt den Zähler am Ausgang CV auf den Wert 0.

PV WORT

Sollwert des Zählers

Q BINÄR

Ein 1-Signal an diesem Ausgang zeigt an, daß der Zählerstand den Sollwert erreicht hat:

$$CV < PV \rightarrow Q = 0$$

$$CV = PV \rightarrow Q = 1$$

$$CV > PV \rightarrow Q = 1$$

CV WORT

Aktueller Zählerstand, der mit jeder positiven Flanke am Eingang CU um 1 erhöht wird.

Wertebereich:

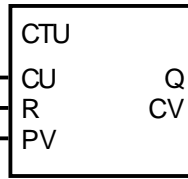
Integer Wort (16 Bit)

- untere Grenze: 0000_H bzw. 0

- obere Grenze: 7FFF_H bzw. +32767

Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr.		Nr.	Art	Anz	Inv	Pf1	Plaus-gruppe	
BIT_IN	J	N	J	0	00	CU	Q	00	0	J	N	J	BIT_OUT
BIT_IN	J	N	J	0	00	R	CV	00	0	J	N	J	WORT_OUT
WORT_IN	J	N	J	0	00	PV							



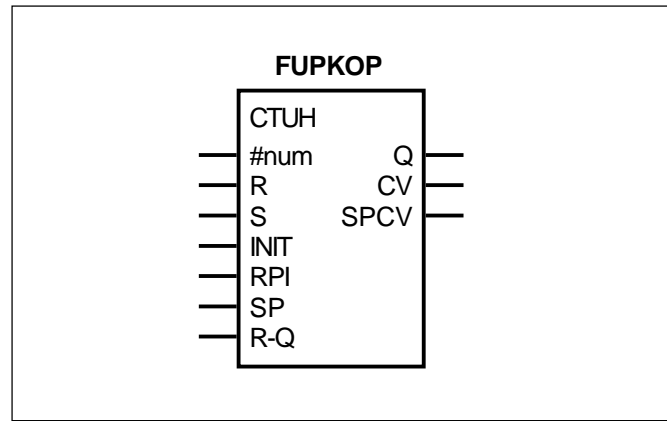
Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	!BA	000	
	00001	CTU		
	00002	PP	00	CU
	00003	PP	00	R
	00004	PP	00	PV
	00005	PP	00	Q
	00006	PP	00	CV

Der Funktionsbaustein dient zum Zählen von Impulsen, die mit höherer Frequenz am Zähl Eingang anstehen. Es werden die drei Betriebsarten der Zähler in den Zentralgeräten der Serie 40 und 50 unterstützt. Damit ermöglicht er die Auswertung von Impulsen, die von einem Encoder erzeugt werden.

Beim Zählvorgang wird jeweils die positive Flanke der Impulse am Eingang E62,00 bzw. E62,01 ausgewertet und der Wert am Ausgang CV um den Wert "1" verändert.

In den Betriebsarten 1 und 2 erhöht sich der Wert, in Betriebsart 3 wird er abhängig von der Drehrichtung (Encoder – Spuren A und B) erhöht oder erniedrigt.



Parameter

Parameter	DIREKTE KONSTANTE	#, #H	Zähler Betriebsart
#num	DIREKTE KONSTANTE	#, #H	Zähler Betriebsart
R	BINÄR	E, M, A, K, S	Rücksetzen des Zählers
S	BINÄR	E, M, A, K, S	Setzen des Zählers
INIT	WORT	EW, MW, AW, KW	Startwert des Zählers
RPI	BINÄR	E, M, A, K, S	Freigabe der CATCH-Eingänge mit Reset des Zählers
SP	BINÄR	E, M, A, K, S	Freigabe der CATCH-Eingänge ohne Reset des Zählers
R-Q	BINÄR	E, M, A, K, S	Rücksetzen des Ausganges Q
Q	BINÄR	M, A	Nulldurchgang des Zählers
CV	WORT	AW, MW	Aktueller Zählerstand
SPCV	WORT	AW, MW	Gespeicherter Zählerstand

Verfügbar ab:

Advant Controller 31, Serie 40 und 50

Beschreibung

Der Funktionsbaustein dient zum Zählen von Impulsen, die mit höherer Frequenz am Zähl Eingang anstehen. Es werden die drei Betriebsarten der Zähler in den Zentralgeräten der Serie 40 und 50 unterstützt.

Der Zählerstand lässt sich im Zähler mit und ohne Rücksetzen des Zählers abspeichern. Hierzu ist am jeweiligen CATCH-Eingang eine 0/1-Flanke anzulegen. Beim Zählvorgang und CATCH-Vorgang wird die positive Flanke des Impulses am Eingang ausgewertet. Die CATCH-Eingänge müssen mit RPI oder SP freigegeben werden.

In den Betriebsarten 1 und 2 erhöht sich der Zählerwert, in Betriebsart 3 wird er abhängig von der Flankenfolge an den beiden Eingängen (Drehrichtung: Encoder – Spuren A und B) erhöht oder erniedrigt.

#num Direkte Konstante

Wahl der Betriebsart: Der Zähler hat drei Betriebsarten:

#1: Zähler1 Zähleingang: E 62,00
 CATCH-Eingang: E 62,03

#2: Zähler2 Zähleingang: E 62,01
 CATCH-Eingang: E 62,03

#3: Incremental-Geber:

Die Zähleingänge E 62,00 und E 62,01 sind an einen Encoder mit Spur A und Spur B angeschlossen. Je nach Flankenfolge (Drehrichtung) wird der Zählwert um den Wert 1 erhöht oder erniedrigt.

CATCH-Eingang: E 62,02

Ist einer der beiden Kanäle defekt oder nicht angeschlossen, dann wechselt der Zählerwert ständig zwischen +1 und -1.

Mit den Betriebsarten 1 und 2 können in einem Programm gleichzeitig zwei schnelle Zähler arbeiten. Ist jedoch bereits ein schneller Zähler mit Betriebsart 3 programmiert, kann kein weiterer schneller Zähler mehr, gleich welcher Betriebsart, im Programm bearbeitet werden.

R BINÄR

Ein 1-Signal setzt den Wert des Zählers in CV und den in SPCV gespeicherten Wert auf 0. Dieser Eingang hat höchste Priorität.

$$R = 1 \rightarrow \begin{matrix} CV = 0 \\ SPCV = 0 \end{matrix}$$

S BINÄR

Ein 1-Signal setzt den Zähler auf den an INIT angegebenen Wert.

$$S = 1 \rightarrow CV = \text{INIT}$$

INIT WORT

Startwert für den schnellen Zähler.

RPI BINÄR

Ein 1-Signal gibt die CATCH-Eingänge frei. RPI hat höhere Priorität als SP.

RPI = 0 → Die CATCH-Eingänge sind gesperrt

RPI = 1 → Eine positive Flanke an den CATCH-Eingängen E 62,02 bzw. E 62,03 speichert den Wert in SPCV. Der Zähler wird zurückgesetzt.

SP BINÄR

Ein 1-Signal gibt die CATCH-Eingänge frei.

SP = 0 → Die CATCH-Eingänge sind gesperrt

SP = 1 → Eine positive Flanke an den CATCH-Eingängen E 62,02 bzw. E 62,03 speichert den Wert in SPCV. Der Zähler wird nicht zurückgesetzt.

R-Q BINÄR

Ein 1-Signal setzt den Ausgang Q zurück.

Q BINÄR

Der Zählerwert hat einen Nulldurchgang:

Q = 1, wenn CV von -1 nach 0 oder von 0 nach -1 wechselt.

CV WORT

Aktueller Wert des Zählers.

SPCV WORT

Durch SP bzw. RPI und die CATCH-Eingänge gespeicherter Wert.

Wertebereich:

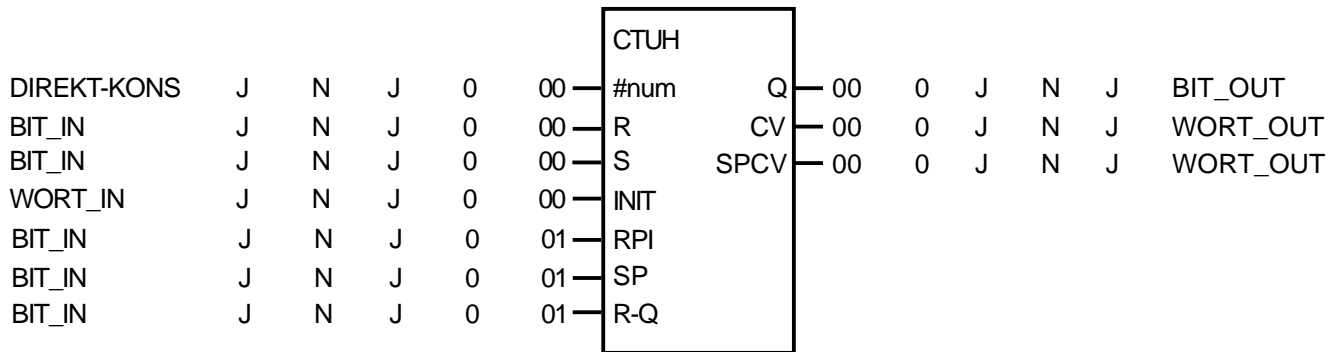
Integer Wort (16 Bit)

Untere Grenze: 8000_{H} -32768

Obere Grenze: $7FFF_{\text{H}}$ +32767

Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr.		Nr.	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	-----	--	-----	-----	-----	-----	-----	--------------



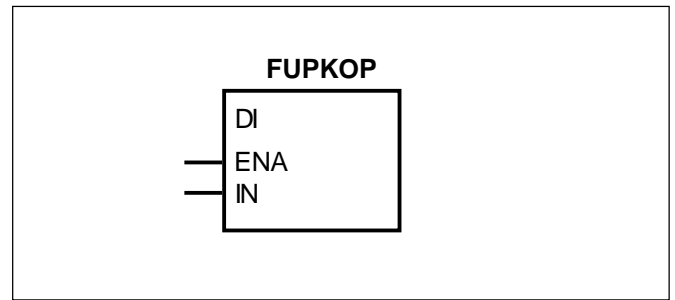
Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	!BA	000	
	00001	CUTH		
	00002	PP	00	#num
	00003	PP	00	R
	00004	PP	00	S
	00005	PP	00	INIT
	00006	PP	00	RPI
	00007	PP	00	SP
	00008	PP	00	R-Q
	00009	PP	00	Q
	00010	PP	00	CV
	00011	PP	00	SPCV

Der Funktionsbaustein DI liest *einen* binären Wert von einem Eingang der Zentraleinheit oder ihrer lokalen Erweiterung direkt ein. Dieser Eingang wird am Anschluß IN angegeben.

Diese Funktion ist nützlich,

- wenn die Zykluszeit zu lang ist
- wenn die Auslastung der Zentraleinheit zu hoch ist



Parameter

ENA	BIT	E, A, M, K, S	Freigabe (Enable)
IN	BIT	E	Eingang, der zu aktualisieren ist

Verfügbar ab: Advant Controller 31, Serie 40 und 50

Beschreibung

Am Anfang jedes Programm-Zyklus erstellt der SPS-Verarbeiter automatisch ein aktuelles Prozeßabbild aller Eingänge. Der Baustein dient dazu, während des Programmzyklus *einen* an IN ausgewählten Eingang im Prozeßabbild zu aktualisieren. Am Eingang IN dürfen nur die direkten Eingänge und die Eingänge der lokalen Erweiterung programmiert werden.

Dies kann bei speziellen Applikationen erforderlich sein, um Signaländerungen an den direkten Eingängen und den Eingängen der lokalen Erweiterung öfter als einmal pro Zyklus zu erfassen und zu verarbeiten.

Bei Verwendung der Funktionsbausteine DI und DO kann eine kürzere Klemme-Klemme-Reaktion an den ausgewählten direkten Ein-/Ausgängen erreicht werden.

Randbedingungen:

- empfehlenswert nur bei Stand-alone-SPS-Verarbeiter
- empfehlenswert nur bei großer Zykluszeit
- nur sinnvoll bei hoher Auslastung der SPS

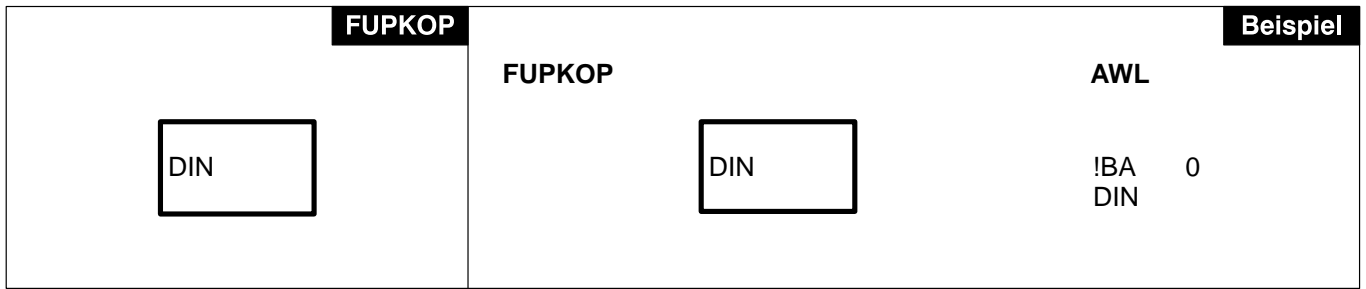
Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr	Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	----	-----	-----	-----	-----	--------------

BIT_IN	J	N	J	0	00	—	DI				
BIT_IN	J	N	J	0	00	—	ENA				
							IN				

Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	!BA	000	
	00001	DI		
	00002		PP 00	ENA
	00003		PP 00	IN



Beschreibung

Der Funktionsbaustein DIN liest mit Hilfe des CS31-Busprozessors alle *direkten* Eingänge des SPS-Verarbeiters ein. Die *direkten* Eingänge sind die Eingänge, die direkt an den Klemmen des SPS-Verarbeiters verfügbar sind.

Anwendung:

Am Anfang eines jeden Programm-Zyklus erstellt der SPS-Verarbeiter automatisch ein *aktuelles* Prozeßabbild der *direkten* Eingänge. Der Baustein dient dazu, *während* des Programmzyklus weitere Aktualisierungen des Prozeßabbildes der *direkten* Eingänge durchzuführen. Dies kann bei speziellen Applikationen erforderlich sein, um Signaländerungen an den *direkten* Eingängen öfter als einmal pro SPS-Zyklus zu erkennen und zu verarbeiten.

Bei Verwendung der Funktionsbausteine DIN und DOUT kann eine kürzere Klemme-Klemme-Reaktion an den *direkten Ein-Ausgängen* erreicht werden.

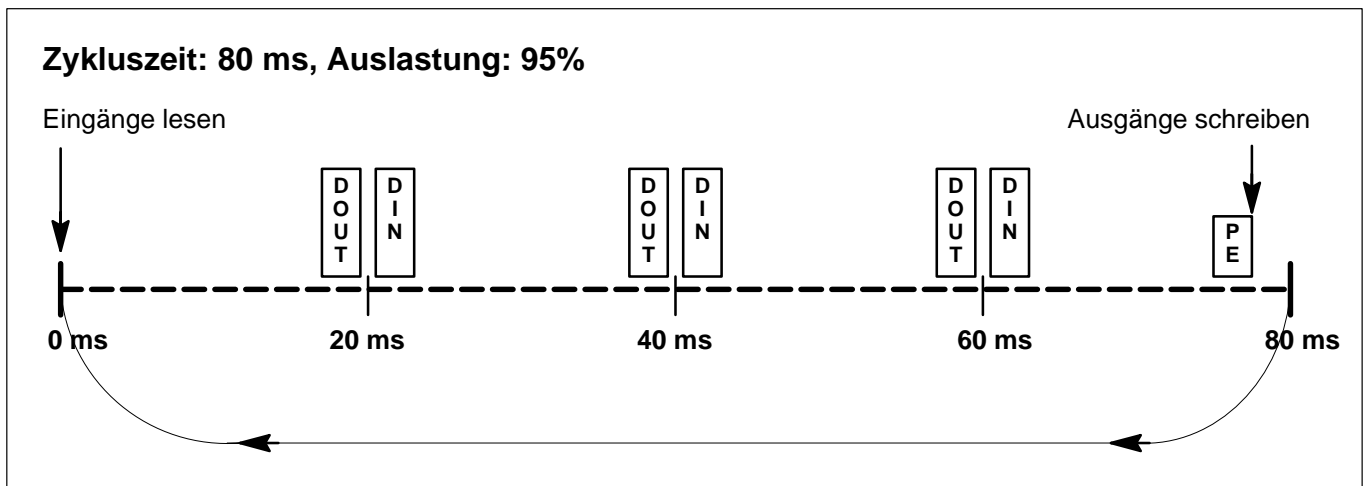
Der Funktionsbaustein ist empfehlenswert:

- bei Stand-alone-SPS-Verarbeitern
- bei großer Zykluszeit
- bei hoher Auslastung der SPS

Beispiel:

- Zykluszeit: 80 ms
- Auslastung: 95%
- 3 DIN + 3 DOUT Bausteine in gleichmäßigem zeitlichen Abstand über das SPS-Programm verteilt (d. h. nach 20 ms, 40 ms und 60 ms)

In diesem Beispiel werden die direkten Eingänge des SPS-Verarbeiters ca. alle 20 ms neu eingelesen. Die aktuellen Werte für die direkten Ausgänge werden im gleichen Zeitraster ausgegeben. Siehe zur Veranschaulichung die untenstehende Skizze.



VE-Daten

Laufzeit:

Stand-alone SPS:

07 KR 91, 07 KT 93:

07 KT 92:

zusätzlich bei MASTER- oder SLAVE-SPS:

Aktualisierung der Ausgänge:

Anzahl der Vergangenheitswerte:

Verfügbar ab:

< 0,2 ms

< 1 ms

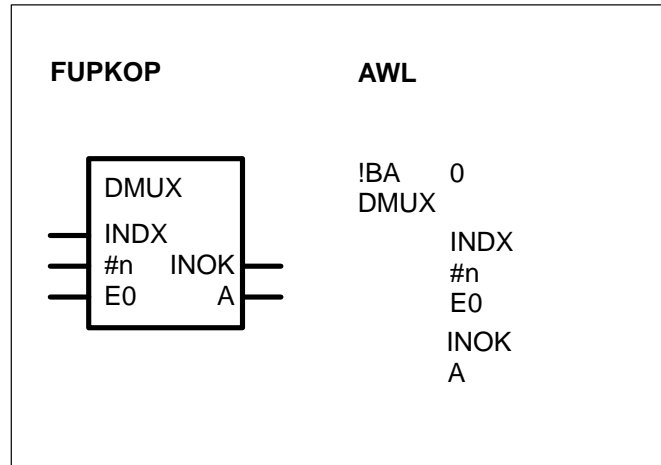
bis zu 4,8 ms

—

0 Worte

ABB Procontic CS31, 07 KT 93, 07 KT 92 d, 07 KR 91 d

Der Funktionsbaustein verbindet in Abhängigkeit vom Eingang INDX einen der Eingänge E0 ... E_{n-1} mit dem Ausgang A.



Parameter

INDX	WORT	EW, AW, MW, KW	Indexeingang
#n	DIREKTE KONSTANTE	#, #H	Anzahl n der Worteingänge E0 ... E _{n-1}
E0	WORT	EW, AW, MW, KW	doppelbarer Worteingang
INOK	BINÄR	A, M	Bereichsüberwachung des Index-Eingangs
A	WORT	AW, MW	Wortausgang, zu dem einer der Eingänge E0...E _{n-1} durchgeschaltet wird.

VE-Daten

Laufzeit:	
Grundlaufzeit:	36 µs
Zusätzliche Laufzeit:	keine
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein verbindet in Abhängigkeit vom Eingang INDX einen der Eingänge E0 ... E_{n-1} mit dem Ausgang A.

Der Wert am Eingang INDX wird auf Zulässigkeit überwacht. Befindet sich der Worteingang INDX nicht im gültigen Bereich, so wird der Ausgang A zu 0 gesetzt. Zusammenhang zwischen E0...E_{n-1}, INDX und A:

Mit dem Eingang INDX wird festgelegt, welcher der Eingänge E0 ... E_{n-1} mit dem Ausgang A verbunden wird.

Es gilt:

- INDX = 1 : E0 → A
- INDX = 2 : E1 → A
- INDX = 3 : E2 → A
- : : :
- INDX = n : E_{n-1} → A

mit: $1 \leq \text{INDX} \leq n \leq 32767$ (theoretisch)

Ist der Eingang INDX = 0, so wird der Ausgang A zu Null gesetzt.

INDX WORT
 Indexeingang zur Auswahl von einem der Eingänge E0 ... E_{n-1}.

Es gilt:
 $1 \leq \text{INDX} \leq n$ (Anzahl der Eingänge E0 ... E_{n-1})

Anmerkung:
 INDX = 0 → Initialisierung des Ausgangs A mit 0.

#n DIREKTE KONSTANTE
Anzahl n der Worteingänge $E_0 \dots E_{n-1}$

E WORT
Doppelbarer Eingang $E_0 \dots E_{n-1}$

Einer der n Eingänge $E_0 \dots E_{n-1}$ wird mit dem Ausgang A verbunden.

INOK BINÄR
Bereichsüberwachung für den Eingang INDX

Der Ausgang INOK zeigt an, ob sich der Eingang INDX im gültigen Bereich befindet. Befindet sich der Wort-eingang INDX nicht im gültigen Bereich, so werden die Ausgänge INOK und A zu 0 gesetzt.

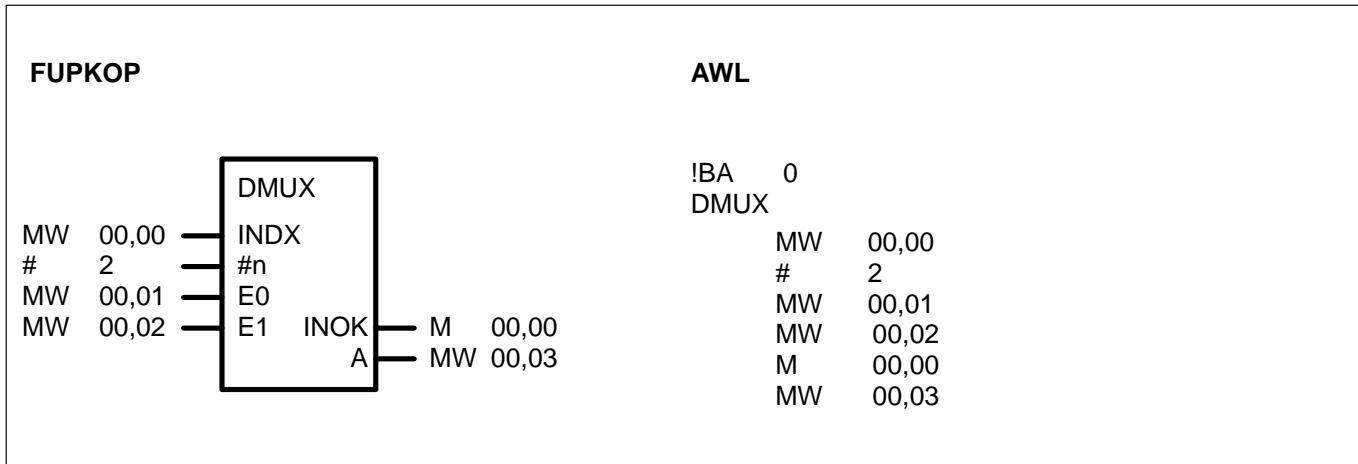
Es gilt:
 $INOK = 1 \rightarrow$ INDX im gültigen Bereich
 $INOK = 0 \rightarrow$ INDX im ungültigen Bereich $\rightarrow A = 0$

gültiger Bereich für den Index: $1 \leq INDX \leq n$

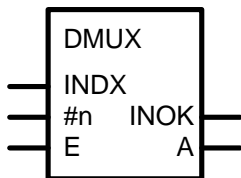
A WORT
Ausgang, zu dem einer der Eingänge $E_0 \dots E_{n-1}$ durchgeschaltet wird.

Mit dem Eingang INDX wird einer der Eingänge $E_0 \dots E_{n-1}$ ausgewählt und sein Wert dem Ausgang A zugewiesen. Befindet sich der Worteingang INDX nicht im gültigen Bereich, so wird der Ausgang A zu Null gesetzt.

Beispiel



VE-FUP-Definition

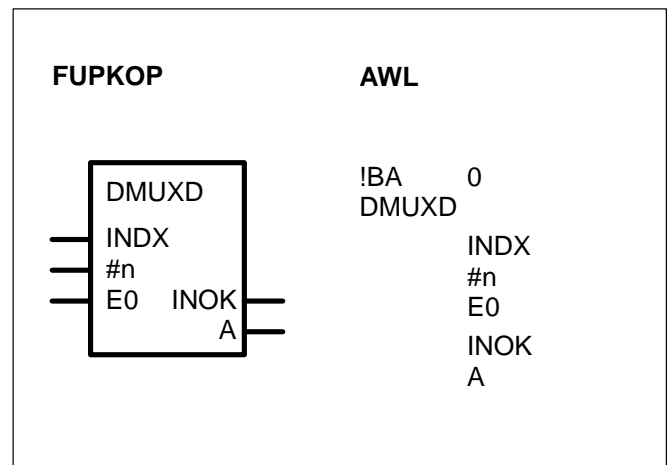


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
INDX	E	W	N	P	J	0	0
#n	K	W	N	P	J	0	0
E	E	W	N	P	J	1	0
INOK	A	L	N	P	J	0	0
A	A	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr (vorbelegt 0)			
00001	DMUX						
00002		PP 0	INDX	Index		(WORT)	
00003		PP 0	#n	# Anzahl		(#)	
	[1					
00004		PP 1	E	Eingang		(WORT)	
]	1					
00005		PP 0	INOK	Ausgang Index gültig		(BINÄR)	
00006		PP 0	A	Ausgang		(WORT)	

Der Funktionsbaustein verbindet in Abhängigkeit vom Eingang INDX einen der Eingänge E0 ... E_{n-1} mit dem Ausgang A.



Parameter

INDX	WORT	EW, AW, MW, KW	Indexeingang
#n	DIREKTE KONSTANTE	#, #H	Anzahl n der Doppelwort-Eingänge E0 ... E _{n-1}
E0	DOPPELWORT	MD, KD	Doppelwort-Eingänge E0 ... E _{n-1} ; der Eingang E0 ist doppelbar.
NOK	BINÄR	A, M	Bereichsüberwachung des Index-Eingangs
A	DOPPELWORT	MD	Doppelwort-Ausgang, zu dem einer der Eingänge E0 ... E _{n-1} durchgeschaltet wird.

VE-Daten

Laufzeit:

Grundlaufzeit:	38 µs
Zusätzliche Laufzeit:	keine
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein verbindet in Abhängigkeit vom Eingang INDX einen der Eingänge E0 ... E_{n-1} mit dem Ausgang A.

Der Wert am Eingang INDX wird auf Zulässigkeit überwacht. Befindet sich der Worteingang INDX nicht im gültigen Bereich, so wird der Ausgang A zu 0 gesetzt.

Zusammenhang zwischen E0 ... E_{n-1}, INDX und A:

Mit dem Eingang INDX wird festgelegt, welcher der Eingänge E0 ... E_{n-1} mit dem Ausgang A verbunden wird.

Es gilt:

INDX = 1	: E0	→ A
INDX = 2	: E1	→ A
INDX = 3	: E2	→ A
:	:	:
INDX = n	: E _{n-1}	→ A

mit: $1 \leq \text{INDX} \leq n \leq 32767$ (theoretisch)

Anmerkung: Ist der Eingang INDX = 0, so wird der Ausgang A zu Null gesetzt.

INDX WORT
Indexeingang zur Auswahl von einem der Eingänge E0 ... E_{n-1}.

Es gilt: $1 \leq \text{INDX} \leq n$ (Anzahl der Eingänge E0 ... E_{n-1})

Anmerkung:
INDX = 0 → Initialisierung des Ausgangs A mit 0.

#n DIREKTE KONSTANTE
Anzahl n der Doppelwort-Eingänge E0 ... E_{n-1}.

E DOPPELWORT

Doppelbarer Eingang $E_0 \dots E_{n-1}$.

Einer der n Eingänge $E_0 \dots E_{n-1}$ wird mit dem Ausgang A verbunden.

INOK BINÄR

Bereichsüberwachung für den Eingang $INDX$

Der Ausgang $INOK$ zeigt an, ob sich der Eingang $INDX$ im gültigen Bereich befindet. Befindet sich der Wort-eingang $INDX$ nicht im gültigen Bereich, so werden die Ausgänge $INOK$ und A gleich 0 gesetzt.

Es gilt:

$INOK = 1 \rightarrow INDX$ im gültigen Bereich

$INOK = 0 \rightarrow INDX$ im ungültigen Bereich $\rightarrow A = 0$

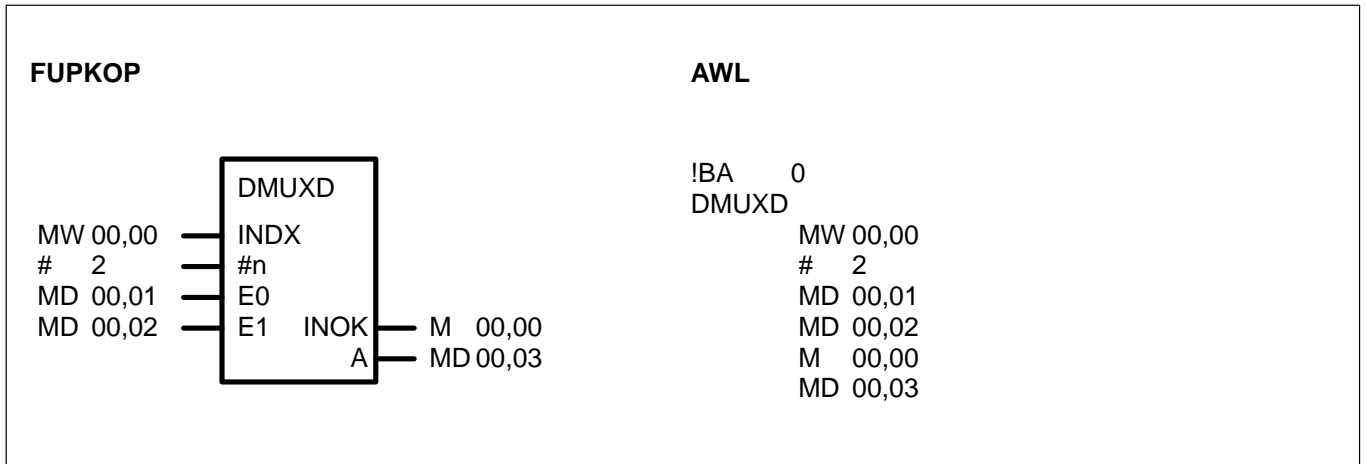
Gültiger Bereich für den Index: $1 \leq INDX \leq n$

A DOPPELWORT

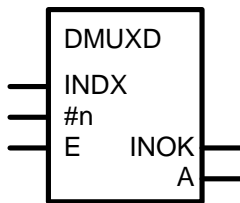
Ausgang, zu dem einer der n Eingänge $E_0 \dots E_{n-1}$ durchgeschaltet wird.

Mit dem Eingang $INDX$ wird einer der Eingänge $E_0 \dots E_{n-1}$ ausgewählt und sein Wert dem Ausgang A zugewiesen. Befindet sich der Eingang $INDX$ nicht im gültigen Bereich, so wird der Ausgang A zu Null gesetzt.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
INDX	E	W	N	P	J	0	0
#n	K	W	N	P	J	0	0
E	E	D	N	P	J	1	0
INOK	A	L	N	P	J	0	0
A	A	D	N	P	J	0	0

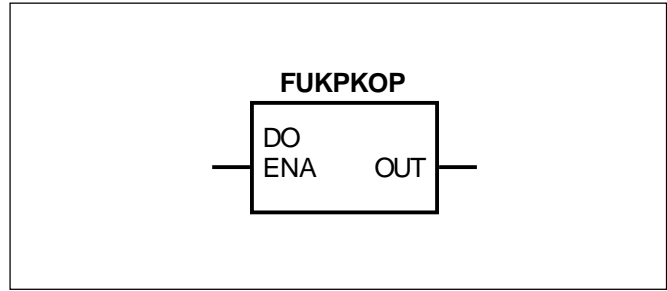
VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DMUXD			
00002		PP 0	INDX	Index (WORT)
00003		PP 0	#n	Anzahl (#)
	[1		
00004		PP 1	E	Eingang (DOPPELWORT)
]	1		
00005		PP 0	INOK	Ausgang Index gültig (BINÄR)
00006		PP 0	A	Ausgang (DOPPELWORT)

Der Funktionsbaustein DO gibt *einen* binären Wert an einem direkten Ausgang der Zentraleinheit oder ihrer lokalen Erweiterung direkt aus. Dieser Ausgang wird am Anschluß OUT angegeben.

Diese Funktion ist nützlich,

- wenn die Zykluszeit zu lang ist
- wenn die Auslastung der Zentraleinheit zu hoch ist



Parameter

ENA	BIT	E, A, M, K, S	Freigabe (Enable)
OUT	BIT	A	Ausgang, der zu aktualisieren ist

Verfügbar ab: Advant Controller 31, Serie 40 und 50

Beschreibung

Während der Programmbearbeitung erstellt der SPS-Verarbeiter ein aktuelles Prozeßabbild aller Ausgänge. Am Zyklusende werden die Werte vom Ausgangsabbild auf die einzelnen Ausgangsadressen gelegt.

Der Baustein dient dazu, *während* des Programmzyklus *einen* an OUT ausgewählten Ausgang zu aktualisieren.

Am Ausgang OUT dürfen nur die direkten Ausgänge und die Ausgänge der lokalen Erweiterung programmiert werden.

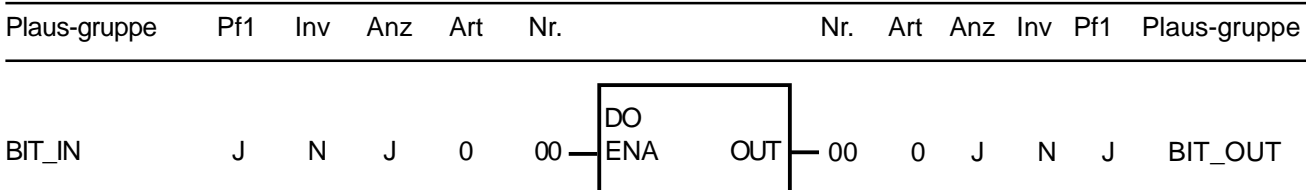
Dies kann bei speziellen Applikationen erforderlich sein, um die Ausgangs-Signaländerungen öfter als einmal pro SPS-Zyklus oder schneller über die direkten/lokalen Ausgänge auszugeben.

Bei Verwendung der Funktionsbausteine DI und DO kann eine kürzere Klemme-Klemme-Reaktion an den ausgewählten direkten Ein-/Ausgängen erreicht werden.

Randbedingungen:

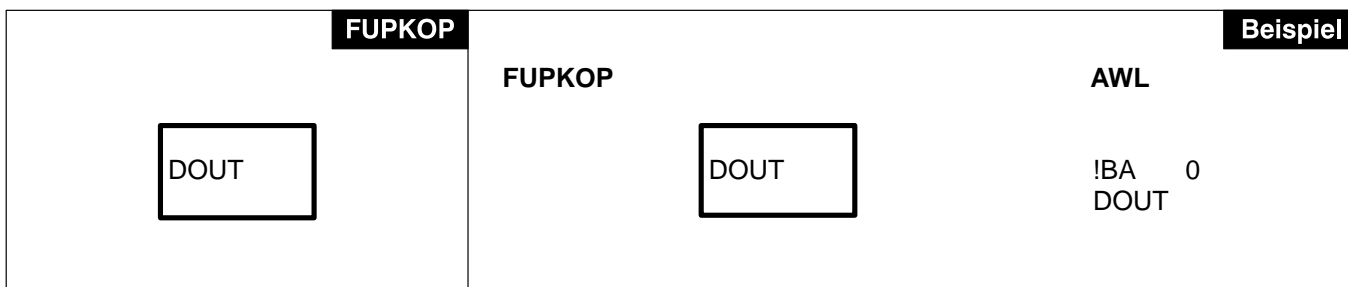
- empfehlenswert nur bei Stand-alone-SPS-Verarbeitern
- empfehlenswert nur bei großer Zykluszeit
- nur sinnvoll bei hoher Auslastung der SPS

Grafik



Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	IBA 000		
	00001	DO		
	00002	PP 00	ENA	
	00003	PP 00	OUT	



Beschreibung

Der Funktionsbaustein DOUT schreibt mit Hilfe des CS31-Busprozessors alle *direkten* Ausgänge des SPS-Verarbeiters. Die *direkten* Ausgänge sind die Ausgänge, die direkt an den Klemmen des SPS-Verarbeiters verfügbar sind.

Anwendung:

Am Ende eines jeden Programm-Zyklus gibt der SPS-Verarbeiter automatisch das während des Programmzyklus erstellte Ausgangs-Prozeßabbild aus. Der Baustein dient dazu, *schon während* des Programmzyklus das momentane Ausgangs-Prozeßabbild für die *direkten* Ausgänge auszugeben. Dies kann bei speziellen Applikationen erforderlich sein, um Ausgangs-Signaländerungen öfter als einmal pro SPS-Zyklus über die direkten Ausgänge zur Verfügung zu haben.

Bei Verwendung der Funktionsbausteine DIN und DOUT kann eine kürzere Klemme-Klemme-Reaktion an den *direkten Ein-/Ausgängen* erreicht werden.

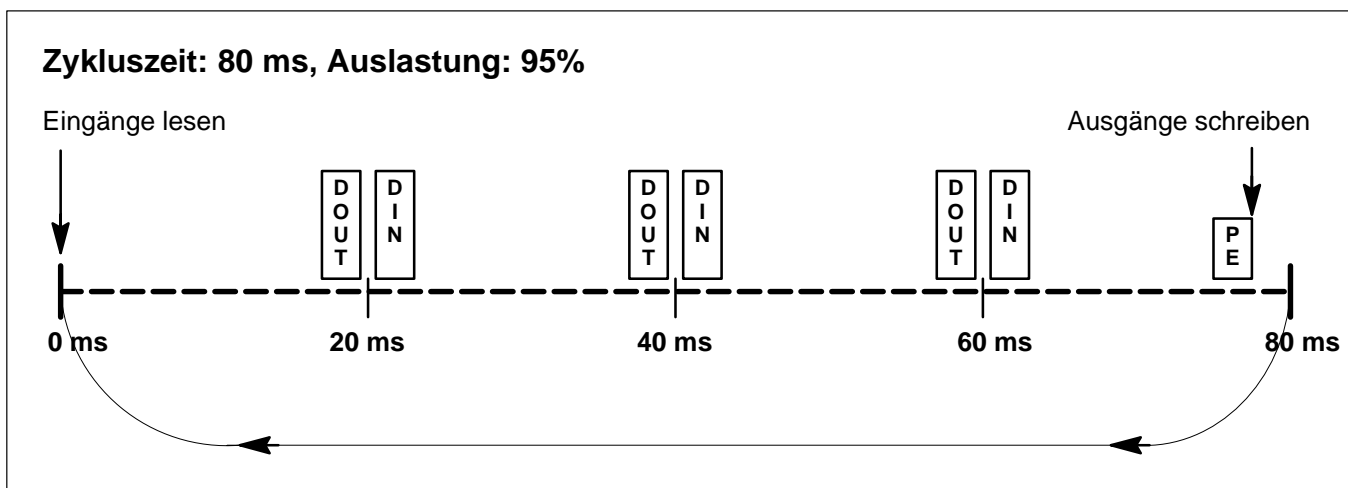
Der Funktionsbaustein ist empfehlenswert:

- bei Stand-alone-SPS-Verarbeitern
- bei großer Zykluszeit
- bei hoher Auslastung der SPS

Beispiel:

- Zykluszeit: 80 ms
- Auslastung: 95%
- 3 DIN + 3 DOUT Bausteine in gleichmäßigem zeitlichen Abstand über das SPS-Programm verteilt (d. h. nach 20 ms, 40 ms und 60 ms)

In diesem Beispiel werden die direkten Eingänge des SPS-Verarbeiters ca. alle 20 ms neu eingelesen. Die aktuellen Werte für die direkten Ausgänge werden im gleichen Zeitraster ausgegeben. Siehe zur Veranschaulichung die untenstehende Skizze.



VE-Daten

Laufzeit:

Stand-alone SPS:

07 KR 91, 07 KT 93:

07 KT 92:

zusätzlich bei MASTER- oder SLAVE-SPS:

< 0,2 ms

< 1 ms

bis zu 4,8 ms

Aktualisierung der Ausgänge:

—

Anzahl der Vergangenheitswerte:

0 Worte

Verfügbar ab:

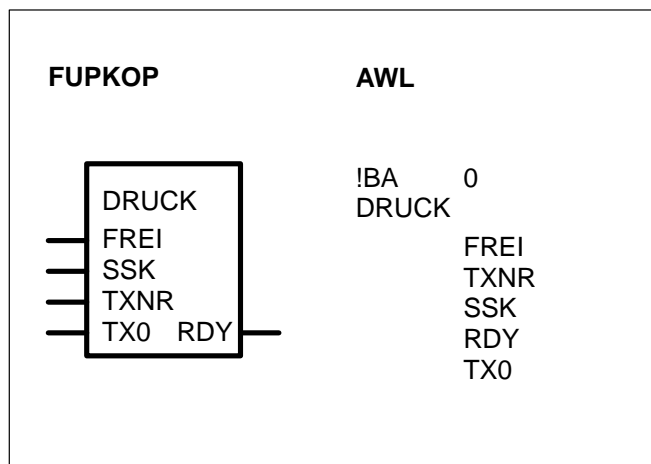
ABB Procontic CS31, 07 KT 93, 07 KT 92 d, 07 KR 91 d

AUSGABE VON ASCII-ZEICHEN UND HEXWERTEN ÜBER EINE SERIELLE SCHNITTSTELLE

DRUCK

Der Funktionsbaustein DRUCK gibt im SPS-Programm abgelegte Texte und Werte von Operanden über eine serielle Schnittstelle aus.

Durch Vorgabe einer Formatkennung werden die auszugebenden Zahlenwerte für mannigfaltige Darstellungsformen aufbereitet.



Parameter

Parameter	Format	Werte	Bedeutung
FREI	BINÄR	A, E, S, M	Freigabe für die Ausgabe eines Textes (0→1-Flanke)
SSK	WORT	AW, EW, MW, KW	Schnittstellenkennung; Nummer der seriellen Schnittstelle
TXNR	WORT	AW, EW, MW, KW	Textnummer; Nummer des auszugebenden Textes
TX0	ALLE	Texte	Texte; doppelbarer Eingang für die Texte
RDY	BINÄR	A, E, M	Ready, Baustein ist bereit zur Ausgabe eines Textes

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:	liegt nicht vor	374 µs
Zusätzliche Laufzeit:	liegt nicht vor	63 µs pro Zeichen
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	1 Wort	1 Wort
Verfügbar ab:	ABB Procontic CS31 / Kommunikationsmodul 07 KP 62	07 KR 31 V 2.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein DRUCK gibt im SPS-Programm abgelegte Texte und Werte von Operanden über eine serielle Schnittstelle aus.

Durch Vorgabe einer Formatkennung werden die auszugebenden Zahlenwerte für mannigfaltige Darstellungsformen aufbereitet.

Die auszugebenden Texte und Operanden-Kennzeichen werden in der SPS direkt hinter dem Baustein DRUCK im Anwenderprogramm abgelegt.

Die Texte haben zur Identifizierung jeweils eine Text-Nummer.

Der DRUCK-Baustein schreibt die auszugebenden Texte und Operandenwerte in einen Buffer, von wo sie ein Schnittstellentreiber abholt und über die serielle Schnittstelle ausgibt. Der Schnittstellentreiber interpretiert dabei

die Anzeigeformate für die Operandenwerte und wandelt diese vor der Ausgabe in das gewünschte Anzeige-Format.

Initialisierung der seriellen Schnittstelle

Bevor der Baustein DRUCK mit einer der möglichen seriellen Schnittstellen kommunizieren kann, muß diese mit dem Baustein SINIT initialisiert werden.

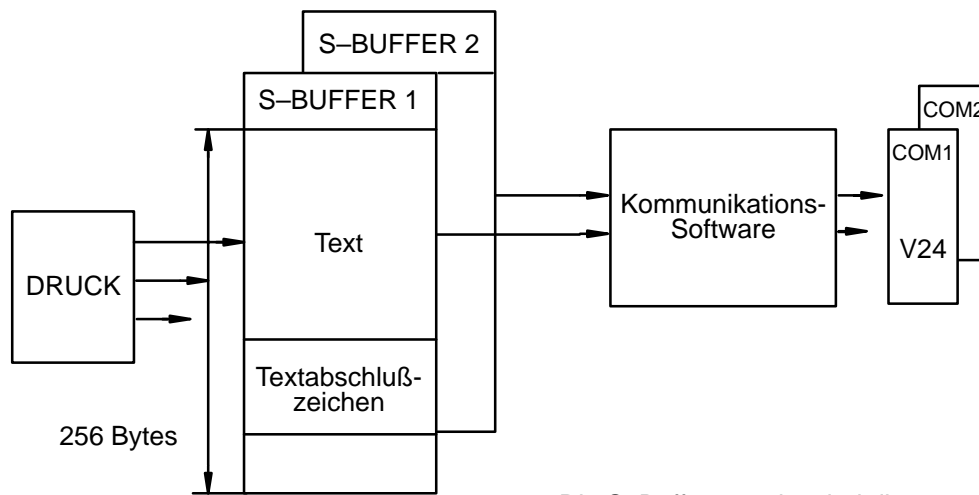
Beim Baustein SINIT wird außer den gewünschten Schnittstellenparametern ein frei projektierbares Text-Abschlußzeichen angegeben, um das Ende des zu sendenden Gesamttextes/Telegramms zu markieren. Ob das Textabschlußzeichen über die Schnittstelle ausgegeben werden soll, wird ebenfalls beim Baustein SINIT angegeben.

Die Synchronisation des Datenflusses erfolgt prinzipiell über die Signale RTS und CTS.

Serielle Schnittstellen der Zentraleinheiten

Folgende serielle Schnittstellen können vom Baustein
DRUCK benutzt werden:

Zentraleinheit 07 KR 91:	COM1
Zentraleinheit 07 KT 92:	COM1, COM2
Kommunikationsprozessor 07 KP 62:	COM1, COM2
Zentraleinheit 07 KR 31:	COM1



Die S-Buffer 1 und 2 sind die
Sende-Buffer, die den
seriellen Schnittstellen COM1
und COM2 zugeordnet sind.

AUSGABE VON ASCII-ZEICHEN UND HEXWERTEN ÜBER EINE SERIELLE SCHNITTSTELLE

DRUCK

Bausteinparameter Kurzübersicht

FREI	RDY	TXNR	SSK	BEDEUTUNG
0/1-FLANKE	1	2	1	Der Text mit der Nr.2 wird über die Schnittstelle ¹ ausgegeben
0/1-FLANKE	0	2	1	0/1-Flanke wird ignoriert, da Baustein nicht bereit
keine 0/1-FLANKE	X	X	X	Keine Ausgabe eines neuen Textes

FREI BINÄR

Eine 0→1-Flanke am Eingang FREI bewirkt die Ausgabe *eines* Textes, sofern der Baustein dazu bereit ist (RDY = 1).

Erfolgt eine 0→1-Flanke am Eingang FREI, obwohl der Ausgang RDY gleich 0 ist, d. h. der Baustein ist noch nicht bereit für eine neue Übertragung, so wird die 0→1-Flanke *ignoriert*. Solange das RDY-Signal auf 0 ist, kann also keine neue Textübertragung gestartet werden.

SSK WORT

Am Eingang SSK wird die Nummer der Schnittstelle angegeben, über die der Text ausgegeben werden soll. Es gilt:

COM1: Nummer = 1
COM2: Nummer = 2

TXNR WORT

Am Eingang TXNR wird die Nummer des auszugebenden Textes angegeben.

Bei einem DRUCK-Baustein sind 1 ... 99 Texte möglich. Die Nummer des auszugebenden Textes muß solange am Eingang TXNR anstehen, bis der Baustein das Ende der Textübertragung mit 1-Signal am BausteinAusgang RDY anzeigt.

RDY BINÄR

Beim Programmzyklus, bei dem der Baustein zum ersten Mal aufgerufen wird, und während der Zeit, in der ein Text ausgegeben wird, gilt:

RDY = 0.

Solange RDY gleich 0 ist kann keine neue Textausgabe aktiviert werden, und eine evtl. anstehende 0→1-Flanke am Eingang FREI wird ignoriert und ist verloren.

Nach dem ersten Aufruf des Bausteins bzw. nach Beendigung einer Text-Ausgabe, gilt:

RDY = 1

Der Baustein ist nun bereit zur Ausgabe eines neuen Textes.

Das RDY-Signal kann z. B. dazu benutzt werden, eine neue Textübertragung mit dem Eingang FREI zu aktivieren.

Ablage der Texte in der Zentraleinheit

Ablage: Direkt hinter dem Baustein DRUCK

Anzahl: 1 ... 99

Länge: max. 256 Zeichen (wegen Sendebufferlänge = 256)

Dabei zählt für die Länge der Texte:

- 07 KR 91, 07 KT 92, 07 KT 93
 - Jedes Textzeichen als 1 Zeichen
 - Jede Formatkennung als 3 Zeichen
 - Jeder Bit-Operand als 1 Zeichen
 - Jeder Wort-Operand als 2 Zeichen
 - Jeder Doppelwort-Operand als 4 Zeichen
- 07 KR 31
 - Jedes Textzeichen als 1 Zeichen
 - Jeder Bit-Operand als 1 Zeichen
 - Jeder Wort-Operand als 1 bis 6 Zeichen *
 - Jeder Doppelwort-Operand als 10 bis 11 Zeichen *

Die SPS überprüft beim Programmstart die Texte auf Überschreitung der maximalen Länge.

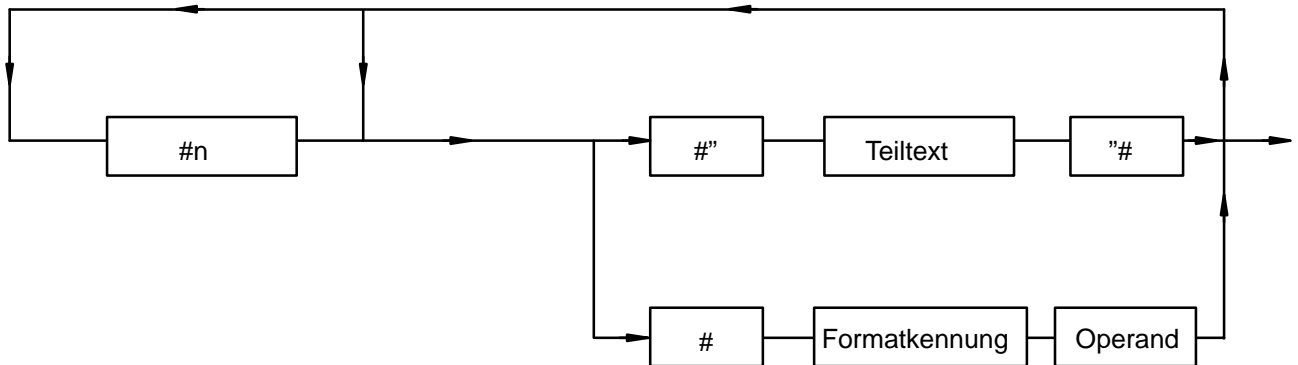
* abhängig vom Format

Syntax (Aufbau) der Texte

Ein Text für den DRUCK-Baustein besteht aus:

- der Text-Nummer
- einem oder mehreren Teiltexten (optional)
- Operanden mit Formatkennung (optional)

Syntaxdiagramm der Texte



#n : Nummer des einzugebenden Textes als direkte Konstante (1 ... 99)

#\" : Anfangs-Kennung für Texteingabe

\"# : Ende-Kennung für Texteingabe (nur im Terminalbetrieb)

Achtung: Diese Endekennung wird bei Verwendung des ABB Procontic Programmiersystems *nicht* angegeben. Nur wenn im Terminalbetrieb Texte direkt in die SPS eingegeben werden, müssen diese mit dieser Kennung abgeschlossen werden.

Teiltext: Alle ASCII-Zeichen 0...7F.
Das ASCII-Zeichen <NUL> ist eine Ausnahme für 07 KR 91 / 07 KT 92 / 07 KT 93. Dies Zeichen wird als Präfix für die Formatkennung im Sendebuffer benutzt.

Formatkennung: Die Formatkennung gibt den Typ des Operanden und das Anzeigeformat seines Wertes an. Die Formatkennung ist eine 3-stellige Zahl. Die 1. Ziffer von links gibt den Operandentyp an. Es gibt 3 Operandentypen:

Binär: 1
Wort: 2
Doppelwort: 3

Die Ziffern 2 und 3 definieren das Anzeigeformat.

Operand: Binär-, Wort- oder Doppelwort-Operanden, deren Werte je nach Anzeigeformat ausgegeben werden.

Eingabe der Texte

Bei der Eingabe der Texte sind zwei Fälle zu unterscheiden:

- Eingabe der Texte in das ABB Procontic Programmiersystem
- Eingabe der Texte mit einem Terminal direkt in die SPS

Eingabe der Texte in das ABB Procontic Programmiersystem

Bei der Eingabe in das Programmiersystem werden folgende Teile des Gesamttextes als eigenständige Operanden behandelt:

- die Text-Nummer z.B. # 1
- ein Teiltext z.B. #" Text1
- eine Formatkennung z.B. # 203
- ein Operand z.B. MW 2,3
- ein weiterer Teiltext z.B. #" Text2

Im Funktionsplan belegt beim Symbol des DRUCK-Bausteins jeder dieser Teile einen der Eingänge TX0 ... TXn.

Eingabe von Sonderzeichen für Bildschirm bzw. Druckersteuerung:

Zur Gliederung des Textes bei der Ausgabe auf einen Bildschirm oder Drucker sind Steuerzeichen wie z.B. "Zeilenvorschub" <LF> oder "Zeilen-Anfang" <CR> erforderlich. Diese Sonderzeichen können innerhalb eines Teiltextes an einer beliebigen Stelle eingefügt werden. Beim Programmiersystem erfolgt die Eingabe dieser Sonderzeichen durch:

\Zahlenwert des Zeichens

Der Zahlenwert des Zeichens wird als dreistellige Dezimalzahl angegeben.

Beispiel:

Auf einem Drucker soll folgende Ausgabe gemacht werden:

Erste Zeile
Leerzeile
Zweite Zeile

Dazu ist die Projektierung des folgenden Textes notwendig:

Erste Zeile <CR> <LF> <LF> Zweite Zeile

Es gilt:

<CR> = 013
<LF> = 010

Die Eingabe des Textes in das Programmiersystem sieht wie folgt aus:

#"Erste Zeile\013\010\010Zweite Zeile

Hinweis:

Die Zeichen mit ASCII-Code >20_H bzw. 32_D müssen über die Tastatur eingegeben werden. So muß beispielsweise anstatt \033 das diesem Code entsprechende "!" eingegeben werden.

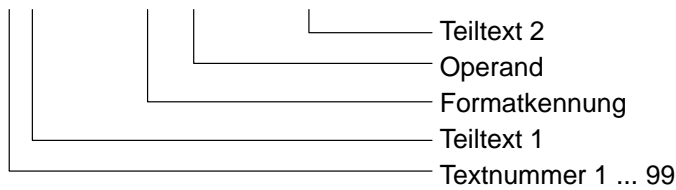
Zeichen, die keine Sonderzeichen sind und auch nicht direkt über die Tastatur eingegeben werden können, werden durch Drücken der <ALT>-Taste und Eingeben des Zahlencodes (numerische Tastatur) eingetragen.

Das Zeichen mit dem ASCII-Code 255 ist innerhalb der Programmiersoftware reserviert und darf nicht verwendet werden.

Beispiel

Beim DRUCK-Baustein ist der Text mit der laufenden Nummer 1 zu projektieren. Der Text besteht aus zwei Teiltextrn (Text1 und Text2). Zwischen diesen Teiltextrn soll der Wert des Wortmerkers MW 2,3 dreistellig (Format 03) ausgegeben werden.

```
#1#"Text1#203MW02,03#"Text2
```



FUPKOP



AWL

```
!BA 0
DRUCK
M 00,00
MW 00,01
MW 00,00
A 00,00
# 1
#"TEXT1
# 203
MW 02,03
#"TEXT2
```

Es können noch weitere Texte mit Nr. 2 ... Nr. 99 projiziert werden. Der Text mit der laufenden Nr. 2 wird in analoger Weise ab dem Eingang TX5 projiziert. Dabei wird am Eingang TX5 die Nummer des nächsten Textes (#2) angegeben.

Eingabe der Texte mit einem Terminal direkt in die SPS

Steht dem Anwender kein ABB Procontic Programmiersystem zur Verfügung, so kann er sein SPS-Programm als Anweisungsliste (AWL) mit einem Terminal direkt in die SPS eingeben bzw. aus ihr auslesen.

Die Eingabe der Texte erfolgt dabei direkt im Anschluß an den DRUCK-Baustein, d.h. der Text wird ab der Programmspeicheradresse, die auf den Operanden am Ausgang RDY folgt, eingeben.

Eingabe der Texte direkt in die SPS

Der Anwender muß bei der direkten Eingabe von Texten in die SPS die Syntaxregeln für den Textaufbau beachten.

Ein Text für den DRUCK-Baustein besteht aus:

- der Text-Nummer
- einem oder mehreren Teiltexen (optional)
- Operanden mit Formatkennung (optional)

Die Eingabe von Texten in die SPS erfolgt innerhalb der Betriebsart "Anwenderprogramm eingeben/ändern" (s.a. Kapitel "Bedien- und Testfunktionen", Systembeschreibung ABB Procontic CS31, Griff 7.3, 'Programmier- und Diagnosegerät TCZ').

Die Eingabe der Texte bzw. Teiltexen (Sonderzeichen sind auch möglich) erfolgt eingebettet in die Sonderzeichen '#' (Anfang der Texteingabe) und '# (Ende der Texteingabe).

Merkmale: Texte für den Baustein DRUCK

Ablage: direkt hinter dem Baustein DRUCK

Anzahl: 1 ... 99

Länge: max. 256 Zeichen (wegen Sendebufferlänge = 256)

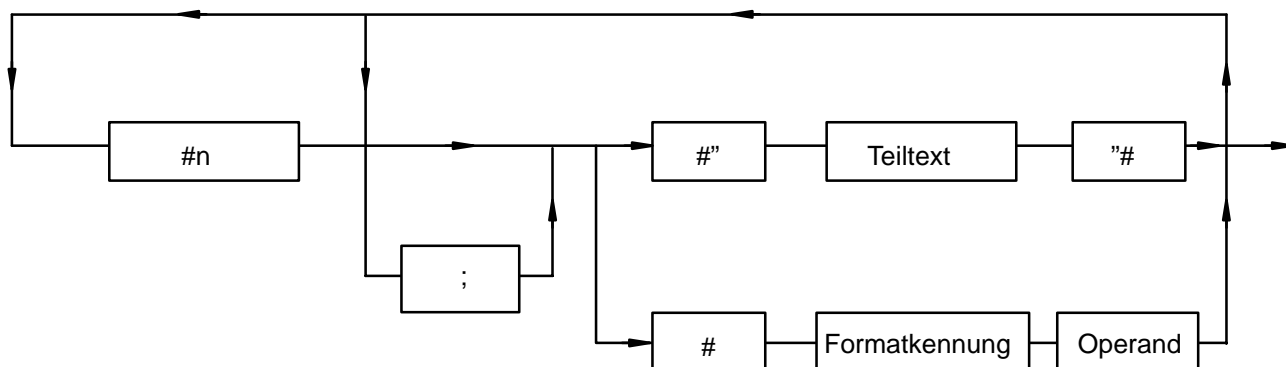
Dabei zählt für die Länge der Texte:

- 07 KR 91, 07 KT 92, 07 KT 93
 - Jedes Textzeichen als 1 Zeichen
 - Jede Formatkennung als 3 Zeichen
 - Jeder Bit-Operand als 1 Zeichen
 - Jeder Wort-Operand als 2 Zeichen
 - Jeder Doppelwort-Operand als 4 Zeichen
- 07 KR 31
 - Jedes Textzeichen als 1 Zeichen
 - Jeder Bit-Operand als 1 Zeichen
 - Jeder Wort-Operand als 1 bis 6 Zeichen *
 - Jeder Doppelwort-Operand als 10 bis 11 Zeichen *

Die Funktion Programmaufbereitung (PA) überprüft die Texte auf Überschreitung der maximalen Länge.

* abhängig vom Format

Syntaxdiagramm der Texte



#n : Nummer des einzugebenden Textes als direkte Konstante (1 ... 99).

;" : Anfangs-Kennung für Texteingabe

"# : Ende-Kennung für Texteingabe im Terminalbetrieb

Achtung: Diese Endekennung wird bei Verwendung des ABB Procontic Programmiersystems *nicht* angegeben. Nur wenn im Terminalbetrieb

Texte direkt in die SPS eingegeben werden, müssen diese mit dieser Kennung abgeschlossen werden.

Teilttext:

Alle ASCII-Zeichen 0...7F.

Das ASCII-Zeichen <NUL> ist eine Ausnahme für 07 KR 91 / 07 KT 92 / 07 KT 93. Dies Zeichen wird als Präfix für die Formatkennung im Sendebuffer benutzt.

Formatkennung: Die Formatkennung gibt den Typ des Operanden und das Anzeigeformat seines Wertes an. Die Formatkennung ist eine 3-stellige Zahl. Die 1. Ziffer von links gibt den Operandentyp an. Es gibt 3 Operandentypen:

Binär: 1
Wort: 2
Doppelwort: 3

Die Ziffern 2 und 3 definieren das Anzeigeformat.

Operand: Binär-, Wort- oder Doppelwort-Operanden, deren Werte je nach Anzeigeformat ausgegeben werden.

Korrektur eines Zeichens bei der Texteingabe mit einem Terminal

Bei der direkten Eingabe eines Textes mit einem Terminal kann ein fehlerhaftes Zeichen *nicht* korrigiert werden. Zur Korrektur muß die Texteingabe mit der Text-eingabe-Endekennung "# abgeschlossen und die Programmeingabe mit <CR> beendet werden. Mit der Programmeingabe ist bei der Adresse des zu korrigierenden Zeichens neu aufzusetzen. Die Texteingabe beginnt dann mit der Texteingabe-Anfangskennung "#".

Eingabe von Texten mit mehr als 80 Zeichen mit einem Terminal

Auf dem Bildschirm können bei einer Texteingabe max. 80 Zeichen pro Bildschirmzeile angezeigt werden. Ist der Text länger als 80 Zeichen, so muß die Texteingabe am Ende der Bildschirmzeile mit der Texteingabe-Endekennung "# abgeschlossen werden und mit dem Semicolon (;) die nächste Bildschirmzeile angesprungen werden. Mit der Texteingabe-Anfangskennung "# wird nun die Texteingabe fortgesetzt.

Kommunikation mehrerer DRUCK-Bausteine mit derselben seriellen Schnittstelle

Mehrere DRUCK-Bausteine können *dieselbe* serielle Schnittstelle benutzen, ohne daß besondere Vorkehrungen zu treffen sind. Ist die serielle Schnittstelle durch einen DRUCK-Baustein belegt, so warten die anderen DRUCK-Bausteine automatisch bis die Schnittstelle wieder frei ist.

– Wollen mehrere DRUCK-Bausteine gleichzeitig auf dieselbe Schnittstelle zugreifen, so kommt der DRUCK-Baustein zum Zuge, der im Anwenderprogramm am weitesten vorn steht. Die Priorität entspricht also der Reihenfolge, in der die DRUCK-Bausteine im Anwenderprogramm aufgerufen werden.

– Wird vom Anwender eine andere Bearbeitungsreihenfolge gewünscht, so ist dies durch eine entsprechende gegenseitige Verriegelung der DRUCK-Bausteine zu projektieren.

Kommunikation eines DRUCK-Bausteins und eines EMAS-Bausteins mit derselben seriellen Schnittstelle

Ein DRUCK-Baustein und ein EMAS-Baustein können *dieselbe* serielle Schnittstelle benutzen, ohne daß besondere Vorkehrungen zu treffen sind.

Anzeigeformate für auszugebende Zahlenwerte

Die SPS ist in der Lage, die auszugebenden Zahlenwerte auf einem Bildschirm oder Drucker in vielfältiger Art und Weise darzustellen.

Das Anzeigeformat für einen Zahlenwert wird durch die Formatkennung festgelegt. Die Formatkennung wird direkt *vor* dem auszugebenden Merker projiziert und besteht aus drei Ziffern. Sie ist wie folgt definiert:

Formatkennung: Die Formatkennung gibt den Typ des Operanden und das Anzeigeformat seines Wertes an. Die Formatkennung ist eine 3-stellige Zahl. Die 1. Ziffer von links gibt den Operandentyp an. Es gibt 3 Operandentypen:

Binär: 1
Wort: 2
Doppelwort: 3

Die Ziffern 2 und 3 definieren das Anzeigeformat.

Beispiele: ● Formatkennung 103

– Ziffer 1: 1: Binär-Operand
– Ziffer 2 u. 3: 03: Anzeigeformat 03 (siehe Tabelle)

● Formatkennung 204

– Ziffer 1: 2: Wort-Operand
– Ziffer 2 u. 3: 04: Anzeigeformat 04 (siehe Tabelle)

● Formatkennung 341

– Ziffer 1: 3: Doppel-Wort-Operand
– Ziffer 2 u. 3: 41: Anzeigeformat 41 (siehe Tabelle)

AUSGABE VON ASCII-ZEICHEN UND HEXWERTEN ÜBER EINE SERIELLE SCHNITTSTELLE

DRUCK

Anzeigeformate

Alle möglichen Anzeigeformate sind in der folgenden Tabelle aufgelistet. Mit Ausnahme der Sonderkennungen 98 und 99 sind alle Kennungen auf die Datentypen Binär, Wort und Doppelwort anwendbar.

Anzeigeformatkenn.	Anzeigeformat	Zahlenbeispiel	Ausgabe ASCII
--------------------	---------------	----------------	---------------

Führende Nullen durch Blanks ersetzt (Blanks werden in diesem Beispiel durch _ dargestellt)

17	x	00234	___ 4
18	xx		___ 34
19	xxx		___ 234
20	xxxx		___ 234
21	xxxxx		___ 234

63	x	0087654321	_____ 1
64	xx		_____ 21
65	xxx		_____ 321
66	xxxx		_____ 4321
67	xxxxx		_____ 54321
68	xxxxxx		_____ 654321
69	xxxxxxx		_____ 7654321
70	xxxxxxxx		_____ 87654321
71	xxxxxxxxx		_____ 87654321
72	xxxxxxxxxx		_____ 87654321

22	xxxx,x	00347	___ 34,7
23	xxx,xx		___ 3,47
24	xx,xxx		___ ,347
25	x,xxxx		___ ,0347
26	,xxxxx		___ ,00347
27	+/- xxxxx		+___ 347
28	+/- xxxxx,x		+___ 34,7

73	xxxxxxxxxx,x	0012345678	___ 1234567,8
74	xxxxxxxxxx,xx		___ 123456,78
75	xxxxxxxx,xxx		___ 12345,678
76	xxxxxx,xxxx		___ 1234,5678
77	xxxxx,xxxxx		___ 123,45678
78	xxxx,xxxxxx		___ 12,345678
79	xxx,xxxxxxx		___ 1,2345678
80	xx,xxxxxxxx		___ ,12345678
81	x,xxxxxxxxx		___ ,012345678
82	,xxxxxxxxxx		___ ,0012345678
83	+/- xxxxxxxxxxx		+ ___ 12345678
84	+/- xxxxxxxxxxx,x		+ ___ 1234567,8

Anzeigeformatkenn.	Anzeigeformat	Zahlenbeispiel	Ausgabe ASCII
--------------------	---------------	----------------	---------------

Zusätzliche Formate mit Unterdrückung führender Nullen

29	xxxxx	00839	839
30	xxxx,x		83,9
31	xxx,xx		8,39
32	xx,xxx		,839
33	+/- xxxxx		+839
34	+/- xxxxx,x		+83,9
35	+/- xxx,xx		+8,39
36	+/- xx,xxx		+,839

85	xxxxxxxxxx	0000000321	321
86	xxxxxxxxxx,x		32,1
87	xxxxxxxx,xx		3,21
88	xxxxxxx,xxx		,321
89	+/- xxxxxxxxxxx		+321
90	+/- xxxxxxxxxxx,x		+32,1
91	+/- xxxxxxxxxxx,xx		+3,21
92	+/- xxxxxxxx,xxx		+,321

Mögliche Anzeigeformate

Anzeigeformatkenn.	Anzeigeformat	Zahlenbeispiel	Ausgabe ASCII
--------------------	---------------	----------------	---------------

Format mit führenden Nullen

01	x	0012345678	8
02	xx		78
03	xxx		678
04	xxxx		5678
05	xxxxx		45678
37	xxxxxx		345678
38	xxxxxxx		2345678
39	xxxxxxx		12345678
40	xxxxxxxx		012345678
41	xxxxxxxxx		0012345678

06	xxxx,x	02215	0221,5
07	xxx,xx		022,15
08	xx,xxx		02,215
09	x,xxxx		0,2215
10	,xxxxx		,02215

42	xxxxxxxx,x	0011223344	001122334,4
43	xxxxxxxx,xx		00112233,44
44	xxxxxxx,xxx		0011223,344
45	xxxxxx,xxxx		001122,3344
46	xxxxx,xxxxx		00112,23344
47	xxxx,xxxxxx		0011,223344
48	xxx,xxxxxxx		001,1223344
49	xx,xxxxxxxx		00,11223344
50	x,xxxxxxxxx		0,011223344
51	,xxxxxxxxxx		,0011223344

11	+/- xxxxx	00331	+00331
12	+/- xxxxx,x		+0033,1
13	+/- xxx,xx		+003,31
14	+/- xx,xxx		+00,331
15	+/- x,xxxx		+0,0331
16	+/- ,xxxxx		+,00331

52	+/- xxxxxxxxxxx	0055667788	+0055667788
53	+/- xxxxxxxxxxx,x		+005566778,8
54	+/- xxxxxxxxxxx,xx		+00556677,88
55	+/- xxxxxxxx,xxx		+0055667,788
56	+/- xxxxxx,xxxx		+005566,7788
57	+/- xxxxx,xxxxx		+00556,67788
58	+/- xxxx,xxxxxx		+0055,667788
59	+/- xxx,xxxxxxx		+005,5667788
60	+/- xx,xxxxxxxx		+00,55667788
61	+/- x,xxxxxxxxx		+0,055667788
62	+/- ,xxxxxxxxxx		+,0055667788

Sonderformat: Ausgabe des HEX-Werts eines Wort-Operanden

Verfügbare Formate in 07 KR 31:

Der Wert eines Wort-Operanden wird direkt als Hexadezimalwert ausgegeben. Es erfolgt also vor der Ausgabe keine Wandlung des Werts in ASCII-Darstellung.

Bit: 101
Worte: 205...211
211...216
217...221
233...236
222...226
299

98 Es wird nur das LOW BYTE (8 Bit) des Wort-Operanden ausgegeben

99 Es wird zuerst das LOW BYTE und anschließend das HIGH BYTE des Wort-Operanden ausgegeben

Doppelworte: 341...351
352...362

Achtung: Dieses Sonderformat ist nur zulässig beim Datentyp "WORT".
Zulässig: 298 und 299
Unzulässig: 198, 199, 398 und 399

Beispiel: Eingabe eines Textes in das Anwenderprogramm und Ausgabe über die Schnittstelle

Es soll folgender Text ausgegeben werden:

Text-NR.: 3
Teiltext 1: ENTFERNUNG
Wert des Merkers: MW0,0
Ausgabeformat: 3-stellig mit führenden Nullen
Teiltext 2: KM

Anwender-Eingabe	Bedeutung	Ausgabe über die serielle Schnittstelle
#3	Text-Nr. 3, direkte Konstante	—
#"	Kennung für den Anfang der Texteingabe	—
E	Buchstabe in ASCII	45 _H
N	Buchstabe in ASCII	4E _H
.	.	.
.	.	.
.	.	.
N	.	4E _H
G	.	47 _H
"# ^)	Kennung für das Ende der Texteingabe	—
#203	Typ, Formatkennung als direkte Konstante Typ: Wort, Format: 03	—
MW0,0	Inhalt des MW0,0	Formatierter Dezimal-Wert des Merkers MW0,0 in ASCII-Darstellung
#"	Kennung für den Anfang der Texteingabe	—
K	Buchstabe in ASCII	4B _H
M	" "	4D _H
"# ^)	Kennung für das Ende der Texteingabe	—
	Text-Abschlusszeichen (z.B.<CR>), falls dies im Baustein SINIT projiziert und zur Ausgabe freigegeben ist	0D _H

*) Die Endekennung "# entfällt, falls die Eingabe in das ABB Procontic Programmiersystem erfolgt. Die Endekennung "# ist nur erforderlich, wenn die Eingabe mit einem Terminal direkt in die SPS erfolgt.

Beispiele für Text-Abschlußzeichen, die beim Baustein SINIT anzugeben sind

<ETX> = 03_H = 3
<EOT> = 04_H = 4
<CR> = 0D_H = 13
<LF> = 0A_H = 10
<SP> = 20_H = 32

. . .
. . .
. . .

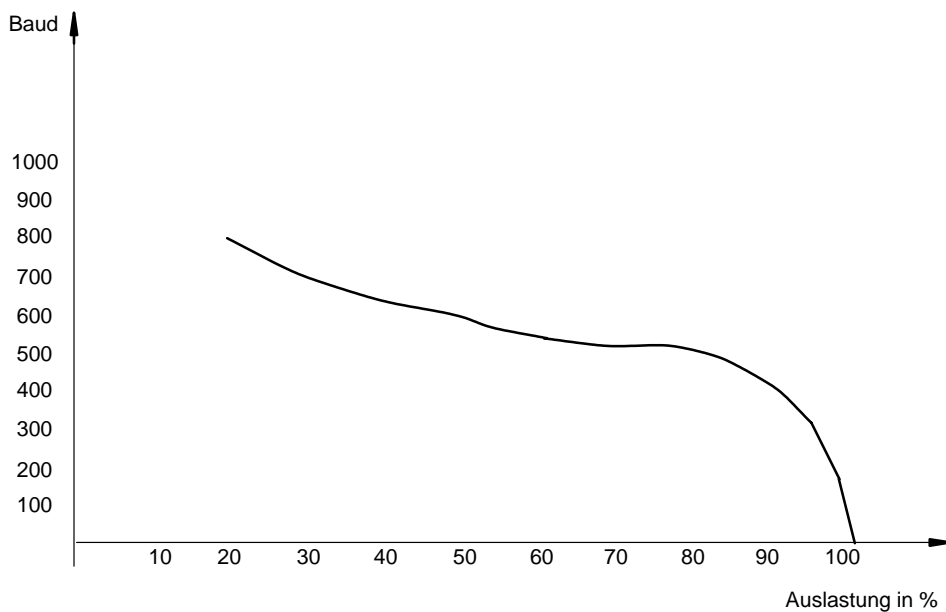
Effektive Datenübertragungsrate beim Baustein DRUCK

Für 07 KR 31 nicht verfügbar.

Die effektive Datenübertragungsrate (Zeichen, die tatsächlich an der seriellen Schnittstelle ausgegeben werden) ist im wesentlichen abhängig von der Auslastung der SPS und der eingestellten Baudrate.

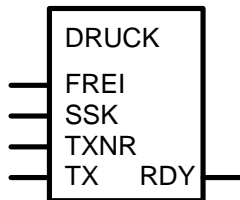
Es gilt grob folgender Zusammenhang:

Randbedingung: eingestellte Baudrate: 9600 Baud



Achtung: Die Auslastungsanzeige (Befehl AL) ist nur korrekt, wenn zum Zeitpunkt der Anzeige *keine* Kommunikation über die seriellen Schnittstellen stattfindet. Durch die Kommunikation wird die Auslastungsanzeige verfälscht und damit unbrauchbar. Sind in einem Anwenderprogramm die Bausteine DRUCK oder EMAS vorhanden, so sind diese bei der Ermittlung der Auslastung zu sperren.

VE-FUP-Definition

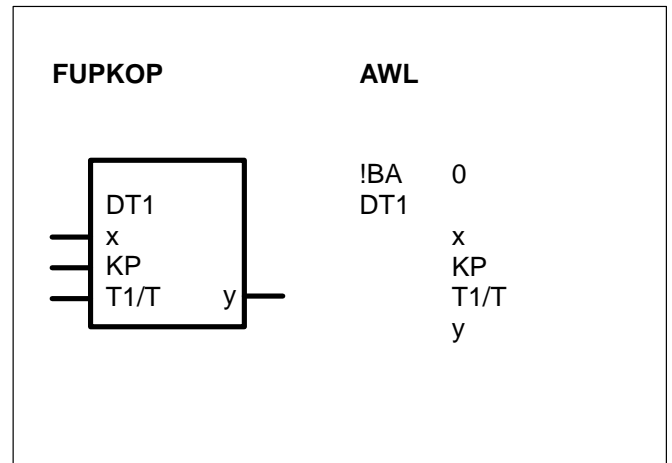


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
SSK	E	W	N	P	J	0	0
TXNR	E	W	N	P	J	0	0
TX	E	X	N	P	J	1	0
RDY	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DRUCK			
00002		PP 0	FREI	Eingang BINÄR (Baustein-Freigabe)
00003		PP 0	TXNR	Eingang WORT (Textnummer)
00004		PP 0	SSK	Eingang WORT (Schnittstellenkennung)
00005		PP 0	RDY	Ausgang BINÄR (Text ausgeben)
	[1		
00006		PP 1	TX	Eingang ALLE (Nr., Text, Format, 0)
]	1		

Die Regelgröße x wird mit dem Proportionalbeiwert KP multipliziert. Der Proportionalbeiwert wird in Prozent angegeben. Ändert sich die Regelgröße nicht mehr, so läuft die Stellgröße y in einer e-Funktion gegen Wert 0. Die Zeitkonstante T1 gibt die Zeit an, wann die Sprungantwort auf ca. 37 Prozent ihres Anfangswertes gefallen ist. Nach 5*T1 ist der Wert unter 1 Prozent gefallen.



Parameter

x	WORT	EW, AW, MW, KW	Regelgröße
KP	WORT	EW, AW, MW, KW	Proportionalbeiwert, Angabe in Prozent
T1/T	WORT	EW, AW, MW, KW	Zeitkonstante auf Zykluszeit normiert
y	WORT	AW, MW	Stellgröße

VE-Daten

Laufzeit:	
Grundlaufzeit:	200 µs
Zusätzliche Laufzeit:	nein
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	4 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

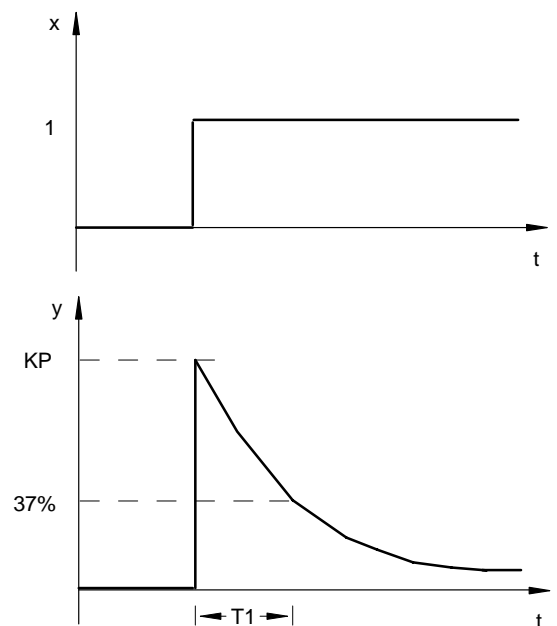
Die Regelgröße x wird mit dem Proportionalbeiwert KP multipliziert. Der Proportionalbeiwert wird in Prozent angegeben. Ändert sich die Regelgröße nicht mehr, so läuft die Stellgröße y in einer e-Funktion gegen Wert 0. Die Zeitkonstante T1 gibt die Zeit an, wann die Sprungantwort auf ca. 37 Prozent ihres Anfangswertes gefallen ist. Nach 5*T1 ist der Wert unter 1 Prozent gefallen.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar.

Übertragungsfunktion:

$$F(s) = \frac{KP \cdot s \cdot T1}{s \cdot T1 + 1}$$

Übergangsfunktion:



x WORT

Am Eingang x wird der Operand für die Regelgröße (Eingangswert für die DT1-Funktion) angegeben.

KP WORT

Der Proportionalbeiwert wird am Eingang KP angegeben. Die Angabe dieses Wertes erfolgt (seit Version V6; GJP5122900 R301) in Prozent und kann positiv oder negativ sein.

Beispiel:

1	=	1	Prozent
55	=	55	Prozent
100	=	100	Prozent
1000	=	1000	Prozent
-100	=	-100	Prozent

100 Prozent heißt, daß der Eingang x durch den Proportionalbeiwert *nicht* beinflusst wird.

T1/T WORT

Am Eingang T1/T wird die Zeitkonstante der DT1-Funktion angegeben. Die Zeitkonstante T1 ist dabei auf die SPS-Zykluszeit T zu normieren.

Für T1/TZ gilt:

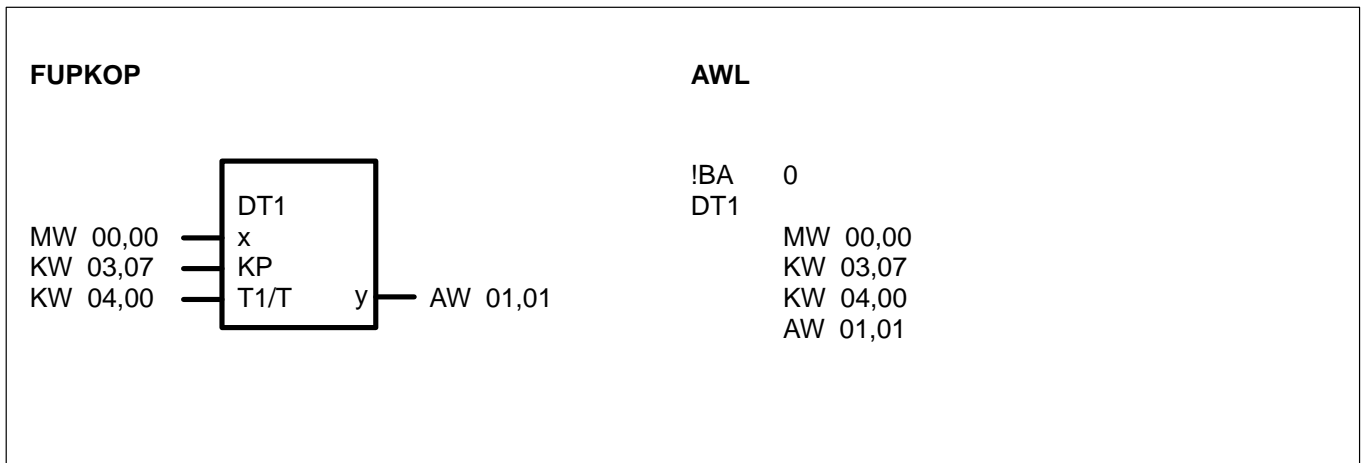
$$10 \leq T1/TZ \leq 32767 \rightarrow T1 > 10 * TZ$$

Wird für T1/T ein negativer Wert vorgegeben, so wird T1/T von der SPS automatisch auf den max. positiven Wert +32767 gesetzt.

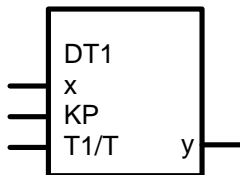
y WORT

Am Ausgang y wird die Stellgröße (Ausgangswert der DT1-Funktion) ausgegeben.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
x	E	W	N	P	J	0	0
KP	E	W	N	P	J	0	0
T1/T	E	W	N	P	J	0	0
y	A	W	N	P	J	0	0

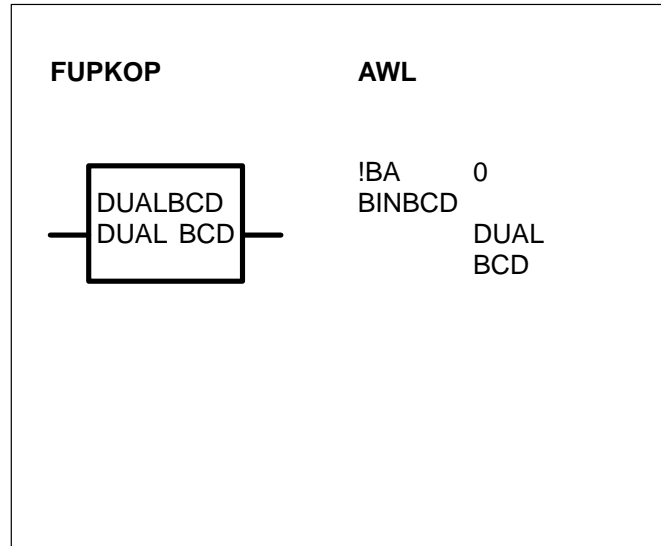
VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DT1			
00002		PP 0	x	Eingang WORT (Regelgröße)
00003		PP 0	KP	Eingang WORT (Verstärkungsf.)
00004		PP 0	T1/T	Eingang WORT (Zeitkonstante)
00005		PP 0	y	Ausgang WORT (Stellgröße)

Die Dualzahl am Eingang DUAL wird in eine BCD codierte Zahl gewandelt und dem Operanden am Ausgang BCD zugewiesen.

Die Dualzahl ist in 16 Bit dargestellt und muß im Bereich $0 \leq \text{Dual} \leq 270F_H$ (entspricht BCD 9999)

liegen. Liegt sie außerhalb dieses Bereichs, so wird die BCD-Zahl auf 9999 begrenzt. Die BCD-Zahl wird in einem 16-Bit-Wort abgelegt.



Parameter

DUAL	WORT	EW, MW, AW, KW	Dualzahl
BCD	WORT	AW,MW	BCD-codierte Zahl

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

Vom Wert abhängig: 28 µs
(Wert=0) ... 162 µs (Wert=9999)

548 µs

Zusätzliche Laufzeit:

—

—

Aktualisierung der Ausgänge:

ja

ja

Anzahl der Vergangenheitswerte:

keine

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Die Dualzahl am Eingang DUAL wird in eine BCD codierte Zahl gewandelt und dem Operanden am Ausgang BCD zugewiesen.

Die Dualzahl ist in 16 Bit dargestellt und muß im Bereich $0 \leq \text{Dual} \leq 270FH$ (entspricht BCD 9999)

liegen. Liegt sie außerhalb dieses Bereichs, so wird die BCD-Zahl auf 9999 begrenzt. Die BCD-Zahl wird in einem 16-Bit-Wort abgelegt.

Der Eingang und der Ausgang ist weder doppelbar noch negierbar.

Definition:

Die Wertigkeit der Ziffern einer Hexadezimalzahl und einer BCD-codierten Zahl sind wie folgt definiert:

HEXDEZ-ZAHL					BCD-ZAHL					
15	11	7	3	0	BIT	15	11	7	3	0
Z4	Z3	Z2	Z1		Z4	Z3	Z2	Z1		

Zahlenwert:

$$\begin{aligned} Z1 &* 1 \\ Z2 &* 16 \\ Z3 &* 256 \\ Z4 &* 4096 \end{aligned}$$

$$0 \leq Zi \leq F$$

Zahlenwert:

$$\begin{aligned} Z1 &* 1 \\ Z2 &* 10 \\ Z3 &* 100 \\ Z4 &* 1000 \end{aligned}$$

$$0 \leq Zi \leq 9$$

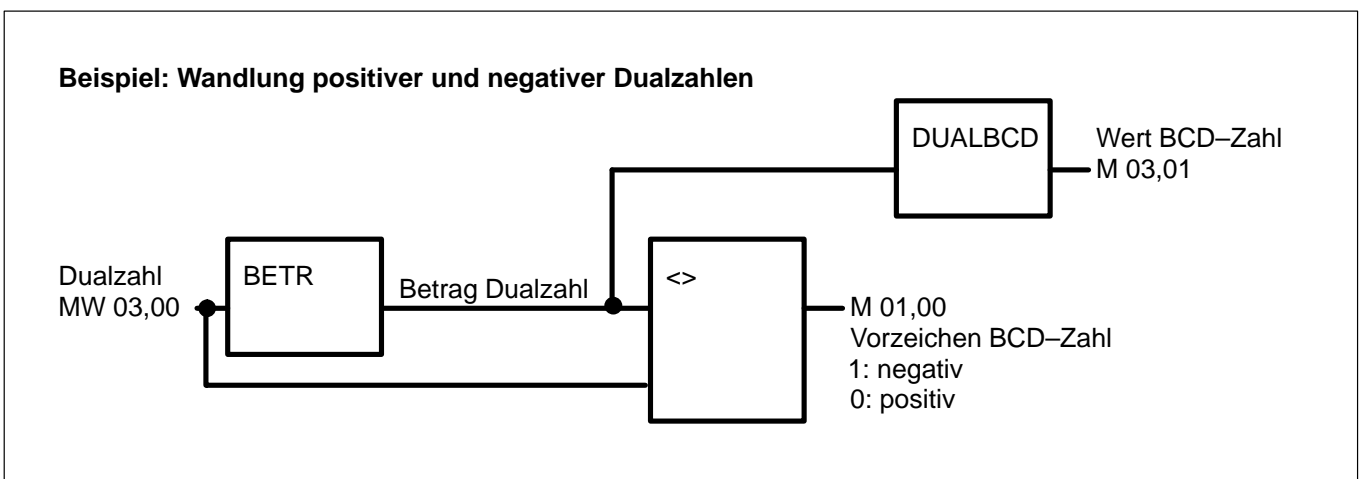
Beispiel:

HEXDEZ-ZAHL					BCD-ZAHL					
15	11	7	3	0	BIT	15	11	7	3	0
0	4	D	2		1	2	3	4		

$Z1 = 2 * 1 = 2$	$Z1 = 4 * 1 = 4$
$Z2 = 13 * 16 = 208$	$Z2 = 3 * 10 = 30$
$Z3 = 4 * 256 = 1024$	$Z3 = 2 * 100 = 200$
$Z4 = 0 * 4096 = 0$	$Z4 = 1 * 1000 = 1000$
+ 1234	+ 1234

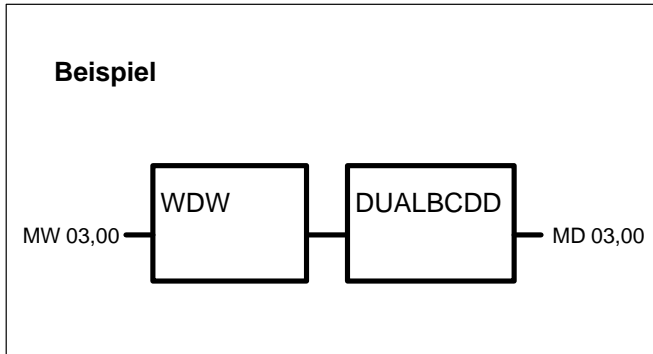
Wandlung einer negativen Dual-Zahl in eine BCD-Zahl

Eine negative Dual-Zahl mit einem Betrag kleiner als 270FH kann in eine BCD-Zahl gewandelt werden, wobei der Wert und das Vorzeichen der BCD-Zahl in jeweils einem Merker abgelegt werden.



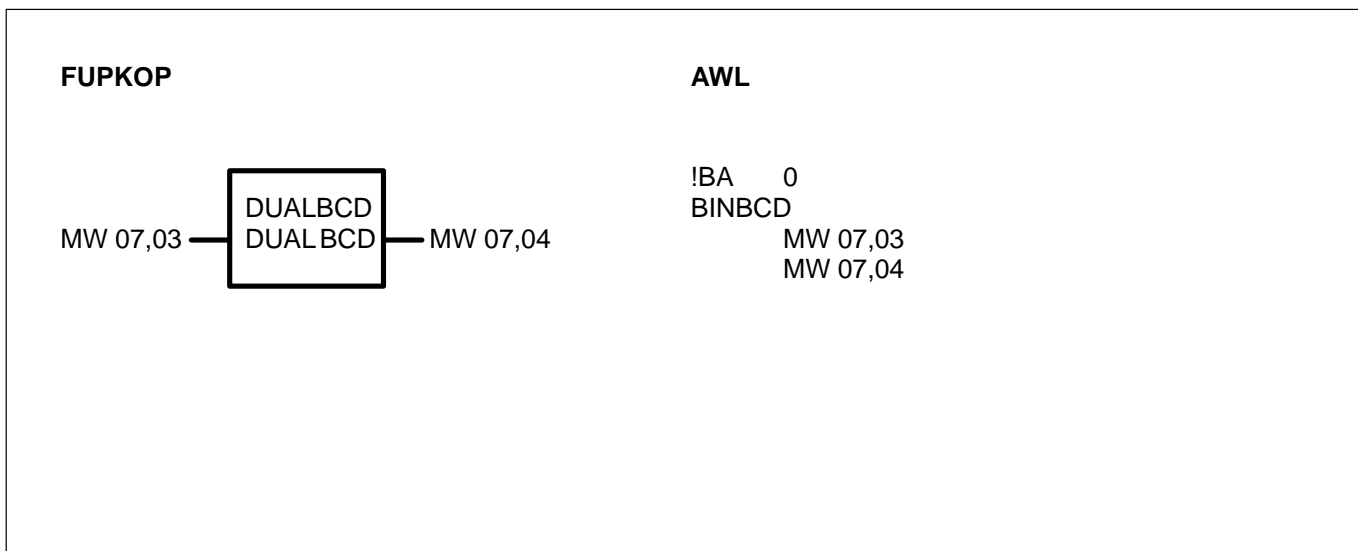
Wandlung einer Dual-Zahl mit Betrag größer als 270F_H

Dual-Zahlen mit einem Betrag größer als 270F_H werden zunächst in ein Doppelwort gewandelt (Funktionsbaustein WDW). Anschließend erfolgt die Wandlung von DUAL nach BCD mit dem Funktionsbaustein DUALBCDD.

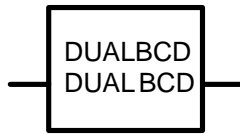


Falls ein Vorzeichen berücksichtigt werden soll, gilt das gleiche Verfahren wie oben.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
DUAL	E	W	N	P	J	0	0
BCD	A	W	N	P	J	0	0

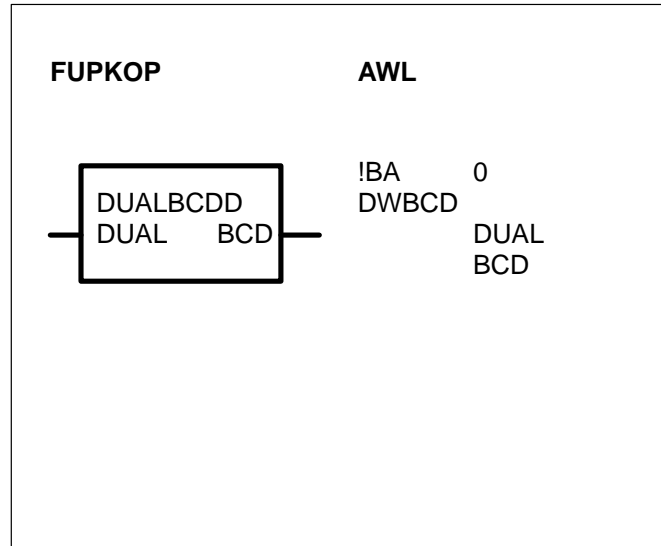
VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	BINBCD			
00002		PP 0	DUAL	Eingang WORT (Dualzahl)
00003		PP 0	BCD	Ausgang WORT (BCD-Zahl)

Die Dualzahl am Eingang DUAL wird in eine BCD-codierte Zahl gewandelt und dem Operanden am Ausgang BCD zugewiesen.

Die Dualzahl ist in 32 Bit dargestellt und muß im Bereich $0 \leq \text{Dual} \leq 5F5E0FF_H$ (entspricht BCD 99999999) liegen.

Liegt sie außerhalb dieses Bereichs, so wird die BCD-Zahl auf 99999999 begrenzt. Die BCD-Zahl wird in einem 32 Bit Wort abgelegt.



Parameter

DUAL	DOPPELWORT MD, KD	Dualzahl
BCD	DOPPELWORT MD	BCD-codierte Zahl

VE-Daten

Laufzeit:	
Grundlaufzeit:	36 – 288 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Die Dualzahl am Eingang DUAL wird in eine BCD-codierte Zahl gewandelt und dem Operanden am Ausgang BCD zugewiesen.

Die Dualzahl ist in 32 Bit dargestellt und muß im Bereich $0 \leq \text{Dual} \leq 5F5E0FF_H$ (entspricht BCD 99999999)

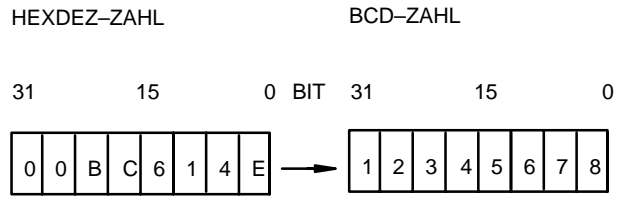
liegen. Liegt sie außerhalb dieses Bereichs, so wird die BCD-Zahl auf 99999999 begrenzt. Die BCD-Zahl wird in einem 32 Bit Wort abgelegt.

Der Ein- und Ausgang ist weder doppelbar noch negierbar.

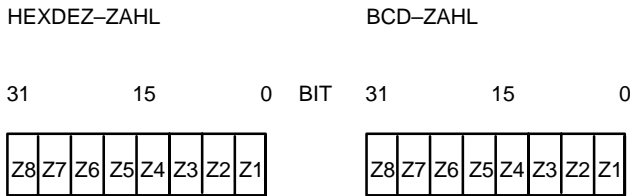
Definition

Die Wertigkeit der Ziffern einer Hexadezimal-Zahl und einer BCD-codierten Zahl sind wie folgt definiert:

Beispiel:



Z1 = 14 * 1	= 14	Z1 = 8 * 1	= 8
Z2 = 4 * 16	= 64	Z2 = 7 * 10	= 70
Z3 = 1 * 256	= 256	Z3 = 6 * 100	= 600
Z4 = 6 * 4096	= 24576	Z4 = 5 * 1000	= 5000
Z5 = 12 * 65536	= 786432	Z5 = 4 * 10000	= 40000
Z6 = 11 * 1048576	= 11534336	Z6 = 3 * 100000	= 300000
Z7 = 0 * 16777216	= 0	Z7 = 2 * 1000000	= 2000000
Z8 = 0 * 268435456	= 0	Z8 = 1 * 10000000	= 10000000
+ 12345678		+ 12345678	



Zahlenwert:

Z1 * 1
Z2 * 16
Z3 * 256
Z4 * 4096
Z5 * 65536
Z6 * 1048576
Z7 * 16777216
Z8 * 268435456

$0 \leq Z_i \leq F$

Zahlenwert:

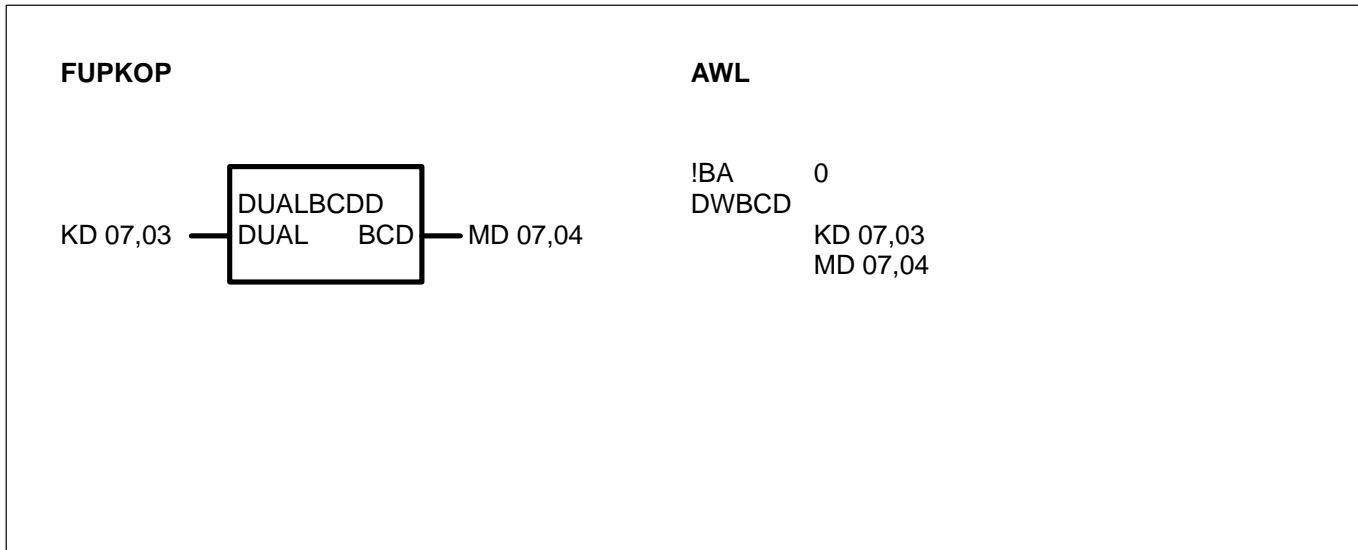
Z1 * 1
Z2 * 10
Z3 * 100
Z4 * 1000
Z5 * 10000
Z6 * 100000
Z7 * 1000000
Z8 * 10000000

$0 \leq Z_i \leq 9$

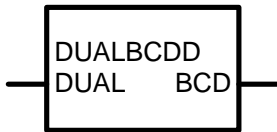
Wandlung einer negativen Dual-Zahl in eine BCD-Zahl

siehe Funktionsbaustein DUALBCD

Beispiel



VE-FUP-Definition

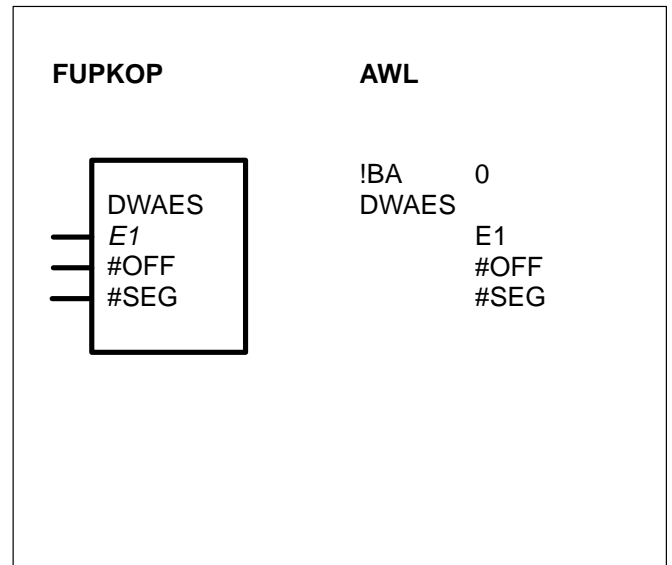


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
DUAL	E	D	N	P	J	0	0
BCD	A	D	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DWBCD			
00002		PP 0	DUAL	Eingang DOPPELWORT (Dualzahl)
00003		PP 0	BCD	Ausgang DOPPELWORT (BCD-Zahl)

Ändert sich der Wert des Operanden am Eingang E1 verglichen mit dem Wert bei der vorhergehenden Bearbeitung des Bausteins, so wird der Wert des Operanden am Eingang E1 auf die angegebene physikalische Adresse geschrieben.



Parameter

E1	DOPPELWORT	MD, KD
#OFF	DIREKTE	#, #H
	KONSTANTE	
#SEG	DIREKTE	#, #H
	KONSTANTE	

Eingang für den zu lesenden Operanden
 Offset–Adresse des Speicherplatzes, auf den der Wert von E1 bei Änderung zu schreiben ist.
 Segment–Adresse des Speicherplatzes, auf den der Wert von E1 bei Änderung zu schreiben ist.

VE–DATEN

Laufzeit:		
Grundlaufzeit:		27 µs
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		2 Worte
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Ändert sich der Wert des Operanden am Eingang E1 verglichen mit dem Wert bei der vorhergehenden Bearbeitung des Bausteins, so wird der Wert des Operanden am Eingang E1 auf die angegebene physikalische Adresse geschrieben.

#OFF DIREKTE KONSTANTE (,#H)
 Am Eingang #OFF wird der Offset–Anteil der zu schreibenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

#SEG DIREKTE KONSTANTE (,#H)
 Am Eingang #SEG wird der Segment–Anteil der zu schreibenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

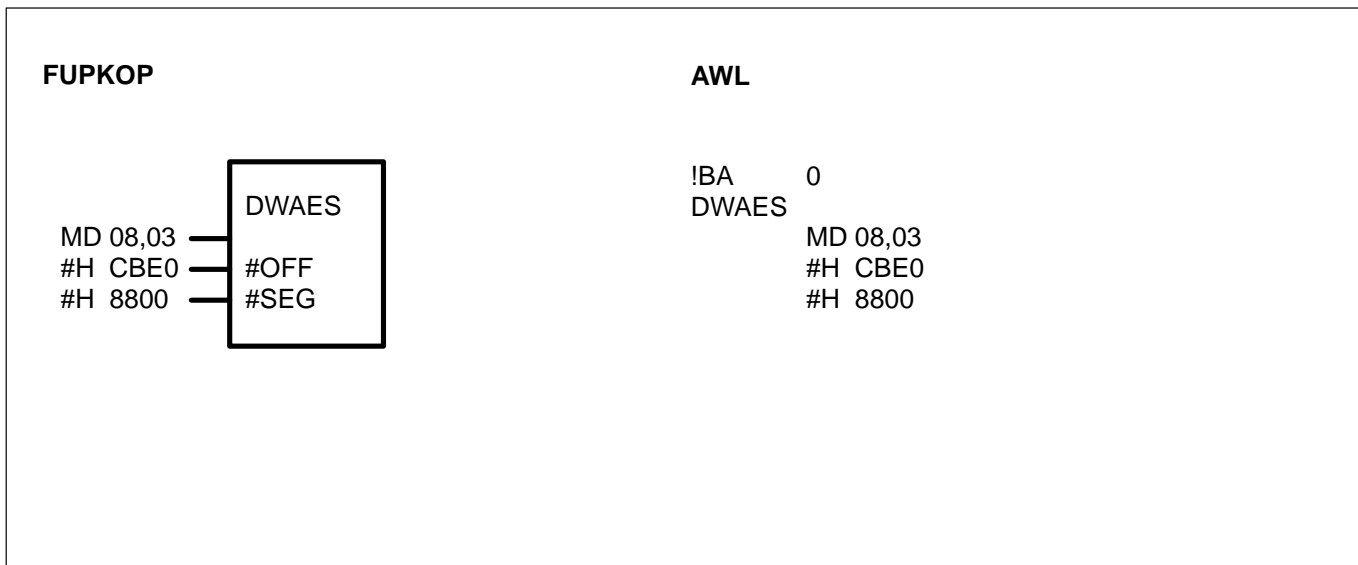
Die physikalische Adresse besteht aus Segment und Offset.

Die Eingänge sind weder doppelbar noch negierbar.

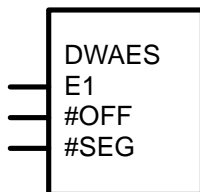
E1 DOPPELWORT

Bei Änderung des Operanden am Eingang E1 wird dessen Wert auf die Adresse, die an den Eingängen #OFF und #SEG angegeben ist, geschrieben.

Beispiel



VE-FUP-Definition

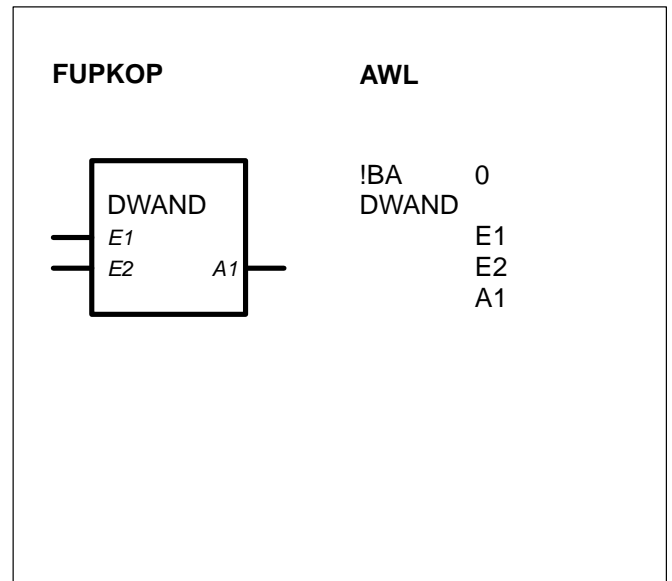


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
#OFF	K	W	N	P	J	0	0
#SEG	K	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DWAES			
00002		PP 0	E1	Eingang DOPPELWORT (Wert)
00003		PP 0	#OFF	# KONSTANTE (Offset-Adresse)
00004		PP 0	#SEG	# KONSTANTE (Segment-Adresse)

Der Funktionsbaustein bildet bitweise die UND-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	DOPPELWORT MD, KD	Operand 1
E2	DOPPELWORT MD, KD	Operand 2
A1	DOPPELWORT MD	Ergebnis der UND-Verknüpfung

VE-Daten

Laufzeit:	
Grundlaufzeit:	49 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

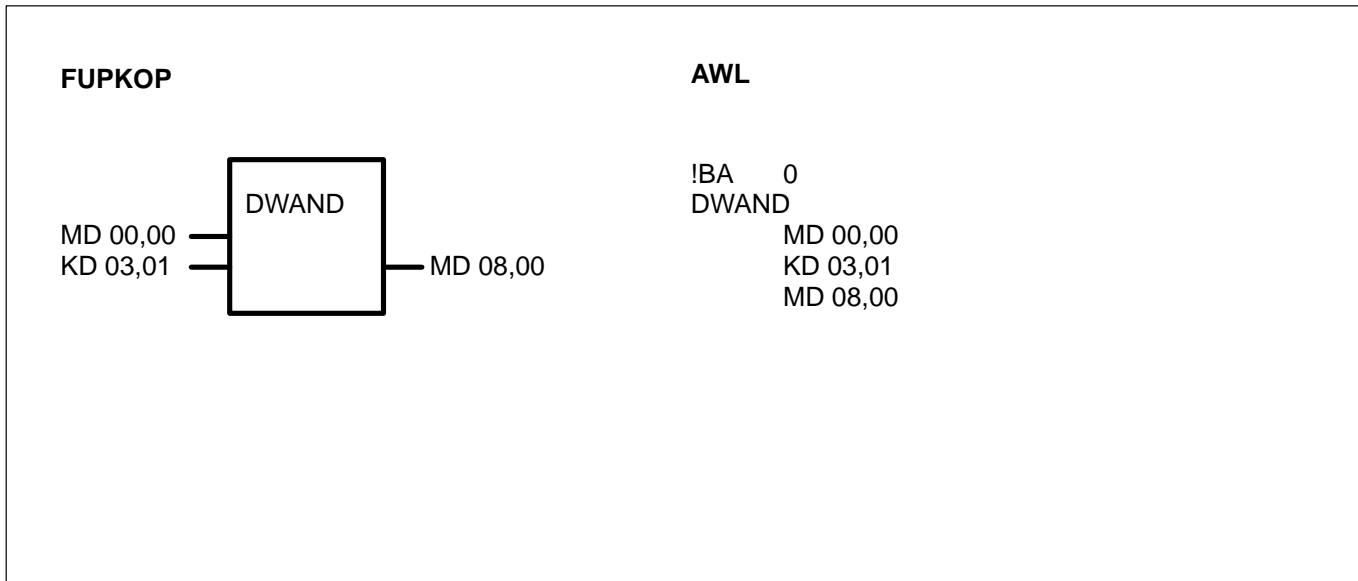
Der Funktionsbaustein bildet bitweise die UND-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar/negierbar.

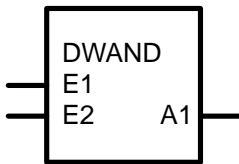
Beispiel

E1	1.0.0.0	0.0.1.1	0.0.1.0	0.1.1.0	1.0.1.0	1.1.0.0	0.0.1.1	0.1.0.1
E2	1.0.0.1	0.1.1.0	0.0.1.0	1.1.1.1	1.1.1.1	0.0.0.0	0.1.1.0	1.1.0.0
A1	1.0.0.0	0.0.1.0	0.0.1.0	0.1.1.0	1.0.1.0	0.0.0.0	0.0.1.0	0.1.0.0

Beispiel



VE-FUP-Definition

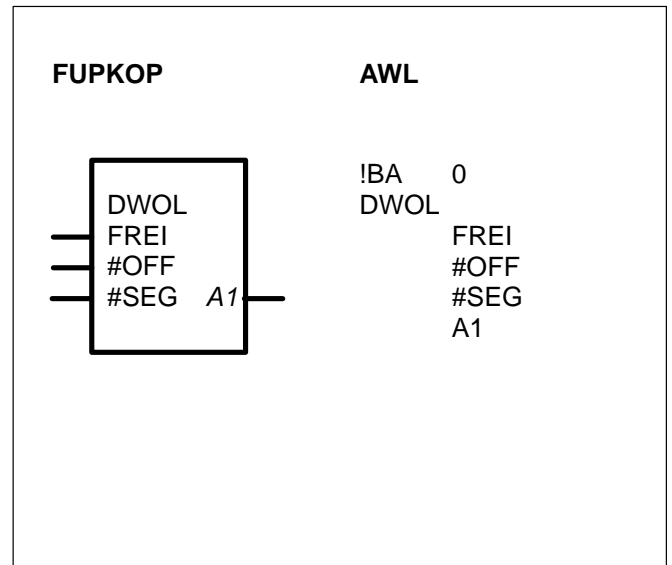


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
E2	E	D	N	P	N	0	0
A1	A	D	N	P	N	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DWAND			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	E2	Eingang DOPPELWORT
00004		PP 0	A1	Ausgang DOPPELWORT

1–Signal am Eingang FREI bewirkt, daß der Wert der angegebenen physikalischen Adresse gelesen und dem Operanden am Ausgang A1 zugewiesen wird.



Parameter

FREI	BINÄR	E, A, M, S, K	Freigabe des Bausteins
#OFF	DIREKTE KONSTANTE	#, #H	Offset–Adresse des Speicherplatzes, dessen Doppelwort–Wert zu lesen ist.
#SEG	DIREKTE KONSTANTE	#, #H	Segment–Adresse des Speicherplatzes, dessen Doppelwort–Wert zu lesen ist.
A1	DOPPELWORT	MD	Ausgang, dem der gelesene Wert zugewiesen wird.

VE–DATEN

Laufzeit:		
Grundlaufzeit:		26,5 µs; 19,0 µs, falls keine Freigabe.
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

1–Signal am Eingang FREI bewirkt, daß der Wert der angegebenen physikalischen Adresse gelesen und dem Operanden am Ausgang A1 zugewiesen wird.

Bei 0–Signal am Eingang FREI erfolgt kein Lesen.

Die physikalische Adresse besteht aus Segment und Offset.

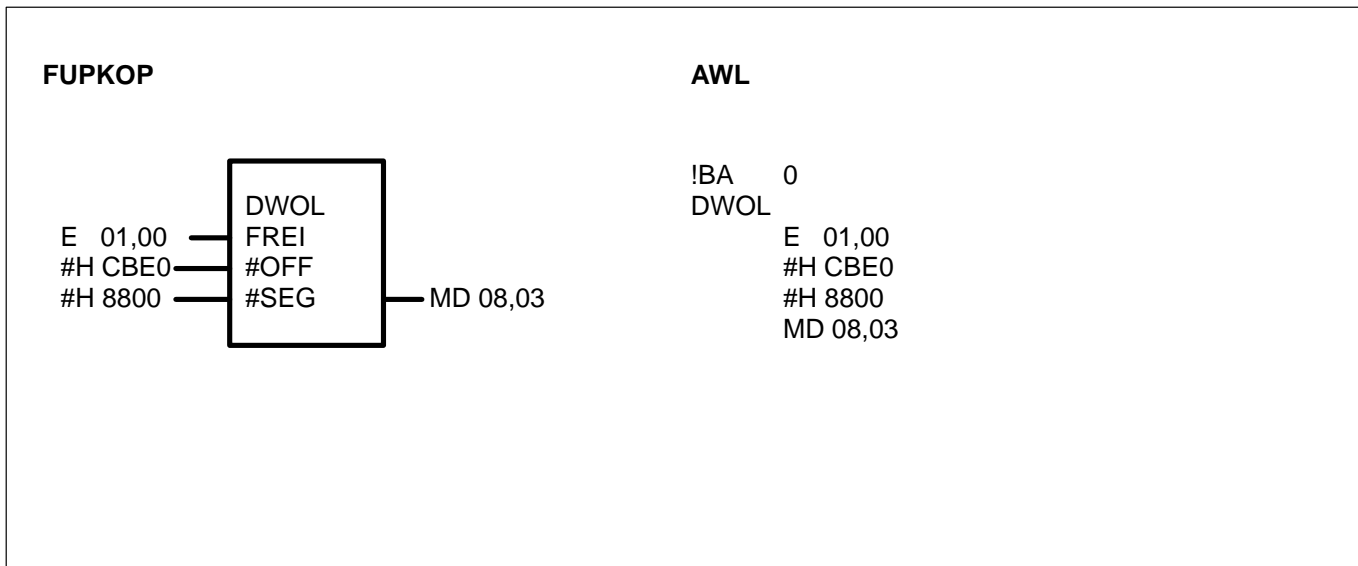
FREI BINÄR
 Mit dem Operand am Eingang FREI wird die Bearbeitung des Bausteins freigegeben oder gesperrt. Es gilt:
 FREI = 0 → Bearbeitung gesperrt
 FREI = 1 → Bearbeitung freigegeben

#OFF DIREKTE KONSTANTE (#, #H)
 Am Eingang #OFF wird der Offset–Anteil der zu lesenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

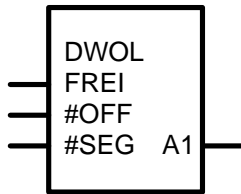
#SEG DIREKTE KONSTANTE (#, #H)
 Am Eingang #SEG wird der Segment–Anteil der zu lesenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

A1 WORT
 Dem Operanden am Ausgang A1 wird der gelesene Wert zugewiesen.

Beispiel



VE-FUP-Definition

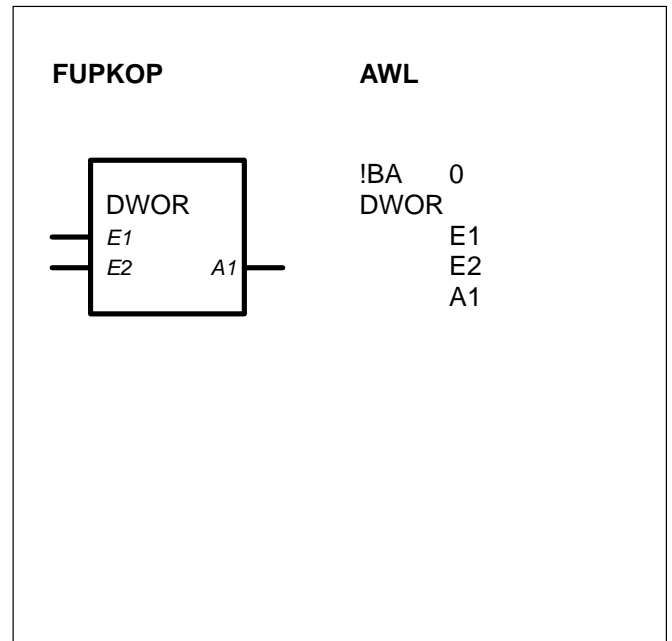


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
#OFF	K	W	N	P	J	0	0
#SEG	K	W	N	P	J	0	0
A1	A	D	N	P	N	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DWOL			
00002		PP 0	FREI	Eingang BINÄR (Baustein-Freigabe)
00003		PP 0	#OFF	# KONSTANTE (Offset-Adresse)
00004		PP 0	#SEG	# KONSTANTE (Segment-Adresse)
00005		PP 0	A1	Ausgang DOPPELWORT (Wert)

Der Funktionsbaustein bildet die bitweise ODER-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	DOPPELWORT MD, KD	Operand 1
E2	DOPPELWORT MD, KD	Operand 2
A1	DOPPELWORT MD	Ergebnis der ODER-Verknüpfung

VE-Daten

Laufzeit:	
Grundlaufzeit:	35 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

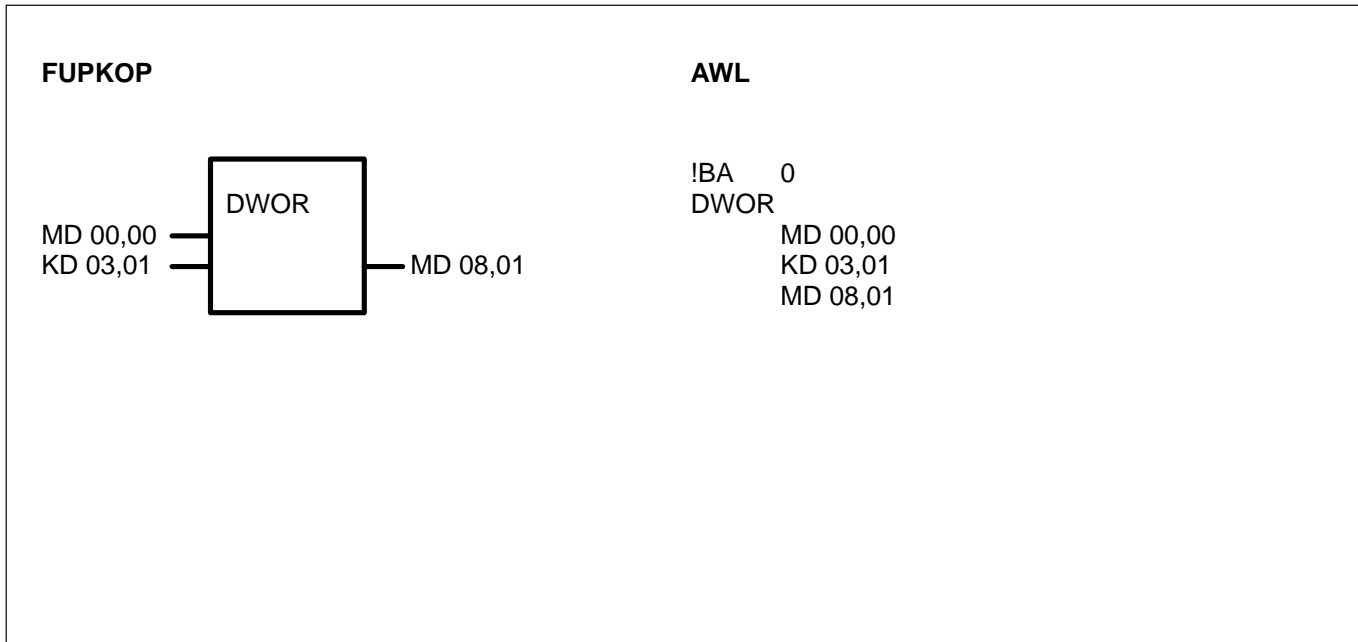
Der Funktionsbaustein bildet die bitweise ODER-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar/negierbar.

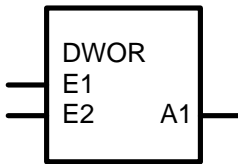
Beispiel

E1	1.0.0.0	0.0.1.1	0.0.1.0	0.1.1.0	1.0.1.0	1.1.0.0	0.0.1.1	0.1.0.1
E2	1.0.0.1	0.1.1.0	0.0.1.0	1.1.1.1	1.1.1.1	0.0.0.0	0.1.1.0	1.1.0.0
A1	1.0.0.1	0.1.1.1	0.0.1.0	1.1.1.1	1.1.1.1	1.1.0.0	0.1.1.1	1.1.0.1

Beispiel



VE-FUP-Definition

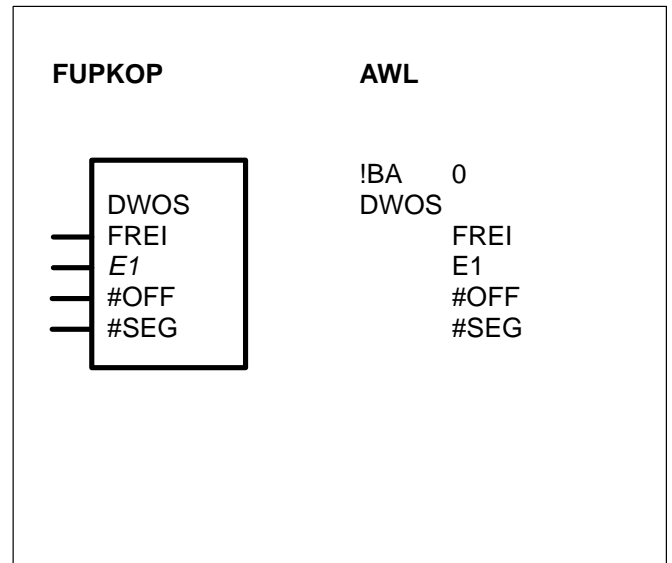


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
E2	E	D	N	P	N	0	0
A1	A	D	N	P	N	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DWOR			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	E2	Eingang DOPPELWORT
00004		PP 0	A1	Ausgang DOPPELWORT

1–Signal am Eingang FREI bewirkt, daß der Wert des Operanden am Eingang E1 gelesen und dann auf die angegebene physikalische Adresse geschrieben wird.



Parameter

FREI	BINÄR	E, M, A, S, K	Freigabe des Bausteins
E1	DOPPELWORT	MD, KD	Eingang für den zu schreibenden Operanden
#OFF	DIREKTE KONSTANTE	#, #H	Offset–Adresse des Speicherplatzes, auf den der Wert von E1 zu schreiben ist.
#SEG	DIREKTE KONSTANTE	#, #H	Segment–Adresse des Speicherplatzes, auf den der Wert von E1 zu schreiben ist.

VE–DATEN

Laufzeit:		
Grundlaufzeit:		38 µs, bei Freigabe; 19 µs, falls keine Freigabe.
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

1–Signal am Eingang FREI bewirkt, daß der Wert des Operanden am Eingang E1 gelesen und dann auf die angegebene physikalische Adresse geschrieben wird.

Bei 0–Signal am Eingang FREI erfolgt kein Schreiben.

Die physikalische Adresse besteht aus Segment und Offset.

Die Eingänge sind weder doppelbar noch negierbar.

FREI BINÄR

Mit dem Operanden am Eingang FREI wird die Bearbeitung des Bausteins freigegeben oder gesperrt. Es gilt:
 FREI = 0 → Bearbeitung gesperrt
 FREI = 1 → Bearbeitung freigegeben

E1 DOPPELWORT

Der Operand am Eingang E1 wird gelesen und sein Wert auf die durch die Eingänge #OFF und #SEG definierte Adresse geschrieben.

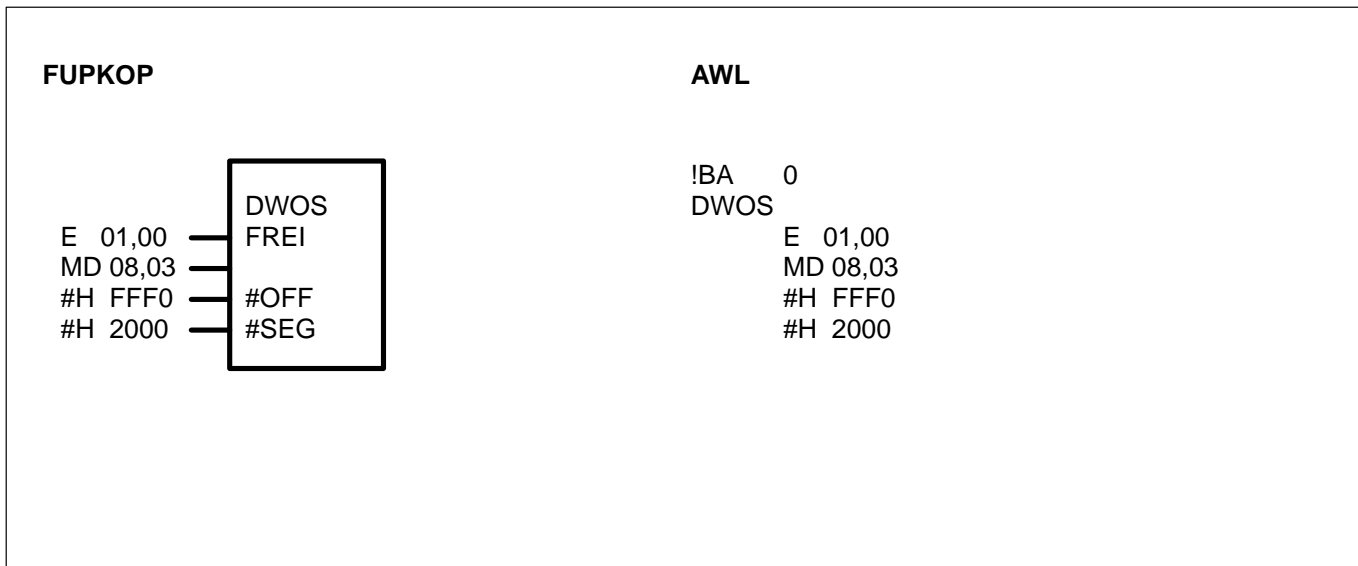
#OFF DIREKTE KONSTANTE (#, #H)

Am Eingang #OFF wird der Offset–Anteil der zu schreibenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

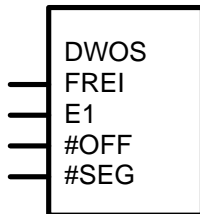
#SEG DIREKTE KONSTANTE (#, #H)

Am Eingang #SEG wird der Segment–Anteil der zu schreibenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

Beispiel



VE-FUP-Definition

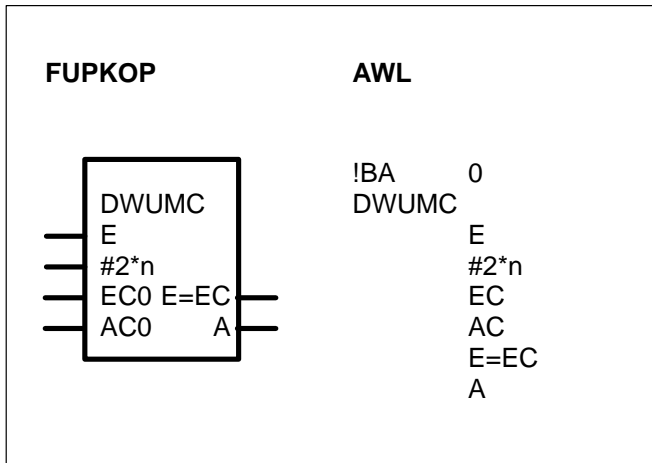


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
E1	E	D	N	P	N	0	0
#OFF	K	W	N	P	J	0	0
#SEG	K	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DWOS			
00002		PP 0	FREI	Eingang BINÄR (Baustein-Freigabe)
00003		PP 0	E1	Eingang DOPPELWORT (Wert)
00004		PP 0	#OFF	# KONSTANTE (Offset-Adresse)
00005		PP 0	#SEG	# KONSTANTE (Segment-Adresse)

Der Funktionsbaustein vergleicht den Wert des Operanden am Eingang E mit den Referenzwerten der Operanden an den Eingängen EC0 ... ECn-1. Stimmt der Eingang E mit mindestens einem der Referenzwerte EC überein, so wird der Ausgang E=EC auf 1 gesetzt. Der Ausgang A erhält den Wert des Ausgangscodes AC, der dem gefundenen Referenzwert EC zugeordnet ist.



Parameter

E	DOPPELWORT	MD, KD	Eingang
#2*n	DIREKTE	#, #H, KW	Anzahl n der Referenzwerte (multipliziert mit 2)
	KONSTANTE		
EC	DOPPELWORT	MD, KD	Referenzwert; Eingang ist doppelbar
AC	DOPPELWORT	MD, KD	Ausgangscodes; Eingang ist doppelbar
E=EC	BINÄR	A, M	Koinzidenzanzeige
A	DOPPELWORT	MD	Ausgabe des Werts des Ausgangscodes

VE-Daten

Laufzeit:	
Grundlaufzeit:	53 µs
Zusätzliche Laufzeit:	10 µs, pro eingegebenem Vergleichswert
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein vergleicht den Wert des Operanden am Eingang E mit den Referenzwerten der Operanden an den Eingängen EC0 ... ECn-1. Stimmt der Eingang E mit mindestens einem der Referenzwerte EC überein, so wird der Ausgang E=EC auf 1 gesetzt. Der Ausgang A erhält den Wert des Ausgangscodes AC, der dem gefundenen Referenzwert EC zugeordnet ist.

Jedem Referenzwert an den Eingängen ECi ist ein Operand für den Ausgangscode ACi zugeordnet. Die Zuordnung von EC zu AC ist am Index i zu erkennen. Der Index beginnt mit 0 und wird bei Dopplung automatisch erzeugt.

Die Anzahl der Eingänge EC und AC ist am Eingang #2*n als direkte Konstante anzugeben.

Die Eingänge und Ausgänge sind nicht negierbar/invertierbar.

E DOPPELWORT

Am Eingang E wird der Operand angegeben, dessen Wert mit den Werten der n Referenzwerte (EC0 ... ECn-1) verglichen werden soll.

#2*n DIREKTE KONSTANTE (#, #H)

Am Eingang #2*n wird die Gesamtanzahl (2*n) der Referenzwerte (EC0 ... ECn-1) und Ausgangscodes (AC0 ... ACn-1) angegeben. Die Angabe erfolgt als indirekte Konstante.

EC0 ... ECn-1 DOPPELWORT

Der Eingang EC0 ist entsprechend der benötigten Anzahl von Referenzwerten zu doppelnd. An den Eingängen EC0 ... ECn-1 werden die Operanden für die Referenzwerte angegeben. Der Wert des Operanden am Eingang E1 wird mit den Referenzwerten verglichen.

AC0 ... ACn-1 DOPPELWORT

Durch die Dopplung des Eingangs EC0 wird der Eingang AC0 automatisch mitgedoppelt. An den Eingängen AC0 ... ACn-1 werden die Ausgangscodes angegeben. Bei Übereinstimmung des Eingangs E mit einem der Referenzwerte ECi wird der Ausgangscode ACi am Ausgang A ausgegeben.

Zuordnung Referenzwerte und Ausgangscodes:

EC0 <-> AC0

EC1 <-> AC1

...

...

ECn-1 <-> ACn-1

E=EC BINÄR

Am Ausgang E=EC wird die Übereinstimmung des Operandenwertes des Eingangs E mit einem der Referenzwerte signalisiert.

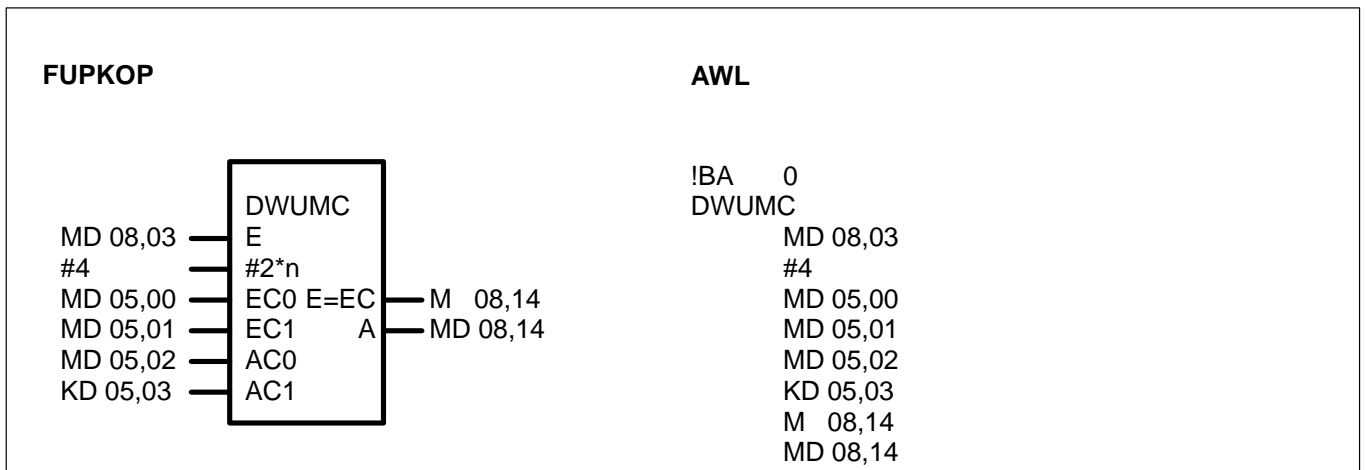
Es gilt: E=EC = 0 -> keine Übereinstimmung
E=EC = 1 -> Übereinstimmung

A DOPPELWORT

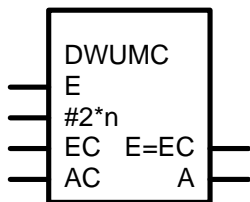
Bei Übereinstimmung des Eingangs E mit einem der Referenzwerte ECi wird der Ausgangscode ACi am Ausgang A ausgegeben.

Es gilt: A = 0 -> keine Übereinstimmung
A = ACi -> Übereinstimmung

Beispiel



VE-FUP-Definition

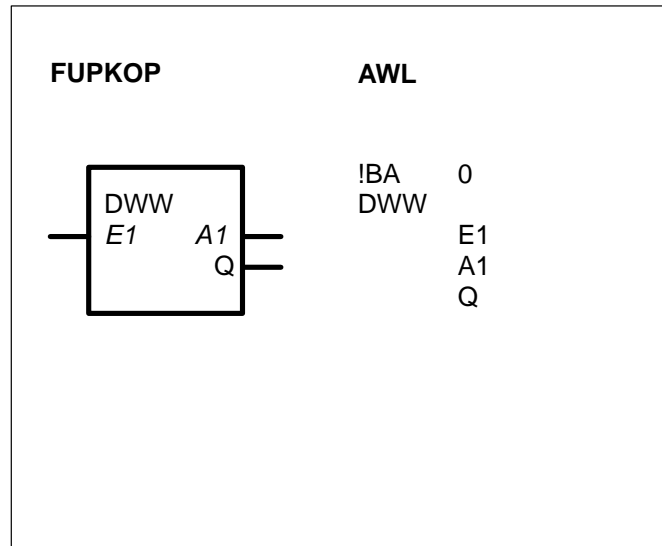


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E	E	D	N	P	J	0	0
#2*n	K	W	N	P	J	0	0
EC	E	D	N	P	J	1	0
AC	E	D	N	P	J	1	0
E=EC	A	L	N	P	J	0	0
A	A	D	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DWUMC			
00002		PP 0	E	Eingang DOPPELWORT (Eingangswert)
00003		PP 0	#2*n	# KONSTANTE (Anzahl EC und AC)
	[1		
00004		PP 1	EC	Eingang DOPPELWORT (Referenzwert)
]	1		
	[1		
00005		PP 1	AC	Eingang DOPPELWORT (Ausgangscode)
]	1		
00006		PP 0	E=EC	Ausgang BINÄR (Eing.wert=Ecode)
00007		PP 0	A	Ausgang DOPPELWORT (Ausgangswert)

Der Wert des Doppelwort-Operanden am Eingang E1 wird in eine Wortgröße gewandelt, und das Ergebnis dem Wort-Operanden am Ausgang A1 zugewiesen.



Parameter

E1	DOPPELWORT	MD, KD	zu wandelnde Doppelwortgröße
A1	WORT	AW, MW	Ergebnis der Wandlung, Wortgröße
Q	BINÄR	A, M	Ergebnis begrenzt

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

36...38 µs

154 µs

Zusätzliche Laufzeit:

—

—

Aktualisierung der Ausgänge:

ja

ja

Anzahl der Vergangenheitswerte:

keine

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

907 PC 331 / 07 KR 31 V 2.0,

07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Wert des Doppelwort-Operanden am Eingang E1 wird in eine Wortgröße gewandelt, und das Ergebnis dem Wort-Operanden am Ausgang A1 zugewiesen.

Das Ergebnis wird auf den maximalen bzw. minimalen Zahlenbereich begrenzt.

max. Zahlenbereich: +32 767 (7FFF_H)

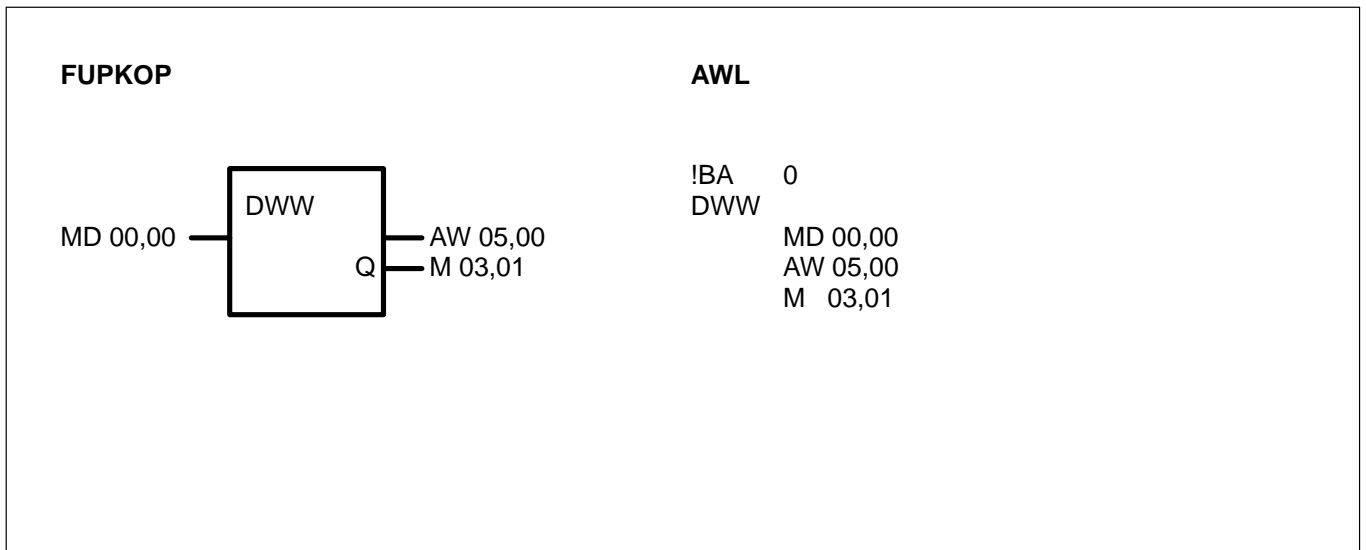
min. Zahlenbereich: - 32 767 (8001_H)

Fand eine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 1-Signal zugewiesen. Fand keine Begrenzung statt, wird dem binären Operanden am Ausgang Q ein 0-Signal zugewiesen.

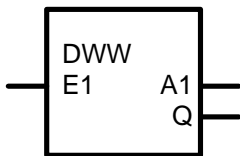
Falls der Wert des Operanden am Eingang E1 außerhalb des zulässigen Zahlenbereiches liegt (Wert 8000 0000_H), wird der gewandelte Wert auf -32 767 (8001_H) gesetzt.

Der Eingang und die Ausgänge sind weder doppelbar noch negierbar.

Beispiel



VE-FUP-Definition

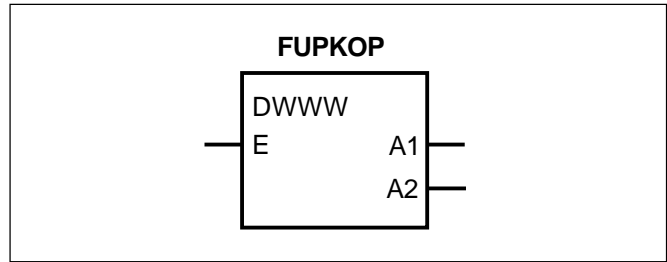


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
A1	A	W	N	P	N	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DWW			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	A1	Ausgang WORT
00004		PP 0	Q	Ausgang BINÄR

Der Baustein packt ein 32-Bit-Doppelwort in 2 16-Bit-Worte. Das Low Word des Doppelwort-Eingangs E (Bit 0 bis Bit 15) wird dem Ausgang A1 und das High Word des Eingangs E (Bit 16 bis 31) dem Ausgang A2 zugewiesen.



Parameter

E	DOPPELWORT	MD	Doppelworteingang
A1	WORT	AW, MW	Ausgang; Low Word des Eingangs E
A2	WORT	AW, MW	Ausgang; High Word des Eingangs E

VE-Daten

Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine

Verfügbar ab: 07 KR 91 R2 Index c, 07 KT 92 R2 Index c, 07 KT 93 R2 Index f, 07 KT 94

Beschreibung

Der Baustein packt ein 32-Bit-Doppelwort in 2 16-Bit-Worte. Das Low Word des Doppelwort-Eingangs E (Bit 0 bis Bit 15) wird dem Ausgang A1 und das High Word des Eingangs E (Bit 16 bis 31) dem Ausgang A2 zugewiesen.

A1 WORT

Am Ausgang A1 wird das Low Word (Bit 0 bis Bit 15) des Eingangs E ausgegeben.

A2 WORT

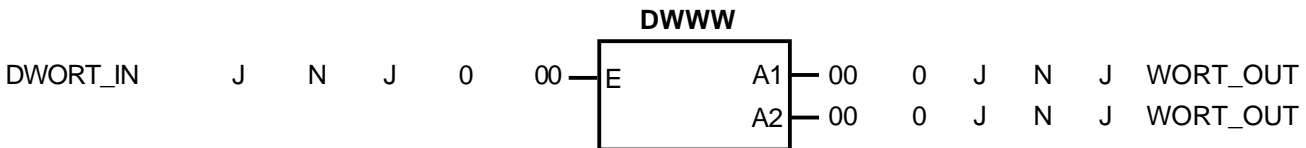
Am Ausgang A2 wird das High Word (Bit 16 bis Bit 31) des Eingangs E ausgegeben.

E DOPPELWORT

Das Doppelwort am Eingang E wird in 2 16-Bit-Worte E1 und E2 zerlegt.

Grafik

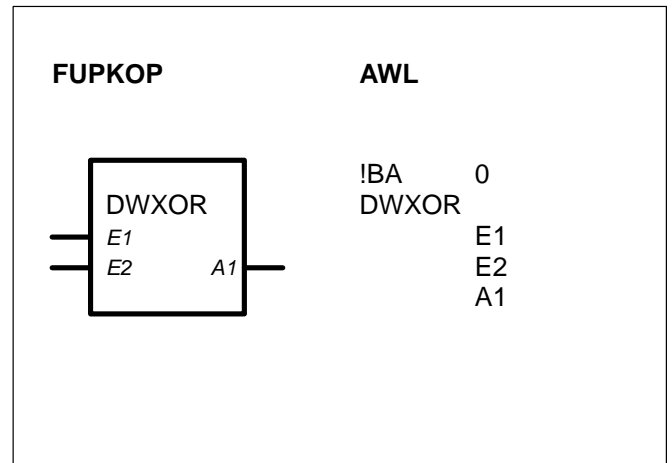
Plaus-gruppe	Pf1	Inv	Anz	Art	Nr		Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	--	----	-----	-----	-----	-----	--------------



Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	DWWW		
	00001	PP 00	E	
	00002	PP 00	A1	
	00003	PP 00	A2	
	00004			

Der Funktionsbaustein bildet die bitweise XOR-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	DOPPELWORT MD, KD	Operand 1
E2	DOPPELWORT MD, KD	Operand 2
A1	DOPPELWORT MD	Ergebnis der XOR-Verknüpfung

VE-Daten

Laufzeit:	
Grundlaufzeit:	35 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

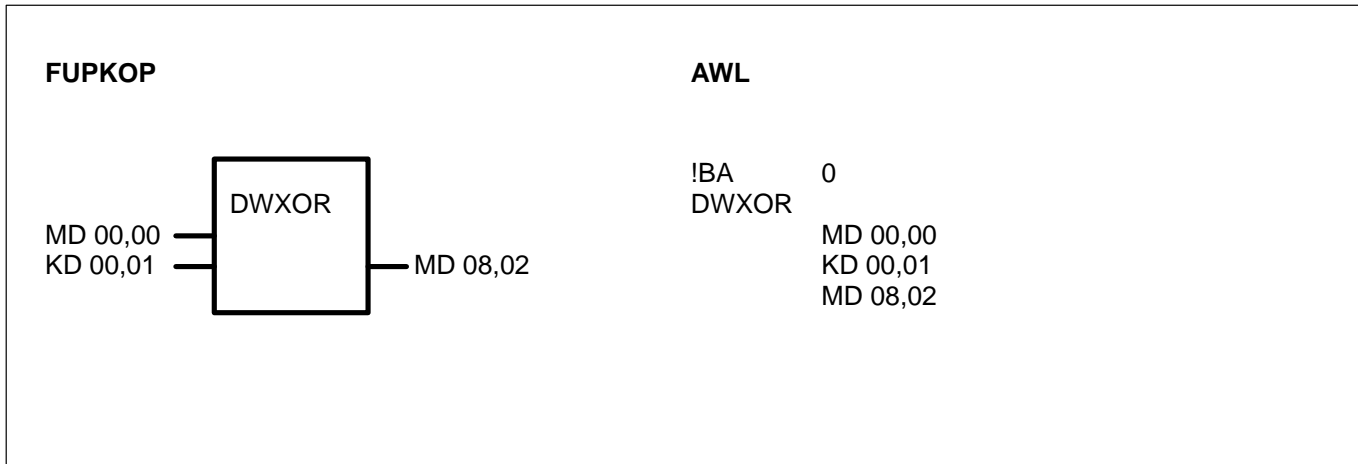
Der Funktionsbaustein bildet die bitweise XOR-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar/invertierbar.

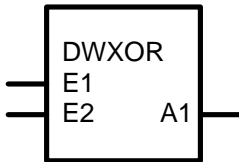
Beispiel

E1	1.0.0.0	0.0.1.1	0.0.1.0	0.1.1.0	1.0.1.0	1.1.0.0	0.0.1.1	0.1.0.1
E2	1.0.0.1	0.1.1.0	0.0.1.0	1.1.1.1	1.1.1.1	0.0.0.0	0.1.1.0	1.1.0.0
A1	0.0.0.1	0.1.0.1	0.0.0.0	1.0.0.1	0.1.0.1	1.1.0.0	0.1.0.1	1.0.0.1

Beispiel



VE-FUP-Definition



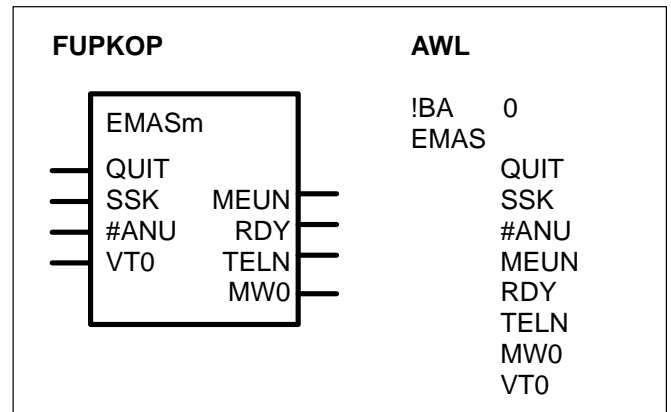
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
E2	E	D	N	P	N	0	0
A1	A	D	N	P	N	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	DWXOR			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	E2	Eingang DOPPELWORT
00004		PP 0	A1	Ausgang DOPPELWORT

Der Funktionsbaustein EMAS

- empfängt Telegramme über eine serielle Schnittstelle der SPS
- vergleicht diese Telegramme mit im Anwenderprogramm hinterlegten Vergleichstelegrammen
- und stellt bei Übereinstimmung die Nutzdaten des empfangenen Telegramms an den BausteinAusgängen zur Verfügung.



Parameter

QUIT	BINÄR	A, E, M, S, K	keine Freigabe zum Empfang von Telegrammen
SSK	WORT	AW, EW, MW, KW	Schnittstellenkennung; Nr. der seriellen Schnittstelle
#ANU	DIREKTE KONSTANTE	#, #H	Anzahl Ausgänge für Nutzinformation
VT0	ALLE	Texte	Vergleichstelegramm; doppelbarer Eingang
MEUN	BINÄR	A, M	Daten ungültig (Merker ungültig)
RDY	BINÄR	A, M	Telegramm wurde empfangen
TELN	WORT	AW, MW	Nummer des Vergleichstelegramms, mit dem das empfangene Telegramm übereinstimmt
MW0	WORT	AW, MW	Nutzinformation; doppelbarer Ausgang

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	liegt nicht vor	332 µs
Zusätzliche Laufzeit:	liegt nicht vor	135 µs pro Zeichen
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	1 Wort	1 Wort
Verfügbar ab:	ABB Procontic CS31 / Kommunikationsmodul 07 KP 62	07 KR 31 V 2.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein EMAS

- empfängt Telegramme über eine serielle Schnittstelle der SPS
- vergleicht diese Telegramme mit im Anwenderprogramm hinterlegten Vergleichstelegrammen
- und stellt bei Übereinstimmung die Nutzdaten des empfangenen Telegramms an den BausteinAusgängen zur Verfügung.

Die Empfangstelegramme werden von einem Schnittstellentreiber von der seriellen Schnittstelle abgeholt und für den EMAS zur weiteren Bearbeitung in einem BUFFER bereitgestellt. Der Treiber erkennt am Telegramm-Abschlußzeichen das Telegrammende. Dieses Telegramm-Abschlußzeichen wird beim Baustein SINIT projiziert.

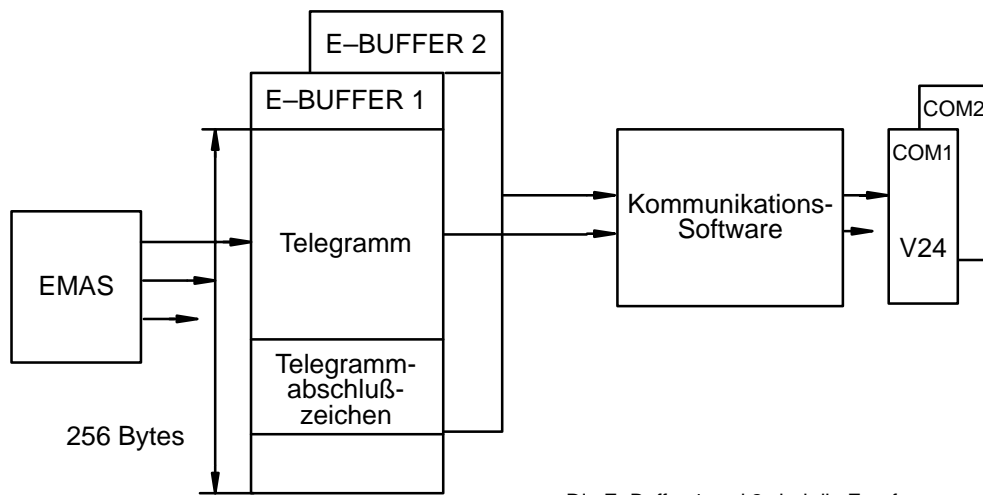
Initialisierung der seriellen Schnittstelle

Bevor der EMAS mit einer der möglichen seriellen Schnittstellen kommunizieren kann, muß diese mit dem Baustein SINIT initialisiert werden. Beim Baustein SINIT wird außer den gewünschten Schnittstellenparametern angegeben:

- ein frei projektierbares Telegramm-Abschlußzeichen um das Ende des zu empfangenden Telegramms zu erkennen
- ob ein an der Schnittstelle empfangenes Zeichen gechot werden soll oder nicht. Die Synchronisation des Datenflusses erfolgt prinzipiell über die Signale RTS und CTS.

Folgende seriellen Schnittstellen können vom Baustein EMAS benutzt werden:

Zentraleinheit 07 KR 91:	COM1
Zentraleinheit 07 KT 92:	COM1, COM2
Kommunikationsprozessor 07 KP 62:	COM1, COM2
Zentraleinheit 07 KR 31:	COM1



Die E-Buffer 1 und 2 sind die Empfangs-Buffer, die den seriellen Schnittstellen COM1 und COM2 zugeordnet sind.

Kurzübersicht der Baustein-Parameter:			
– QUIT – RDY – MEUN			
QUIT	MEUN	RDY	Bedeutung
1	0	0	Durch QUIT = 1 ist der EMAS gesperrt. Die Ausgänge MEUN und RDY werden dabei permanent auf 0 gesetzt.
0	1	0	EMAS ist zum Empfang freigegeben, es wurde aber noch kein Telegramm empfangen und ausgewertet.
0	0	1	EMAS hat ein gültiges Telegramm empfangen, und er ist bereit ein neues Telegramm zu empfangen.
0	1	1	EMAS hat ein ungültiges Telegramm empfangen. Um ein neues Telegramm empfangen zu können, ist eine Quittierung am Eingang QUIT erforderlich. QUIT: 0→1
0→1	0	0	Quittierung nach Empfang eines ungültigen Telegramms. <i>Nach</i> der Quittierung wird durch QUIT = 0 der EMAS wieder freigegeben.

QUIT BINÄR

Der Eingang QUIT steuert den Empfang von Telegrammen und dient außerdem zur Quittierung im Fehlerfall.

- QUIT = 0: Freigabe für den Empfang von Telegrammen.
- QUIT = 1: *keine* Freigabe für den Empfang von Telegrammen.
Quittierung nach dem Empfang eines ungültigen Telegramms.

Wird beim Vergleich eines empfangenen Telegramms mit *keinem* der hinterlegten Vergleichstelegramme Übereinstimmung festgestellt, dann geht der EMAS automatisch in den Betriebszustand "Fehler". In diesem Fall wird vom EMAS solange kein neues Telegramm mehr bearbeitet, bis mit einem 1-Signal am Eingang QUIT der Fehler quittiert wird, und anschließend (nächster Zyklus) mit einem 0-Signal am Eingang QUIT der Empfang von Telegrammen wieder freigegeben wird.

SSK WORT

Am Eingang SSK (Schnittstellenkennung) wird die Nummer der Schnittstelle angegeben, über die der Baustein seine Telegramme empfängt.

Es gilt:

- COM1: Nummer = 1
- COM2: Nummer = 2

#ANU DIREKTE KONSTANTE

Am Eingang #ANU (Anzahl Nutzinformatoren) wird die Anzahl der Ausgänge MW0 ... MWn angegeben, an de-

nen der Baustein die empfangenen Nutzinformatoren zur Verfügung stellt. Die Angabe erfolgt als direkte Konstante.

VT0 ... VTn ALLE

Der Eingang VT0 ... VTn ist doppelbar. An den Eingängen VT0 ... VTn werden die im SPS-Programm zu hinterlegenden Vergleichstelegramme angegeben. Der Baustein kann 1 ... 99 Telegramme bearbeiten. Ein Telegramm belegt zwei Eingänge, wobei jeweils die Telegrammnummer an einem Eingang angegeben wird und der eigentliche Telegrammtext am darauffolgenden Eingang. Die genaue Syntax und Handhabung der Vergleichstelegramme ist in dieser Beschreibung gesondert erläutert.

MEUN BINÄR

Am Ausgang MEUN (Merker ungültig) wird angezeigt, ob die Daten an den Ausgängen MW0 ... MWn gültig oder ungültig sind.

Wird ein Telegramm ordnungsgemäß empfangen und bearbeitet, so werden die Daten an den Ausgängen MW0 ... MWn für *gültig* erklärt. Stimmt das empfangene Telegramm mit keinem abgelegten Vergleichstelegramm überein, oder kann das empfangene Telegramm nicht ordnungsgemäß bearbeitet werden, so werden die Daten an den Ausgängen MW0 ... MWn für *ungültig* erklärt.

MEUN = 0 → Daten an den Ausgängen MW0 ... MWn sind *gültig*

MEUN = 1 → Daten an den Ausgängen MW0 ... MWn sind *ungültig*

RDY BINÄR

Am Ausgang RDY (Ready) wird angezeigt, daß ein Telegramm empfangen und bearbeitet wurde.

Der Ausgang RDY macht keine Aussage darüber, ob ein gültiges oder ungültiges Telegramm empfangen wurde.

RDY = 0 → Es wurde noch kein Telegramm empfangen

RDY = 1 → Ein Telegramm wurde empfangen und bearbeitet

TELN WORT

Wird ein gültiges Telegramm empfangen, so wird am Ausgang TELN (Telegrammnummer) die Nummer des zugehörigen Vergleichstelegrammes ausgegeben.

MW0 ... MWn WORT

Der Ausgang MW0 ist doppelbar (MW0 ... MWn).

An den Ausgängen MW0 ... MWn werden die Nutzdaten ausgegeben, die im gerade empfangenen Telegramm übermittelt werden. Diese Nutzdaten können Zahlenwerte oder beliebige Zeichen sein. Dies hängt davon ab, welche Art von Platzhaltern im Vergleichstelegramm projiziert wurden.

Die Nutzdaten eines Telegramms werden beginnend mit dem Ausgang MW0 abgelegt und zwar in der Reihenfolge, wie sie im Vergleichstelegramm projiziert sind. Es müssen so viele Ausgänge MW0 ... MWn vorgesehen werden, daß sie für das Telegramm mit den *meisten* Nutzdaten ausreichen.

Ausführliche Beschreibung der Funktionalität des Funktionsbausteins EMAS

Der Funktionsbaustein EMAS

- empfängt Telegramme über eine serielle Schnittstelle der SPS
- vergleicht diese Telegramme mit im Anwenderprogramm hinterlegten Vergleichstelegrammen
- und stellt bei Übereinstimmung die Nutzdaten des empfangenen Telegramms an den Bausteinausgängen zur Verfügung.

Die Empfangstelegramme werden von einem Schnittstellentreiber von der seriellen Schnittstelle abgeholt und für den EMAS zur weiteren Bearbeitung in einem BUFFER bereitgestellt. Der Treiber erkennt am Telegramm–Abschlußzeichen das Telegrammende. Dieses Telegramm–Abschlußzeichen wird beim Baustein SINIT projiziert.

Initialisierung der seriellen Schnittstelle

Bevor der EMAS mit einer der möglichen seriellen Schnittstellen kommunizieren kann, muß diese mit dem

Baustein SINIT initialisiert werden. Beim Baustein SINIT wird außer den gewünschten Schnittstellenparametern angegeben:

- ein frei projektierbares Telegramm–Abschlußzeichen um das Ende des zu empfangenden Telegramms zu erkennen
- ob ein an der Schnittstelle empfangenes Zeichen gecheckt werden soll oder nicht. Die Synchronisation des Datenflusses erfolgt prinzipiell über die Signale RTS und CTS.

Vergleichs–Telegramme

Direkt hinter dem Baustein EMAS sind 1 ... 99 Vergleichstelegramme abgelegt.

Die Vergleichstelegramme dienen zur Identifikation

- der empfangenen aktuellen Telegramme
- und der in den empfangenen Telegrammen enthaltenen Nutzdaten

Die abgelegten Vergleichstelegramme (V–Telegramme) haben jeweils zur Kennung eine Telegramm–Nummer. Jedes V–Telegramm darf bis zu 255 Zeichen lang sein.

Die V–Telegramme bestehen aus:

- ASCII–Zeichen, die nur zur Identifikation des empfangenen Telegramms dienen.
- Platzhalter für die zu empfangenden und an die Bausteinausgänge auszugebenden Nutzinformationen.

Bei den Platzhaltern für die Nutzinformationen unterscheidet der EMAS zwischen Platzhaltern für *Ziffern* und Platzhaltern für *Zeichen*.

Platzhalter für Ziffern: # (für jede Ziffer ein #)

Platzhalter für Zeichen: * (für jedes Zeichen/Byte ein *)

Platzhalter für Ziffern

Für jeden Ziffern–Platzhalter (#) des Vergleichstelegramms erwartet der EMAS im zu empfangenden Telegramm genau *eine* ASCII–codierte Dezimalziffer.

Maximal 5 Ziffernplatzhalter bilden eine Platzhalter–Gruppe. Solch eine Gruppe von Ziffernplatzhaltern repräsentiert den Zahlenwert einer maximal 5–stelligen Dezimalzahl.

Für das Vorzeichen der Dezimalzahl wird *kein* Platzhalter vorgegeben, da dieses vom EMAS automatisch berücksichtigt wird.

Der EMAS weist jeweils den Zahlenwert, der zu einer Platzhaltergruppe gehört, einem Nutzinfo-Ausgang zu. Dabei wird der Wertebereich vom EMAS automatisch auf ± 32767 beschränkt.

Platzhalter für Zeichen

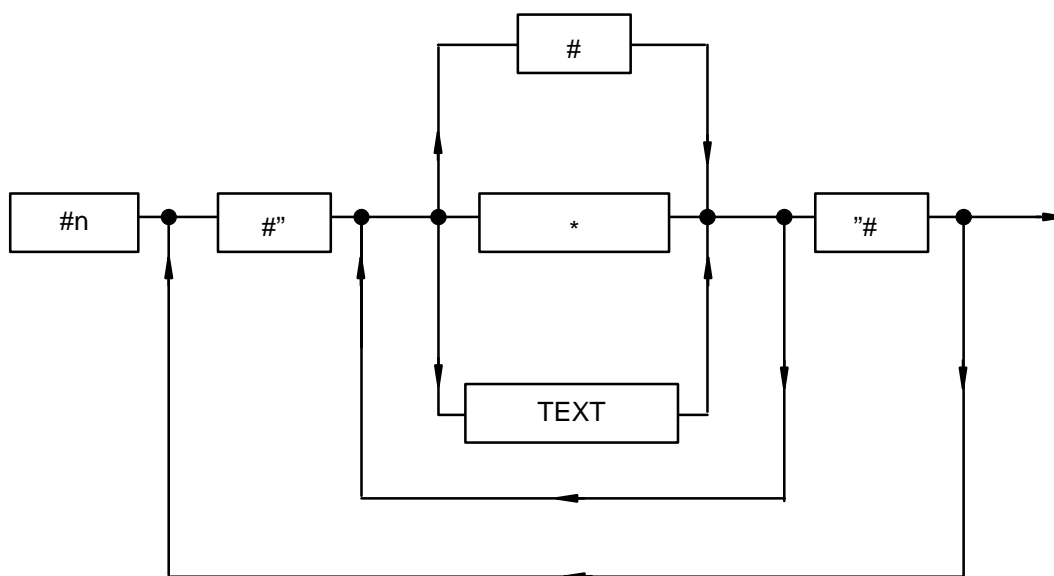
Für jeden Zeichen-Platzhalter (*) des Vergleichstelegramms erwartet der EMAS im zu empfangenden Telegramm *ein* beliebiges Zeichen/Byte. Dies können ASCII-

Zeichen von Buchstaben aber auch alle anderen HEX-Werte von 1 ... FF sein.

Die Länge einer Zeichen-Platzhaltergruppe ist maximal 255. In diesem Fall würde das ganze Vergleichs-Telegramm nur aus Zeichenplatzhaltern bestehen.

Die empfangenen Zeichen/Byte werden vom EMAS ohne Veränderung und fortlaufend seinen Nutzinfo-Ausgängen MW0 ... MWn zugewiesen.

Syntaxdiagramm: Aufbau der Vergleichstelegramme



#n : Fortlaufende Telegrammnummer (direkte Konstante 1...99)

#" : Anfangskennung für Texteingabe

"# : Ende-Kennung für Texteingabe im Terminalbetrieb

Achtung: Diese Endekennung wird bei Verwendung des ABB Procontic Programmiersystem *nicht* angegeben. Nur, wenn im Terminalbetrieb Texte direkt in die SPS eingegeben werden, müssen diese mit dieser Endekennung abgeschlossen werden.

* : Platzhalter für Zeichen/Byte

: Platzhalter für Ziffern

TEXT: Alle ASCII-Zeichen (1 ... FF) außer * und #

Eingabe der Vergleichstelegramme

Bei der Eingabe der Vergleichstelegramme sind zwei Fälle zu unterscheiden:

- Eingabe in das ABB Procontic Programmiersystem
- Eingabe mit einem Terminal direkt in die SPS

Eingabe der Vergleichstelegramme in das ABB Procontic Programmiersystem

Jedes Vergleichstelegramm besteht aus

- der Telegramm-Nummer
- dem Telegramm-Text

Die Telegramm-Nummer und der Telegramm-Text sind jeweils eigene Operanden. Die Telegramm-Nummer und der der Telegramm-Text belegen deshalb beim FUP-Symbol des EMAS-Bausteins jeweils einen eigenen Eingang. Für ein Vergleichstelegramm werden also zwei Eingänge benötigt.

Beispiel:

Eingang VT0: #1

(Nr. des ersten V-Telegramms)

Eingang VT1: #"DRUCK### KENNUNG****

(Text des ersten V-Telegramms)

Außer den ASCII-Zeichen für * und # sind im Telegramm-Text alle ASCII-Zeichen möglich.

Bei der Eingabe von speziellen ASCII-Zeichen wie z. B. "Zeilenanfang" <CR> ist beim ABB Programmiersystem folgendes zu beachten:

Die Eingabe der Sonderzeichen erfolgt durch:

 \Zahlenwert des Zeichens

Der Zahlenwert des Zeichens wird dabei als dreistellige Dezimalzahl angegeben.

Beispiel:

Folgender Telegrammtext soll in das ABB Programmiersystem eingegeben werden:

 Temperatur <CR> Kessel 1

Es gilt: <CR> = 013

Die Eingabe des Textes in das Programmiersystem sieht wie folgt aus:

 #"Temperatur\013Kessel 1

Hinweis:

Die Zeichen mit ASCII-Code >20_H bzw. 32_D müssen über die Tastatur eingegeben werden. So muß beispielsweise anstatt \033 das diesem Code entsprechende "!" eingegeben werden.

Zeichen, die keine Sonderzeichen sind und auch nicht direkt über die Tastatur eingegeben werden können, werden durch Drücken der <ALT>-Taste und Eingeben des Zahlencodes (numerische Tastatur) eingetragen.

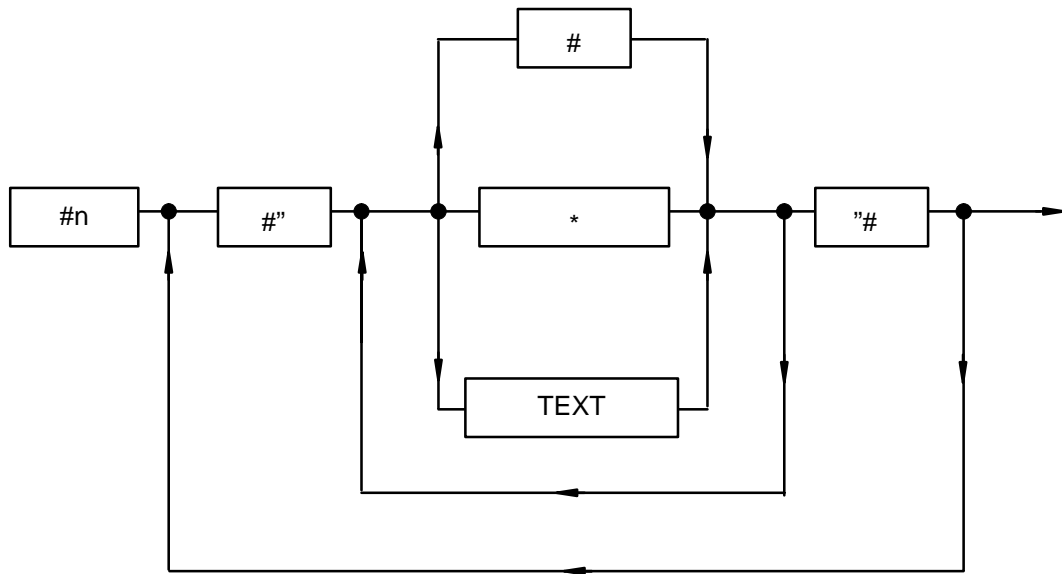
Das Zeichen mit dem ASCII-Code 255 ist innerhalb der Programmiersoftware reserviert und darf nicht verwendet werden.

Eingabe der Vergleichstelegramme mit einem Terminal direkt in die SPS

Die Eingabe der Telegramme erfolgt direkt im Anschluß an den EMAS-Baustein, d. h. das erste Vergleichstelegramm beginnt bei der Programmspeicher-adresse, die auf den Operanden am Ausgang MWn folgt.

Die Telegramm-Nummer wird als direkte Konstante eingegeben. Die Eingabe des Telegramm-Textes erfolgt eingebettet in die Sonderzeichen #" (Anfang der Texteingabe) und "# (Ende der Texteingabe). Außer den ASCII-Zeichen für * und # sind dabei alle ASCII-Zeichen von 1...FF möglich; das ASCII-Zeichen <NUL> wird als Präfix für die Formatkennung im Sendebuffer benutzt.

Syntaxdiagramm: Texteingabe direkt in die SPS mit einem Terminal



#n : Fortlaufende Telegrammnummer (direkte Konstante 1 ... 99)

#\" : Anfangskennung für Texteingabe

\"# : Ende-Kennung für Texteingabe im Terminalbetrieb

* : Platzhalter für Zeichen / Byte

: Platzhalter für Ziffern

TEXT: Alle ASCII-Zeichen (1 ... 7F) außer * und #

Korrektur eines Zeichens innerhalb einer Texteingabe im Terminalbetrieb

Bei der Eingabe eines Textes kann ein falsch eingegebenes Zeichen *nicht* korrigiert werden. Die Texteingabe muß zur Korrektur mit der Texteingabe-Endekennung \"# abgeschlossen und die Programmeingabe mit <CR> beendet werden. Mit der Programmeingabe ist bei der Adresse des zu korrigierenden Zeichens neu aufzusetzen. Die Texteingabe beginnt dann mit der Texteingabe-Anfangskennung #\".

Texteingabe von Texten mit mehr als 80 Zeichen über ein Terminal

Auf dem Bildschirm können bei einer Texteingabe max. 80 Zeichen pro Bildschirmzeile angezeigt werden. Ist der Text länger als 80 Zeichen, so muß die Texteingabe am Ende der Bildschirmzeile mit der Texteingabe-Endekennung \"# abgeschlossen werden und mit dem Zeichen ; die nächste Bildschirmzeile angesprungen werden. Mit der Texteingabe-Anfangskennung #\" wird nun die Texteingabe fortgesetzt.

Empfang und Bearbeitung von Telegrammen

Empfang von Telegrammen

Der Eingang QUIT steuert den Empfang von Telegrammen und dient außerdem zur Quittierung im Fehlerfall.

- QUIT = 0: Freigabe für den Empfang von Telegrammen.
- QUIT = 1: *keine* Freigabe für den Empfang von Telegrammen. Quittierung nach dem Empfang eines ungültigen Telegramms.

Wird beim Vergleich eines empfangenen Telegramms mit *keinem* der hinterlegten Vergleichstelegramme Übereinstimmung festgestellt, dann geht der EMAS automatisch in den Betriebszustand "Fehler". In diesem Fall wird vom EMAS solange kein neues Telegramm mehr bearbeitet, bis mit einem 1-Signal am Eingang QUIT der Fehler quittiert wird, und anschließend (nächster Zyklus) mit einem 0-Signal am Eingang QUIT der Empfang von Telegrammen wieder freigegeben wird.

Bearbeitung von Telegrammen

Der EMAS vergleicht ein empfangenes Telegramm Zeichen für Zeichen mit den abgelegten Vergleichs-Telegrammen. Dabei wird auch die Übereinstimmung der Platzhalter im Vergleichstelegramm und der zugehörigen aktuellen Zeichen im empfangenen Telegramm geprüft.

Achtung: Die Telegrammendekennung kennzeichnet für den EMAS das Ende des zu empfangenden Telegramms. *Innerhalb* eines Telegramms darf deshalb die Telegrammendekennung *nicht* vorkommen. Dies gilt auch für Nutzzeichen, die aufgrund eines Platzhalters (# bzw. *) empfangen werden.

Übereinstimmung liegt vor

Stellt der EMAS Übereinstimmung zwischen dem empfangenen Telegramm und einem seiner Vergleichs-Telegramme fest, so gilt für die Ausgänge:

MEUN	= 0, da das Telegramm <i>ordnungsgemäß</i> empfangen und bearbeitet werden konnte
RDY	= 1, da ein Telegramm <i>empfangen</i> und <i>bearbeitet</i> wurde
TELN	= Nummer des relevanten Vergleichstelegramms
MW0 ... MWn	= Aktuelle Nutzdaten aus dem empfangenen Telegramm

Die zusammengehörenden Ziffern, die im Vergleichstelegramm durch eine Ziffernplatzhalter-Gruppe definiert sind, werden aus dem empfangenen Telegramm gelesen, zu *einem* Zahlenwert zusammengefügt und dieser an *einen* Wort-Operanden (Nutzinfo-Ausgang) ausgegeben. Textzeichen, die im Vergleichstext durch Textzeichen-Platzhalter (*) markiert sind, werden aus dem empfangenen Telegramm gelesen und jedes *einzelne* Zeichen/Byte unverändert an einen Wort-Operanden (Nutzinfo-Ausgang) ausgegeben.

Die Zuordnung der Nutzinformationen zu den Ausgängen MW0 ... MWn erfolgt in der selben Reihenfolge wie die Nutzinformationen im gerade empfangenen Telegramm auftreten (s. a. Beispiele).

Am Eingang #ANU wird die max. Anzahl der auszugehenden Nutzinfo-Wortoperanden vorgeben. Die max. Anzahl ergibt sich aus dem Telegramm mit den meisten zu reservierenden Nutzinfo-Wortoperanden (max. 256 Wortoperanden, dabei würde das Telegramm *nur* aus Nutzinformationen bestehen).

Sonderfall:

Wenn in einem V-Telegramm mehr Nutzinfos projiziert werden, als Nutzinfo-Ausgänge am Baustein zur Verfügung stehen, d. h. am Eingang #ANU angegeben sind, so werden die empfangenen Nutzdaten solange an den Nutzinfo-Ausgängen ausgegeben, bis das Ende der Ausgänge erreicht ist. Die übrigen empfangenen Nutzdaten werden nicht ausgegeben.

In diesem Fehlerfall gilt:

MEUN	= 1, da das Telegramm <i>nicht</i> ordnungsgemäß bearbeitet werden konnte
RDY	= 1, da ein Telegramm empfangen und bearbeitet wurde
TELN	= Text-Nr. des relevanten Vergleichstelegramms
MW0...MWn	= Der Teil der aktuellen Nutzdaten aus dem empfangenen Telegramm, für die die Anzahl der projizierten Nutzinfo-Ausgänge ausreicht.

Keine Übereinstimmung liegt vor

Das empfangene Telegramm stimmt mit *keinem* der hinterlegten Vergleichs-Telegramme überein.

Für die Ausgänge gilt dann:

MEUN	= 1, da das empfangene Telegramm ungültig ist
RDY	= 1, da ein Telegramm empfangen und seine Bearbeitung beendet ist

Der EMAS befindet sich nun im Betriebszustand "FEHLER". In diesem Fall wird vom EMAS solange kein neues Telegramm bearbeitet, bis mit einem 1-Signal am Eingang QUIT der Fehler quittiert wird, und anschließend (nächster Zyklus) mit einem 0-Signal am Eingang QUIT der Empfang von Telegrammen wieder freigegeben wird.

Quittierung des Fehlers

QUIT = 1 und im nächsten Zyklus QUIT = 0 setzen.

Nach der Freigabe des Empfangs von Telegrammen kann es sein, daß noch kein Telegramm zum Empfang ansteht, oder daß ein Telegramm noch nicht vollständig empfangen und ausgewertet wurde.

In diesem Fall werden die Ausgänge

- MEUN	= 1	und
- RDY	= 0	

gesetzt, bis ein Telegramm vollständig empfangen und bearbeitet wurde.

Kommunikation mehrerer EMAS–Bausteine mit derselben seriellen Schnittstelle

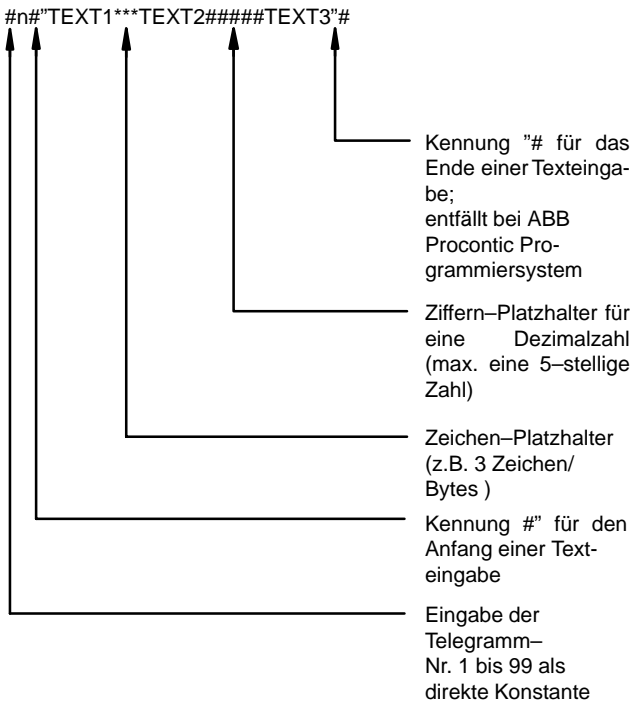
- EMAS–Bausteine eines Anwenderprogramms, die auf *dieselbe* serielle Schnittstelle zugreifen, sind so gegeneinander zu verriegeln, daß immer nur *ein* EMAS–Baustein aktiv ist. Wird dies unterlassen, so können Telegramme vom falschen EMAS bearbeitet und für ungültig erklärt werden.
- Sind sowohl im Anwenderprogramm 1 als auch im Anwenderprogramm 2 EMAS–Bausteine vorhanden, die auf *dieselbe* serielle Schnittstelle zugreifen, so sind sie so gegeneinander zu verriegeln, daß immer nur *ein* EMASm–Baustein aktiv ist. Wird dies unterlassen, so können Telegramme vom falschen EMAS bearbeitet und für ungültig erklärt werden.

Ein Telegrammverlust kann durch eine gegenseitige Verriegelung der EMAS–Bausteine vermieden werden. Die Verriegelung muß so projektiert werden, daß immer nur derjenige EMAS–Baustein freigegeben wird, für den das über die Schnittstelle ankommende Telegramm bestimmt ist.

Kommunikation eines EMAS–Bausteins und eines DRUCK–Bausteins mit derselben seriellen Schnittstelle

Ein EMAS–Baustein und ein DRUCK–Baustein können *dieselbe* serielle Schnittstelle benutzen, ohne daß besondere Vorkehrungen zu treffen sind.

Beispiel eines Vergleichstelegramms



Anmerkungen:

- Die im Beispiel mit TEXT1, TEXT2, TEXT3 bezeichneten Texte sind festprojektierte Texte, die nicht vom EMAS auf Nutzinfo–Ausgänge ausgegeben werden. Sie dienen ausschließlich zur Identifikation eines ankommenden Telegramms.

Die Reihenfolge von Texten, Platzhaltern für Zeichen und Ziffern ist beliebig.

- Als Platzhalter für Ziffern können max. fünf # angegeben werden. Für jede Ziffer der Dezimalzahl muß genau *ein* Platzhalter (#) im Vergleichstelegramm angegeben werden. Für das Vorzeichen der Dezimalzahl ist *kein* Platzhalter vorzugeben. Das Vorzeichen der Dezimalzahl wird vom EMAS automatisch berücksichtigt.

z.B.:	Dezimalzahl	Platzhalter
	1234	#####
	+1234	#####
	–1234	#####

Der Baustein EMAS überprüft die empfangene Dezimalzahl auf ihren Wertigkeitsbereich. Es können in der SPS nur Zahlen im Bereich ±32767 bearbeitet werden. Überschreitet die empfangene Dezimalzahl den Wertigkeitsbereich, so wird vom Baustein EMAS automatisch der jeweilige max. Grenzwert eingesetzt. Bei einer positiven Zahl +32767, bei einer negativen Zahl –32767.

Weitere Beispiele von Vergleichs- telegrammen

Eingabe mit ABB Procontic Programmiersystem

Beispiel 1: Text 1: #1#"RO104

Beispiel 2: Text 2: #2#"DO##"##"###

↑
Texteingabe
abschließen und
wieder neu aufsetzen
s. a. Sonderfall

Beispiel 3: Text 3: #3#"CO##"****

Beispiel 4: Text 4: #4#"*****"###"

Sonderfall Beispiel 2:

Wenn das Textzeichen (") benutzt wird und dem Textzeichen ein Platzhalter für Ziffern (#) folgt, so muß die Eingabe nach dem Textzeichen (") beendet werden. Ein nachfolgender Text oder Platzhalter beginnt mit der Eingabe-Kennung einer neuen Texteingabe (#").

Eingabe im Terminalbetrieb direkt in die SPS

Beispiel 1: Text 1: #1#"RO104"#

Beispiel 2: Text 2: #2#"DO##"##"###"##

↑
Texteingabe
abschließen und
wieder neu aufsetzen
s. a. Sonderfall

Beispiel 3: Text 3: #3#"CO##"****"##

Beispiel 4: Text 4: #4#"*****"###"##"##

Sonderfall Beispiel 2:

Wenn das Textzeichen (") benutzt wird, und dem Textzeichen ein Platzhalter für Ziffern (#) folgt, so muß die Eingabe nach dem Textzeichen (") beendet werden. Ein nachfolgender Text oder Platzhalter beginnt mit der Eingabe-Kennung einer neuen Texteingabe (#").

Eingabe der Vergleichstelegramme und das entsprechende empfangene Telegramm

Anwender- eingabe	Bedeutung	über serielle Schnittstelle empfangenes Telegramm
#1	Telegramm-Nr. 1, direkte Konstante	
#"	Kennung für den Anfang der Texteingabe	
R	Buchstabe in ASCII	52 _H
O	" "	4F _H
1	Zahl in ASCII	31 _H
0	" "	30 _H
4	" "	34 _H
	Textabschlußzeichen z. B. <CR>	0D _H
"#	Kennung für das Ende der Texteingabe (<i>entfällt beim ABB Progr.syst.</i>)	
#2	Telegramm-Nr. 2, direkte Konstante	
#"	Kennung für den Anfang der Texteingabe	
D	Buchstabe in ASCII	44 _H
O	" "	4F _H
	Vorzeichen (+)	2B _H
#	1. Ziffer von 2 Ziffern	xx _H
#	2. Ziffer von 2 Ziffern	xx _H
"	Zeichen in ASCII	22 _H
"#	Kennung für das Ende der Texteingabe (<i>entfällt beim ABB Progr.syst.</i>)	
#"	Kennung für den Anfang der Texteingabe	
#	1. Ziffer von 3 Ziffern	yy _H
#	2. " " 3 "	yy _H
#	3. " " 3 "	yy _H
"	Zeichen in ASCII	22 _H
"#	Kennung für das Ende der Texteingabe (<i>entfällt beim ABB Progr.syst.</i>)	
	Textabschlußzeichen z. B. <CR>	0D _H
#3	Telegramm-Nr. 3, direkte Konstante	
#"	Kennung für den Anfang der Texteingabe	
C	Buchstabe in ASCII	43 _H
O	" "	4F _H
	Vorzeichen (-)	2D _H
#	1. Ziffer von 2 Ziffern	zz _H
#	2. " " 2 "	zz _H
*	1. Zeichen von 4 Zeichen	nn _H
*	2. " " 4 "	nn _H
*	3. " " 4 "	nn _H
*	4. " " 4 "	nn _H
	Textabschlußzeichen z. B. <CR>	0D _H
"#	Kennung für das Ende der Texteingabe (<i>entfällt beim ABB Progr.syst.</i>)	
#4	Telegramm-Nr. 4, direkte Konstante	
#"	Kennung für den Anfang der Texteingabe	
*	1. Zeichen von 5 Zeichen	mm _H
*	2. " " 5 "	"
*	3. " " 5 "	"
*	4. " " 5 "	"
*	5. " " 5 "	"
#	1. Ziffer von 3 Ziffern	xx _H
#	2. " " 3 "	"
#	3. " " 3 "	xx _H
*	1. Zeichen von 2 Zeichen	nn _H
*	2. " " 2 "	nn _H
	Textabschlußzeichen <CR>	0D _H
"#	Kennung für das Ende der Texteingabe (<i>entfällt beim ABB Progr.syst.</i>)	

Folgende Telegramme wurden empfangen und ausgewertet:

Beispiel 1

RO104<CR>

Am Ausgang TELN wird die Nummer 1 ausgegeben.

Die Ausgänge MW0 ... MWn werden *nicht* geschrieben, da keine Nutzinformation projiziert ist.

Da das empfangene Telegramm ausgewertet wurde und mit einem V-Telegramm übereinstimmt, gilt:

MEUN	= 0
RDY	= 1
TELN	= 1

Beispiel 2

DO+12"457"<CR>

Am Ausgang TELN wird die Nummer 2 ausgegeben.

Am Ausgang MW0 steht als Wortmerkerinhalt der Wert 12.

Am Ausgang MW1 steht als Wortmerkerinhalt der Wert 457.

Da das empfangene Telegramm ausgewertet wurde und mit einem Vergleichs-Telegramm übereinstimmt, gilt:

MEUN	= 0
RDY	= 1
TELN	= 2
MW0	= 12
MW1	= 457

Beispiel 3

CO-11AUTO<CR>

Am Ausgang TELN wird die Nummer 3 ausgegeben.

Am Ausgang MW0 steht als Wortmerkerinhalt -11.

Am Ausgang MW1 steht als Wortmerkerinhalt 41_H (A in ASCII).

Am Ausgang MW2 steht als Wortmerkerinhalt 55_H (U in ASCII).

Am Ausgang MW3 steht als Wortmerkerinhalt 54_H (T in ASCII).

Am Ausgang MW4 steht als Wortmerkerinhalt 4F_H (O in ASCII).

Da das empfangene Telegramm ausgewertet wurde und mit einem Vergleichs-Telegramm übereinstimmt, gilt:

MEUN	= 0
RDY	= 1
TELN	= 3
MW0	= -11
MW1	= A
MW2	= U
MW3	= T
MW4	= O

Beispiel 4

TEMPO120KM<CR>

Am Ausgang TELN wird die Nummer 4 ausgegeben.

Am Ausgang MW0 steht als Wortmerkerinhalt 54_H (T in ASCII).

Am Ausgang MW1 steht als Wortmerkerinhalt 45_H (E in ASCII).

Am Ausgang MW2 steht als Wortmerkerinhalt 4D_H (M in ASCII).

Am Ausgang MW3 steht als Wortmerkerinhalt 50_H (P in ASCII).

Am Ausgang MW4 steht als Wortmerkerinhalt 4F_H (O in ASCII).

Am Ausgang MW5 steht als Wortmerkerinhalt 120.

Am Ausgang MW6 steht als Wortmerkerinhalt 4B_H (K in ASCII).

Am Ausgang MW7 steht als Wortmerkerinhalt 4D_H (M in ASCII).

Da das empfangene Telegramm ausgewertet wurde und mit einem Vergleichs-Telegramm übereinstimmt, gilt:

MEUN	= 0
RDY	= 1
TELN	= 4
MW0	= T
MW1	= E
MW2	= M
MW3	= P
MW4	= O
MW5	= 120
MW6	= K
MW7	= M

Beispiel 5

XY25OTTO<CR>

Das empfangene Telegramm wurde ausgewertet, stimmt aber mit *keinem* der Vergleichs-Telegramme überein.

Daher gilt:

MEUN	= 1
RDY	= 1
TELN	= wird nicht zugewiesen

Der Empfang von weiteren Telegrammen ist gesperrt, bis am Eingang QUIT der Empfang des ungültigen Telegramms quittiert wird.

Telegramm-Abschlußzeichen

Der Empfangs-Treiber des Bausteins EMAS erkennt das Ende eines empfangenen Telegramms am Telegramm-Abschlußzeichen.

Beispiele für Telegramm-Abschlußzeichen, die beim Baustein SINIT anzugeben sind:

- <ETX> = 03_H = 3
- <EOT> = 04_H = 4
- <CR> = 0D_H = 13
- <LF> = 0A_H = 10
- <SP> = 20_H = 32
- · ·
- · ·

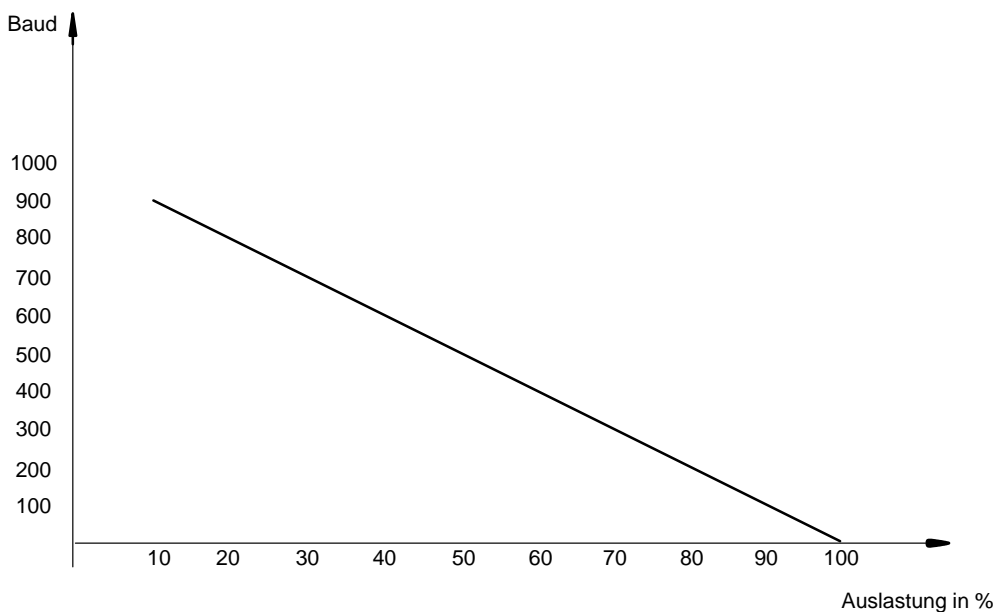
Effektive Datenübertragungsrate beim Baustein EMAS

Für 07 KR 31 nicht verfügbar.

Die effektive Datenübertragungsrate (Zeichen, die tatsächlich über die serielle Schnittstelle eingelesen werden) ist abhängig von der Auslastung der SPS und der eingestellten Baudrate.

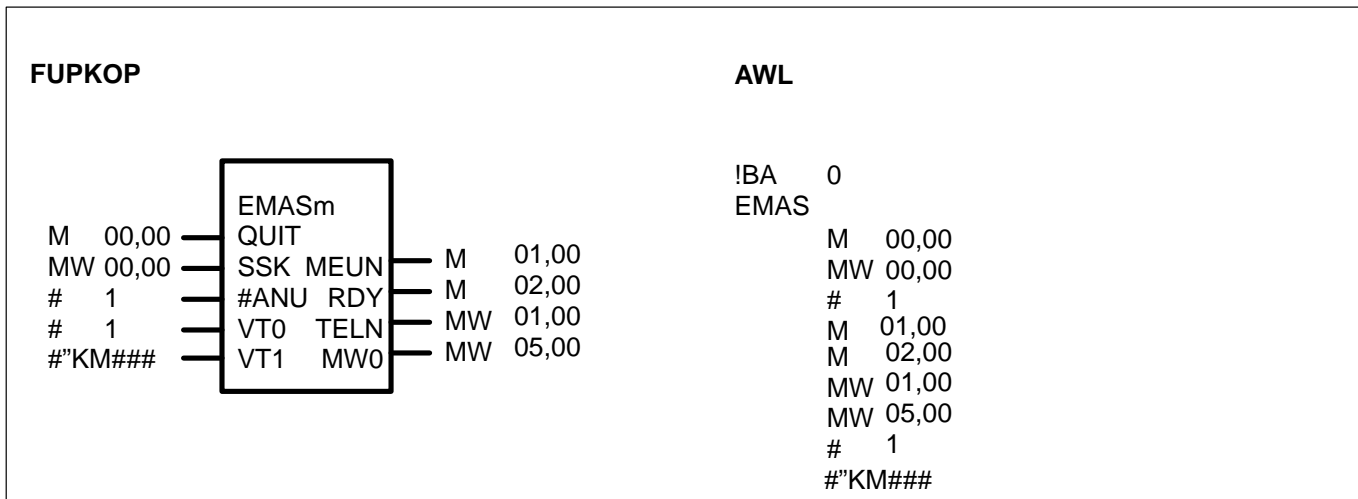
Es gilt grob folgender Zusammenhang:

Randbedingung: eingestellte Baudrate: 9600 Baud

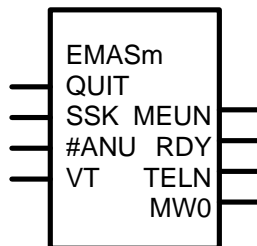


Achtung: Die Auslastungsanzeige (Befehl AL) ist nur korrekt, wenn zum Zeitpunkt der Anzeige *keine* Kommunikation über die seriellen Schnittstellen stattfindet. Durch die Kommunikation wird die Auslastungsanzeige verfälscht und damit unbrauchbar. Sind in einem Anwenderprogramm die Bausteine DRUCK oder EMAS vorhanden, so sind diese bei der Ermittlung der Auslastung zu sperren.

Beispiel



VE-FUP-Definition

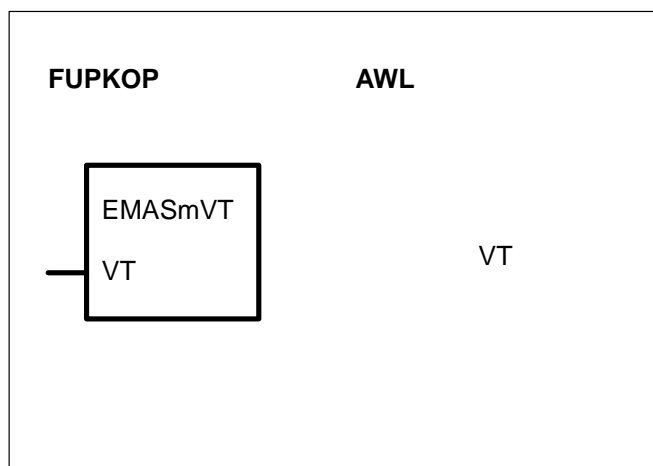


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
QUIT	E	L	N	P	J	0	0
SSK	E	W	N	P	J	0	0
#ANU	K	W	N	P	J	0	0
VT	E	X	N	P	J	2	0
MEUN	A	L	N	P	J	0	0
RDY	A	L	N	P	J	0	0
TELN	A	W	N	P	J	0	0
MW	A	W	N	P	J	1	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	EMAS			
00002		PP 0	QUIT	Eingang BINÄR (Quit,keine Freigabe)
00003		PP 0	SSK	Eingang WORT (Schnittstellenk)
00004		PP 0	#ANU	DIREKTE KONSTANTE (Anzahl MW's)
00005		PP 0	MEUN	Ausgang BINÄR (Daten ungültig)
00006		PP 0	RDY	Ausgang BINÄR (Text Empfang)
00007		PP 0	TELN	Ausgang WORT (Telegrammnummer)
[1			
00008		PP 1	MW	Ausgang WORT (Nutzinformation)
]		1		
[2		
00009		PP 2	VT	Eingang ALLE (Vergleichstext)
]		2		

Das Verknüpfungselement EMASmVT dient zur zusätzlichen Projektierung von Vergleichstelegrammen beim Baustein EMASm.



Parameter

VT ALLE

Vergleichstelegramm, doppelbarer Eingang

VE-Daten

Laufzeit:

Grundlaufzeit:

liegt nicht vor

Zusätzliche Laufzeit:

liegt nicht vor

Aktualisierung der Ausgänge:

entfällt

Anzahl der Vergangenheitswerte:

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

Beschreibung

Das Verknüpfungselement EMASmVT dient zur zusätzlichen Projektierung von Vergleichstelegrammen beim Baustein EMASm.

Beim Programmiersystem ist die Anzahl der Dopplungen eines Ein-/Ausgangs auf 63 begrenzt. Damit ist auch die Anzahl der projektierbaren Vergleichstelegramme auf max. 32 beschränkt.

Mit dem Verknüpfungselement EMASmVT lassen sich bei jedem Baustein EMASm weitere Vergleichstelegramme hinzufügen. Die maximale Anzahl der Vergleichstelegramme pro EMASm beträgt 99. Dabei werden die mit dem Verknüpfungselement EMASmVT hinzugefügten Vergleichstelegramme mitgezählt.

Alle am Baustein EMASm projektierten Vergleichstelegramme werden in der Anweisungsliste direkt hinter EMASm abgelegt. Die am Verknüpfungselement EMASmVT projektierten Vergleichstelegramme werden in der AWL direkt im Anschluß an die Vergleichstelegramme des Bausteins EMASm abgelegt.

Achtung:

Das/Die Verknüpfungselement(e) EMASmVT muß/müssen im FUP direkt nach dem Baustein EMASm projektiert werden.

Beim Baustein EMASm wird die maximal vorkommende Anzahl von Nutzausgängen (#ANU) angegeben. Bei dieser Angabe müssen auch die am Verknüpfungselement EMASmVT projektierten Vergleichstelegramme berücksichtigt werden.

VT Vergleichstelegramme

Am doppelbaren Eingang VT werden die Vergleichstelegramme projektiert. Die Projektierung ist vollkommen gleich wie am Eingang VT des Bausteins EMASm.

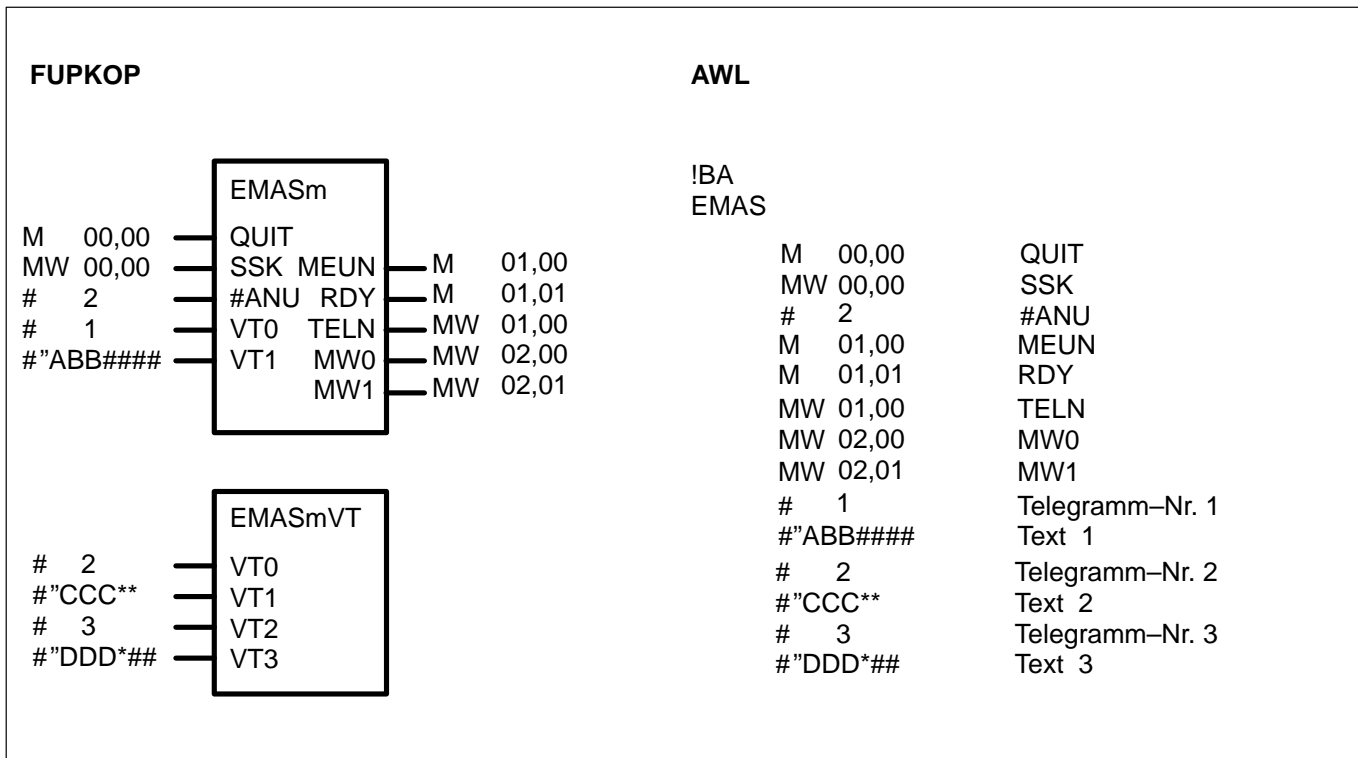
Hinweis:

Werden Vergleichstelegramme in der AWL nicht im Programmiersystem abgelegt, sondern mit einem Terminal direkt in die SPS eingegeben, so gilt:

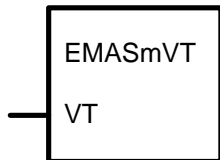
- der Vergleichstext beginnt mit der Anfangskennung: #"
- der Vergleichstext endet mit der Endkennung: "#

S. a. SPS-Beschreibung, ABB Procontic T300, Kap. Textverarbeitung.

Beispiel



VE-FUP-Definition



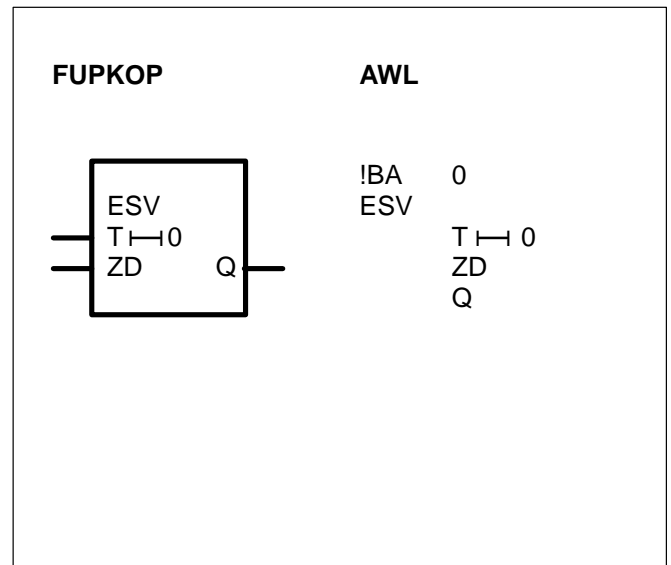
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
VT	E	X	N	P	J	1	0

VE-AWL-Definition

00000	[1					
		PP 1	VT		Eingang Vergleichstelegramme		
]	1					

Die 0–1–Flanke des Eingangs $T \rightarrow 0$ wird um die Zeitdauer ZD verzögert und als 0–1–Flanke am Ausgang Q ausgegeben.

Geht der Eingang $T \rightarrow 0$ vor Ablauf der Zeit ZD wieder auf 0–Pegel, so bleibt der Ausgang Q auf 0–Pegel.



Parameter

$T \rightarrow 0$	BINÄR	E, M, A, S, K	Eingangssignal
ZD	DOPPELWORT	MD, KD	Verzögerungszeit
Q	BINÄR	M, A	verzögertes Signal

VE–Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	38 μ s; Höchstlaufzeit 400 μ s, bei Flanken am Eingang $T \rightarrow 0$	97 μ s / 554 μ s
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	1 Wort	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Die 0–1–Flanke des Eingangs $T \rightarrow 0$ wird um die Zeitdauer ZD verzögert und als 0–1–Flanke am Ausgang Q ausgegeben.

Geht der Eingang $T \rightarrow 0$ vor Ablauf der Zeit ZD wieder auf 0–Pegel, so bleibt der Ausgang Q auf 0–Pegel.

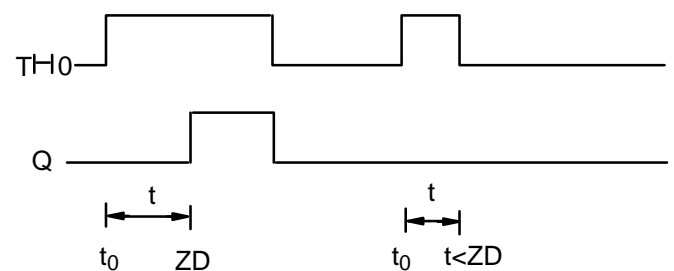
Die Zeitangabe erfolgt in Millisekunden, wobei nur ganzzahlige Vielfache von 5 ms zulässig sind (Bsp.: 5 ms, 10 ms, 500 ms, 100 000 ms ...). Angebbare Zeitbereich: 5 ms ... 24,8 Tage.

Bei der indirekten Konstanten KD wird der Wert der Verzögerungszeit im Langtext angegeben.

Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit

Sinnvoller Bereich für ZD: > 1 Zykluszeit

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar.



Allgemeines Verhalten

- Gestartete Zeitwerke werden vom Betriebssystem der SPS bearbeitet und sind deshalb vollkommen unabhängig von der Bearbeitung des SPS-Programms. Erst nach Ablauf des Zeitwerks erfolgt eine entsprechende Meldung des Betriebssystems an den zugehörigen Zeitbaustein im SPS-Programm.
- Die Bearbeitung eines Zeitwerks im Betriebssystem der SPS wird durch folgende Befehle *nicht* beeinflusst:
 - Programm abbrechen
 - Programm starten
 - Programm stoppen
 - Programm fortsetzen

D.h. ist ein Zeitwerk gestartet, so wird es im Betriebssystem der SPS auch dann weiter bearbeitet, wenn das zugehörige SPS-Programm abgebrochen, wieder gestartet oder gestoppt und wieder fortgesetzt wird.

Initialisierung

Die Zeitwerke werden grundsätzlich bei jedem Kalt- oder Warmstart der SPS initialisiert. Ein laufendes Zeitwerk wird also stets durch einen Kalt- oder Warmstart der SPS abgebrochen.

Kaltstart:

- Befehl KALT <CR>
- erstmaliges Zuschalten der Spannung

Warmstart:

- Befehl WARM <CR>
- Zuschalten der Spannung
- RESET-Schalter

Wichtiger Hinweis

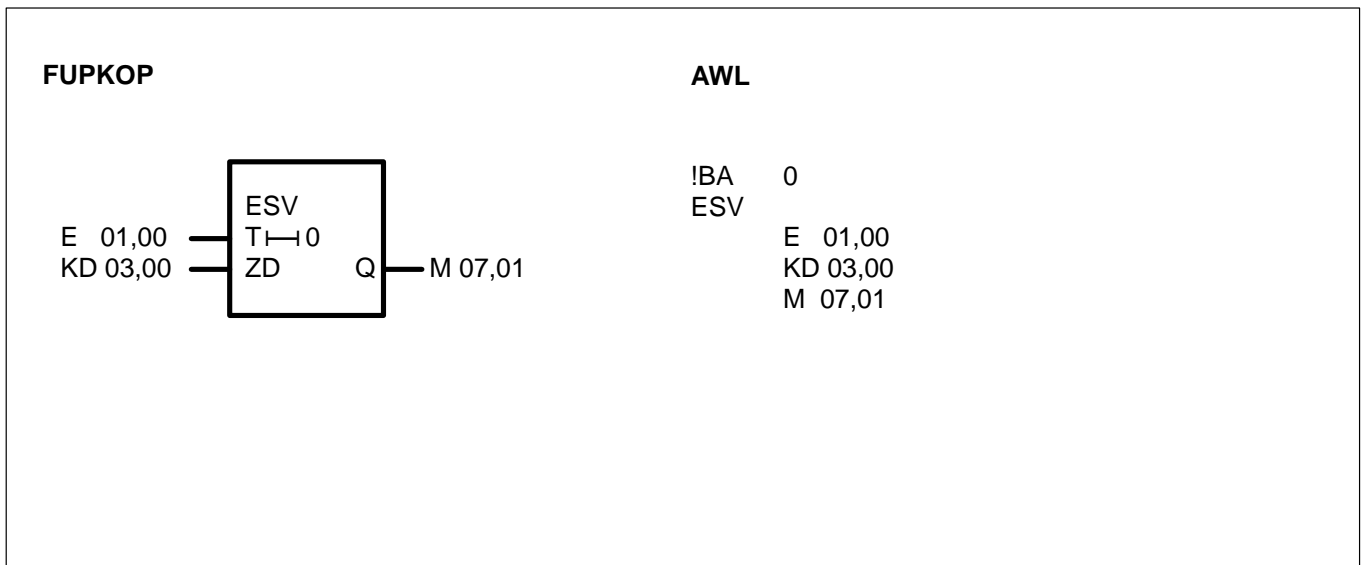
Das Verhalten der Zeitwerke ist geändert worden. Alle laufenden Zeitwerke werden bei folgenden Aktionen automatisch gestoppt:

- SPS-Programm abbrechen
- RUN/STOP-Schalter von RUN -> STOP

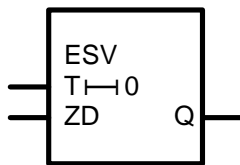
Diese Änderung gilt ab:

07 KR 91 Index g
07 KT 92 Index g
07 KT 93

Beispiel



VE-FUP-Definition



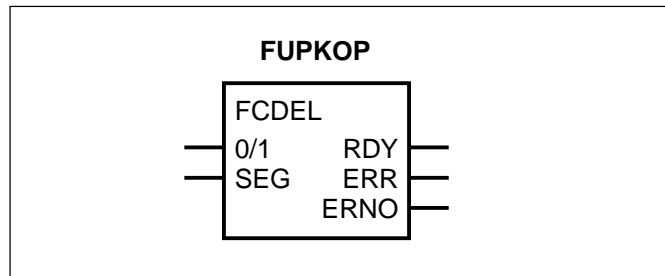
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
T I-10	E	L	N	P	J	0	0
ZD	E	D	N	P	J	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	ESV			
00002		PP 0	T I-10	Eingang BINÄR (Starte Zeit)
00003		PP 0	ZD	Eingang DOPPELWORT (Zeitwert)
00004		PP 0	Q	Ausgang BINÄR (Zeit)

Der Funktionsbaustein löscht ein Datensegment in der SmartMedia Card. Beim Löschvorgang gehen alle Daten in diesem Datensegment verloren.

Auf die SmartMedia Card kann auch über die Programmierschnittstelle mit Hilfe der Bedien- und Testfunktionen zugegriffen werden (Initialisierung der SmartMedia Card und Bedienfunktionen siehe AC31 Systembeschreibung Griff 7.3, Kapitel 2.4.1, Kommandos FCINIT, FCWR, FCRD, FCDEL).



Parameter

0/1	BINÄR	E, K, M, S
SEG	WORT	EW, MW
RDY	BINÄR	A, M
ERR	BINÄR	A, M
ERNO	WORT	AW, MW

Löschen eines Datensegments mit 0/1-Flanke
 Nummer des Datensegments, gültige Werte: 1...250
 Löschvorgang beendet
 Fehler aufgetreten, Datensegment nicht löschar
 Fehlernummer

VE-Daten

Aktualisierung der Ausgänge:	nein
Anzahl der Vergangenheitswerte:	2 Worte

Verfügbar ab: 07 KT 94 Index e

Beschreibung

Der Funktionsbaustein löscht ein Datensegment in der SmartMedia Card. Beim Löschvorgang gehen alle Daten in diesem Datensegment verloren.

Auf die SmartMedia Card kann auch über die Programmierschnittstelle mit Hilfe der Bedien- und Testfunktionen zugegriffen werden (Initialisierung der SmartMedia Card und Bedienfunktionen siehe AC31 Systembeschreibung Griff 7.3, Kapitel 2.4.1, Kommandos FCINIT, FCWR, FCRD, FCDEL).

Der Eingang **SEG** legt das Datensegment in der SmartMedia Card fest. Das Löschen eines Datensegments kann mehrere SPS-Zyklen dauern.

Mit einer 0/1-Flanke am Eingang 0/1 wird der einmalige Löschvorgang des Datensegments ausgelöst. Bis zur Beendigung des Löschvorgangs (RDY = 1) wird der Eingang nicht mehr ausgewertet.

Nach Beendigung des Löschvorgangs werden alle Baustein-Ausgänge aktualisiert. Bei RDY = 1 und ERR = 0 war das Löschen erfolgreich. Sind die Ausgänge RDY = 1 und ERR = 1, so konnte das Datensegment nicht gelöscht werden.

0/1 BINÄR

Über den Eingang 0/1 wird die Bearbeitung des Bausteines gesteuert. Es gilt:

0/1 = 0: Alle Ausgänge werden auf den Wert 0 gesetzt. Dies gilt nicht während eines Löschvorganges.

0/1 = 0/1-Flanke: Es wird der einmalige Löschvorgang des Datensegments ausgelöst. Bis zur Beendigung des Löschvorganges (RDY = 1) wird der Eingang 0/1 nicht mehr ausgewertet.

0/1 = 1: Der Baustein wird nicht bearbeitet, d. h. der Baustein verändert seine Ausgänge nicht mehr. Dies gilt nicht während eines Löschvorganges.

SEG WORT

Am Eingang SEG wird die Nummer des Datensegments in der SmartMedia Card angegeben. In der SmartMedia Card stehen 250 Datensegmente zur Verfügung. Gültige Werte: 1...250.

RDY BINÄR

Am Ausgang RDY wird angezeigt, daß der Löschvorgang des Datensegmentes abgeschlossen ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang ERR betrachtet werden. Es gilt:

RDY = 1 und ERR = 0: Der Löschvorgang ist abgeschlossen. Das Datensegment wurde gelöscht.

RDY = 1 und ERR = 1: Beim Löschvorgang ist ein Fehler aufgetreten. Das Datensegment konnte nicht gelöscht werden.

ERR BINÄR

Am Ausgang ERR wird angezeigt, ob beim Löschvorgang ein Fehler aufgetreten ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang RDY betrachtet werden.

Kann das Datensegment nicht gelöscht werden, so gilt: RDY = 1 und ERR = 1.
Der Ausgang ERNO signalisiert die Fehlernummer.

ERNO WORT

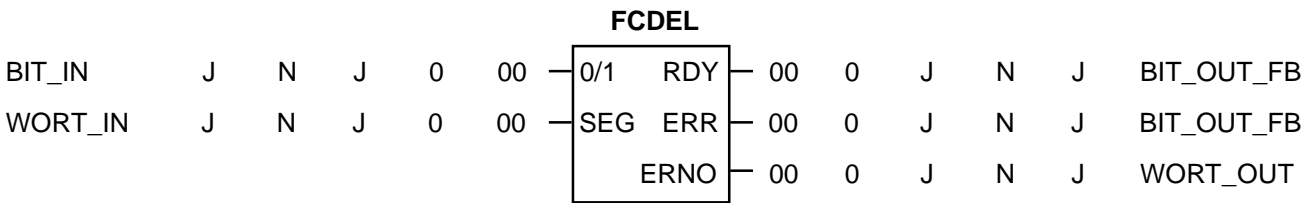
Am Ausgang ERNO wird eine Fehlernummer ausgegeben. Dieser Ausgang muß immer im Zusammenhang mit den Ausgängen RDY und ERR betrachtet werden. Es gilt:

- ERNO = 0: Es ist kein Fehler aufgetreten.
- ERNO = 1: Blocknummer und Anzahl der Blöcke ist größer als 63 (Bit 0 von ERNO = 1).
- ERNO = 2: Das angegebene Datensegment ist größer als 250 (Bit 1 von ERNO = 1).
- ERNO = 4: Block nicht programmierbar oder Checksum-Fehler (Bit 2 von ERNO = 1).
- ERNO = 8: Block enthält bereits Daten (Bit 3 von ERNO = 1).
- ERNO = 16: Sektor nicht löschtbar (Bit 4 von ERNO = 1).
- ERNO = 32: SmartMedia Card nicht initialisiert (Bit 5 von ERNO = 1).
- ERNO = 64: SmartMedia Card nicht gesteckt (Bit 6 von ERNO = 1).

Die Eingänge und Ausgänge sind nicht doppelbar und nicht invertierbar.

Grafik

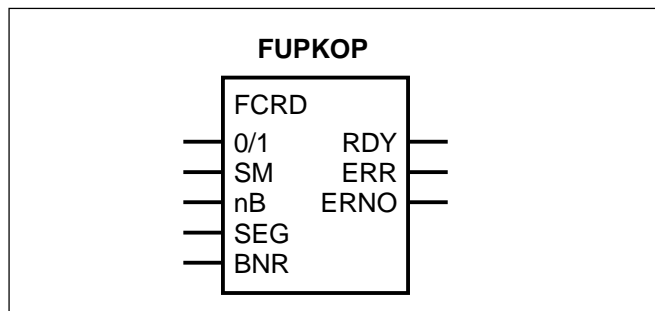
Plaus-gruppe	Pf1	Inv	Anz	Art	Nr	Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	----	-----	-----	-----	-----	--------------



Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	FCDEL		
	00001	PP 00	0/1	
	00002	PP 00	SEG	
	00003	PP 00	RDY	
	00004	PP 00	ERR	
	00005	PP 00	ERNO	
	00006			

Der Funktionsbaustein liest einen Datensatz von einem Datensegment der SmartMedia Card und legt diesen Datensatz ab dem projektierten Anfangsmerker SM ab.



Parameter

0/1	BINÄR	E, K, M, S	Sichern eines Datensatzes mit 0/1-Flanke
SM	BINÄR	A, E, K, M, S	
	WORT	AW, EW, KW, MW	
	DOPPELWORT	KD, MD	Anfangsmerker des Datensatzes
nB	WORT	EW, KW, MW	Anzahl der Blöcke des Datensatzes; gültige Werte: 1...128
SEG	WORT	EW, KW, MW	Nummer des Datensegments, gültige Werte: 1...250
BNR	WORT	EW, KW, MW	Nummer des Blocks im Datensegment; gültige Werte: 0...127
RDY	BINÄR	A, M	Lesevorgang beendet
ERR	BINÄR	A, M	Fehler aufgetreten
ERNO	WORT	AW, MW	Fehlernummer

VE-Daten

Aktualisierung der Ausgänge:	nein
Anzahl der Vergangenheitswerte:	2 Worte

Verfügbar ab: 07 KT 94 Index e

Beschreibung

Der Funktionsbaustein liest einen Datensatz von einem Datensegment der SmartMedia Card und legt diesen Datensatz ab dem projektierten Anfangsmerker SM ab. Die Daten des Datensatzes wurden durch den Funktionsbaustein FCWR oder durch den Bedienbefehl FCWR in der SmartMedia Card abgelegt.

Wichtiger Hinweis:

Der Zugriff auf die SmartMedia Card ist nur mit den Funktionsbausteinen FCWR und FCRD oder über die Programmierschnittstelle mit Hilfe der Bedien- und Testfunktionen möglich (Initialisierung der SmartMedia Card und Bedienfunktionen siehe AC31 Systembeschreibung Griff 7.3, Kapitel 2.4.1, Kommandos FCINIT, FCWR, FCRD, FCDEL).

Die Eingänge SM und nB legen fest, welche Daten aus der SmartMedia Card gelesen werden. Der Eingang SEG legt das Datensegment der SmartMedia Card fest. Die Anzahl der Daten, die von einem Block gelesen werden, ist abhängig vom Eingang SM.

Es gilt: Pro Block werden 64 Binär-Biten oder 32 Wort-Daten gelesen. Die Daten jedes Blocks sind mit einer Prüfsumme gesichert.

Der Lesevorgang eines Datensatzes wird durch eine 0/1-Flanke am Eingang 0/1 einmalig durchgeführt. Ist beim Lesen der Daten kein Fehler aufgetreten, werden die Ausgänge RDY und ERNO auf 0 gesetzt. Der Datensatz wird ab dem projektierten Anfangsmerker SM abgelegt. Die Ablage des Datensatzes kann mehrere SPS-Zyklen dauern.

Tritt beim Lesen des Datensatzes ein Fehler auf, wird RDY = 1 und ERR = 1 gesetzt. Die Art des Fehlers wird am Ausgang ERNO signalisiert.

Durch ein 0-Signal am Eingang 0/1 werden die Ausgänge RDY, ERR und ERNO auf 0 gesetzt.

0/1 BINÄR

Über den Eingang 0/1 wird die Bearbeitung des Bausteines gesteuert. Es gilt:

0/1 = 0: Die Ausgänge RDY, ERR und ERNO werden auf den Wert 0 gesetzt.

0/1 = 0/1-Flanke: Es wird der einmalige Lesevorgang des Datensatzes durchgeführt.

0/1 = 1: Der Baustein wird nicht bearbeitet, d. h. der Baustein verändert seine Ausgänge nicht mehr.

SM BINÄR/WORT/DOPPELWORT

Am Eingang SM wird der erste Binär/Wort-/Doppelwort-Merker für die Ablage des Datensatzes angegeben.

nB WORT

Am Eingang nB wird die Anzahl der Blöcke des Datensatzes angegeben. Die Anzahl der Daten, die von einem Block gelesen werden, ist abhängig vom Eingang SM. Pro Block werden 64 Binärdaten oder 32 Wort-Daten oder 16 Doppelwort-Daten gelesen.

Gültige Werte: 1...128

Beispiele:

- SM = M 01,00 und nB = 1: Ablage der Daten von M 01,00 bis M 04,15
(1 Block = 64 Binär-Daten)
- SM = M 01,00 und nB = 2: Ablage der Daten von M 01,00 bis M 08,15
(2 Blöcke = 128 Binär-Daten)
- SM = MW 02,00 und nB = 1: Ablage der Daten von MW 02,00 bis Mw 03,15
(1 Block = 32 Wort-Daten)
- SM = MW 02,00 und nB = 2: Ablage der Daten von MW 02,00 bis MW 05,15
(2 Blöcke = 64 Wort-Daten)
- SM = MD 03,00 und nB = 1: Ablage der Daten von MD 03,00 bis MD 03,15
(1 Block = 16 Doppelwort-Daten)
- SM = MD 03,00 und nB = 2: Ablage der Daten von MD 03,00 bis MD 04,15
(2 Blöcke = 32 Doppelwort-Daten)

SEG WORT

Am Eingang SEG wird die Nummer des Datensegments in der SmartMedia Card angegeben. In der SmartMedia Card stehen 250 Datensegmente zur Verfügung.

Gültige Werte: 1...250.

BNR WORT

Am Eingang BNR wird die Nummer des Blocks im Daten-segment angegeben.

Gültige Werte: 0...127

RDY BINÄR

Am Ausgang RDY wird angezeigt, daß der Lesevorgang des Datensatzes abgeschlossen ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang ERR betrachtet werden. Es gilt:

RDY = 1 und ERR = 0: Der Lesevorgang ist abgeschlossen. Der Datensatz wurde ab dem projektierten Eingang SM abgelegt.

RDY = 1 und ERR = 1: Beim Lesevorgang ist ein Fehler aufgetreten. Der Ausgang ERNO signalisiert die Fehlernummer.

ERR BINÄR

Am Ausgang ERR wird angezeigt, ob beim Lesevorgang ein Fehler aufgetreten ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang RDY betrachtet werden.

Ist ein Fehler aufgetreten, so gilt:

RDY = 1 und ERR = 1.

Der Ausgang ERNO signalisiert die Fehlernummer.

ERNO WORT

Am Ausgang ERNO wird eine Fehlernummer ausgegeben. Dieser Ausgang muß immer im Zusammenhang mit den Ausgängen RDY und ERR betrachtet werden. Die Fehlernummer ist binärcodiert. Es gilt:

ERNO = 0: Es ist kein Fehler aufgetreten.

ERNO = 1: Blocknummer und Anzahl der Blöcke ist größer als 63 (Bit 0 von ERNO = 1).

ERNO = 2: Das angegebene Datensegment ist größer als 250 (Bit 1 von ERNO = 1).

ERNO = 4: Block nicht programmierbar oder Checksum-Fehler (Bit 2 von ERNO = 1).

ERNO = 8: Block enthält bereits Daten (Bit 3 von ERNO = 1).

ERNO = 16: Sektor nicht löschar (Bit 4 von ERNO = 1).

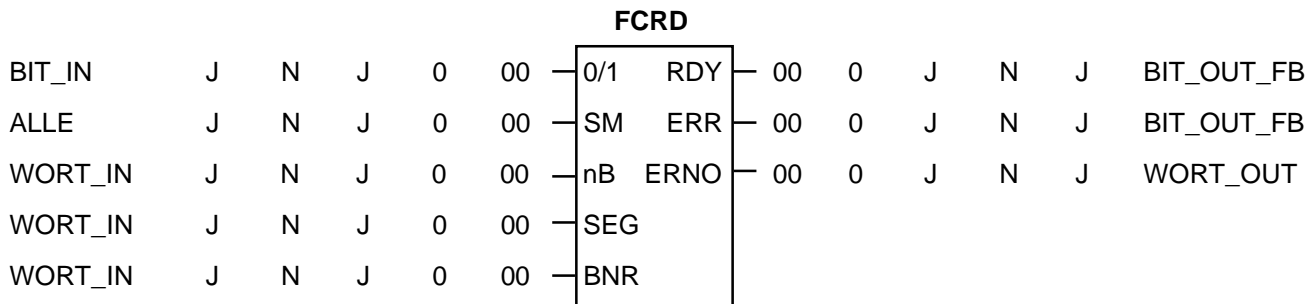
ERNO = 32: SmartMedia Card nicht initialisiert (Bit 5 von ERNO = 1).

ERNO = 64: SmartMedia Card nicht gesteckt (Bit 6 von ERNO = 1).

Die Eingänge und Ausgänge sind nicht doppelbar und nicht invertierbar.

Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr		Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	--	----	-----	-----	-----	-----	--------------

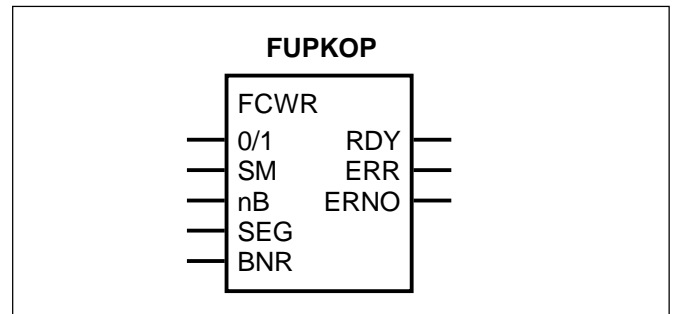


Source

SN	ZN	Befehl / Variable	Symbol	Langtext
----	----	-------------------	--------	----------

00000	00000	FCRD		
	00001	PP 00	0/1	
	00002	PP 00	SM	
	00003	PP 00	nB	
	00004	PP 00	SEG	
	00005	PP 00	BNR	
	00006	PP 00	RDY	
	00007	PP 00	ERR	
	00008	PP 00	ERNO	
	00009			

Der Funktionsbaustein schreibt einen Datensatz in ein Datensegment der SmartMedia Card.



Parameter

0/1	BINÄR	E, K, M, S	Sichern eines Datensatzes mit 0/1-Flanke
SM	BINÄR	A, E, K, M, S	
	WORT	AW, EW, KW, MW	
	DOPPELWORT	KD, MD	Anfangsmerker des Datensatzes
nB	WORT	EW, KW, MW	Anzahl der Blöcke des Datensatzes; gültige Werte: 1...128
SEG	WORT	EW, KW, MW	Nummer des Datensegments, gültige Werte: 1...250
BNR	WORT	EW, KW, MW	Nummer des Blocks im Datensegment; gültige Werte: 0...127
RDY	BINÄR	A, M	Schreibvorgang beendet
ERR	BINÄR	A, M	Fehler aufgetreten
ERNO	WORT	AW, MW	Fehlernummer

VE-Daten

Aktualisierung der Ausgänge:	nein
Anzahl der Vergangenheitswerte:	2 Worte

Verfügbar ab: 07 KT 94 Index e

Beschreibung

Der Funktionsbaustein schreibt einen Datensatz in ein Datensegment der SmartMedia Card. Dazu stehen in der SmartMedia Card 250 Datensegmente (1...250) zur Verfügung. Ein Löschvorgang (Funktionsbaustein FCDEL) löscht immer ein komplettes Datensegment. Jedes Datensegment besteht aus 128 Blöcken (0...127). Jeder Block besteht aus 32 Worten.

Nach einem Löschvorgang kann jeder dieser 128 Blöcke eines Datensegments nur einmal mit Daten beschrieben werden. Soll ein Block, der Daten enthält, mit neuen Daten überschrieben werden, muß das gesamte Datensegment vorher gelöscht werden. Dadurch gehen alle Daten in diesem Datensegment verloren.

Wichtiger Hinweis:

Der Zugriff auf die SmartMedia Card ist nur mit den Funktionsbausteinen FCWR und FCRD oder über die Programmierschnittstelle mit Hilfe der Bedien- und Testfunktionen möglich (Initialisierung der SmartMedia Card

und Bedienfunktionen siehe AC31 Systembeschreibung Griff 7.3, Kapitel 2.4.1, Kommandos FCINIT, FCWR, FCRD, FCDEL).

Die Eingänge SM und nB legen fest, welche Daten in der SmartMedia Card geschrieben werden. Der Eingang SEG legt das Datensegment in der SmartMedia Card fest. Die Anzahl der Daten, die ein Block aufnehmen kann, ist abhängig vom Eingang SM.

Es gilt: Pro Block werden 64 Binär-Baten oder 32 Wort-Daten oder 16 Doppelwort-Daten geschrieben. Die Daten jedes Blocks sind mit einer Prüfsumme gesichert.

Wird der Schreibvorgang eines Datensatzes gestartet, (0/1-Flanke am Eingang 0/1), dann dürfen die Daten des Datensatzes bis zur Beendigung des Schreibvorgangs nicht mehr verändert werden. Die Ablage des Datensatzes in der SmartMedia Card kann mehrere SPS-Zyklen dauern.

Mit einer 0/1-Flanke am Eingang 0/1 wird der einmalige Schreibvorgang des Datensatzes ausgelöst. Bis zur Beendigung der Ablage (RDY = 1) wird der Eingang 0/1 nicht mehr ausgewertet.

Nach Beendigung des Schreibvorgangs werden die Baustein-Ausgänge RDY, ERR und ERNO aktualisiert. Bei RDY = 1 und ERR = 0 war der Schreibvorgang erfolgreich. Sind die Ausgänge RDY = 1 und ERR = 1, so ist ein Fehler aufgetreten. Die Art des Fehlers signalisiert der Ausgang ERNO.

Nach der Ablage des Datensatzes in der SmartMedia Card werden durch ein 0-Signal am Eingang 0/1 die Baustein-Ausgänge RDY, ERR und ERNO auf 0 gesetzt. Eine erneute 0/1-Flanke am Eingang 0/1 startet einen neuen Schreibvorgang. Da ohne vorheriges Löschen des Datensatzes keine neuen Daten in Blöcke, die bereits Daten enthalten, geschrieben werden können, muß beim nächsten Schreibvorgang der Eingang BNR auf den nächsten freien Block zeigen.

0/1 BINÄR

Über den Eingang 0/1 wird die Bearbeitung des Bausteines gesteuert. Es gilt:

0/1 = 0: Die Ausgänge RDY, ERR und ERNO werden auf den Wert 0 gesetzt. Dies gilt nicht während eines Schreibvorganges.

0/1 = 0/1-Flanke: Es wird der einmalige Schreibvorgang des Datensatzes ausgelöst. Bis zur Beendigung des Schreibvorganges (RDY = 1) wird der Eingang nicht mehr ausgewertet.

0/1 = 1: Der Baustein wird nicht bearbeitet, d. h. der Baustein verändert seine Ausgänge nicht mehr. Dies gilt nicht während des Schreibvorganges.

SM BINÄR/WORT/DOPPELWORT

Am Eingang SM wird der erste Binär/Wort-/Doppelwort-Merker des Datensatzes angegeben. Wird der Schreibvorgang eines Datensatzes gestartet (0/1-Flanke am Eingang 0/1), dann dürfen die Daten des Datensatzes bis zur Beendigung des Schreibvorganges (RDY = 1) nicht mehr verändert werden.

nB WORT

Am Eingang nB wird die Anzahl der Blöcke des Datensatzes angegeben. Die Anzahl der Daten, die ein Block aufnehmen kann, ist abhängig vom Eingang SM. Pro Block werden 64 Binärdaten oder 32 Wort-Daten oder 16 Doppelwort-Daten abgelegt.
Gültige Werte: 1...128

Beispiele:

- SM = M 01,00 und nB = 1: Schreiben der Daten von M 01,00 bis M 04,15
(1 Block = 64 Binär-Daten)
- SM = M 01,00 und nB = 2: Schreiben der Daten von M 01,00 bis M 08,15
(2 Blöcke = 128 Binär-Daten)
- SM = MW 02,00 und nB = 1: Schreiben der Daten von MW 02,00 bis Mw 03,15
(1 Block = 32 Wort-Daten)
- SM = MW 02,00 und nB = 2: Schreiben der Daten von MW 02,00 bis MW 05,15
(2 Blöcke = 64 Wort-Daten)
- SM = MD 03,00 und nB = 1: Schreiben der Daten von MD 03,00 bis MD 03,15
(1 Block = 16 Doppelwort-Daten)
- SM = MD 03,00 und nB = 2: Schreiben der Daten von MD 03,00 bis MD 04,15
(2 Blöcke = 32 Doppelwort-Daten)

SEG WORT

Am Eingang SEG wird die Nummer des Datensatzes in der SmartMedia Card angegeben. In der SmartMedia Card stehen 250 Datensätze zur Verfügung.

Gültige Werte: 1...250.

BNR WORT

Am Eingang BNR wird die Nummer des Blocks im Datensatz angegeben.

Gültige Werte: 0...127

RDY BINÄR

Am Ausgang RDY wird angezeigt, daß der Schreibvorgang des Datensatzes abgeschlossen ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang ERR betrachtet werden. Es gilt:

RDY = 1 und ERR = 0: Der Schreibvorgang ist abgeschlossen. Der Datensatz wurde in der SmartMedia Card abgelegt.

RDY = 1 und ERR = 1: Beim Schreibvorgang ist ein Fehler aufgetreten. Der Ausgang ERNO signalisiert die Fehlernummer.

ERR BINÄR

Am Ausgang ERR wird angezeigt, ob beim Schreibvorgang ein Fehler aufgetreten ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang RDY betrachtet werden. Ist ein Fehler aufgetreten, so gilt:

RDY = 1 und ERR = 1.

Der Ausgang ERNO signalisiert die Fehlernummer.

ERNO WORT

Am Ausgang ERNO wird eine Fehlernummer ausgegeben. Dieser Ausgang muß immer im Zusammenhang mit den Ausgängen RDY und ERR betrachtet werden. Die Fehlernummer ist binärcodiert. Es gilt:

- ERNO = 0: Es ist kein Fehler aufgetreten.
- ERNO = 1: Blocknummer und Anzahl der Blöcke ist größer als 63 (Bit 0 von ERNO = 1).
- ERNO = 2: Das angegebene Datensegment ist größer als 250 (Bit 1 von ERNO = 1).
- ERNO = 4: Block nicht programmierbar oder Checksum-Fehler (Bit 2 von ERNO = 1).

ERNO = 8: Block enthält bereits Daten (Bit 3 von ERNO = 1).

ERNO = 16: Sektor nicht löscherbar (Bit 4 von ERNO = 1).

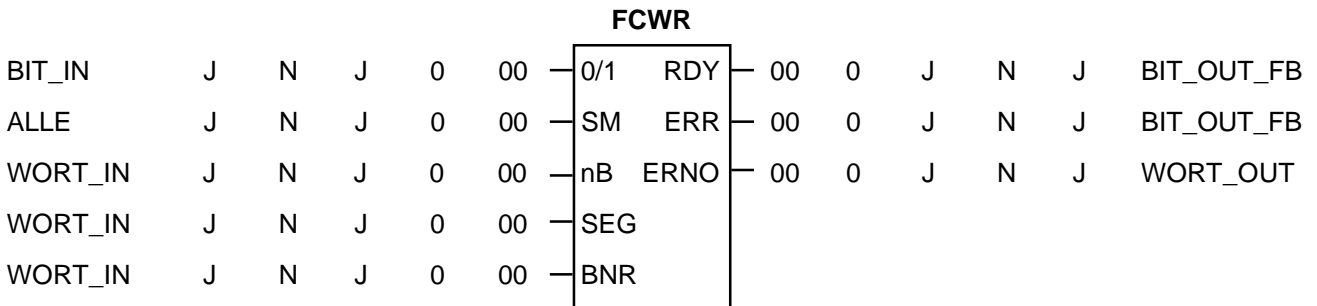
ERNO = 32: SmartMedia Card nicht initialisiert (Bit 5 von ERNO = 1).

ERNO = 64: SmartMedia Card nicht gesteckt (Bit 6 von ERNO = 1).

Die Eingänge und Ausgänge sind nicht doppelbar und nicht invertierbar.

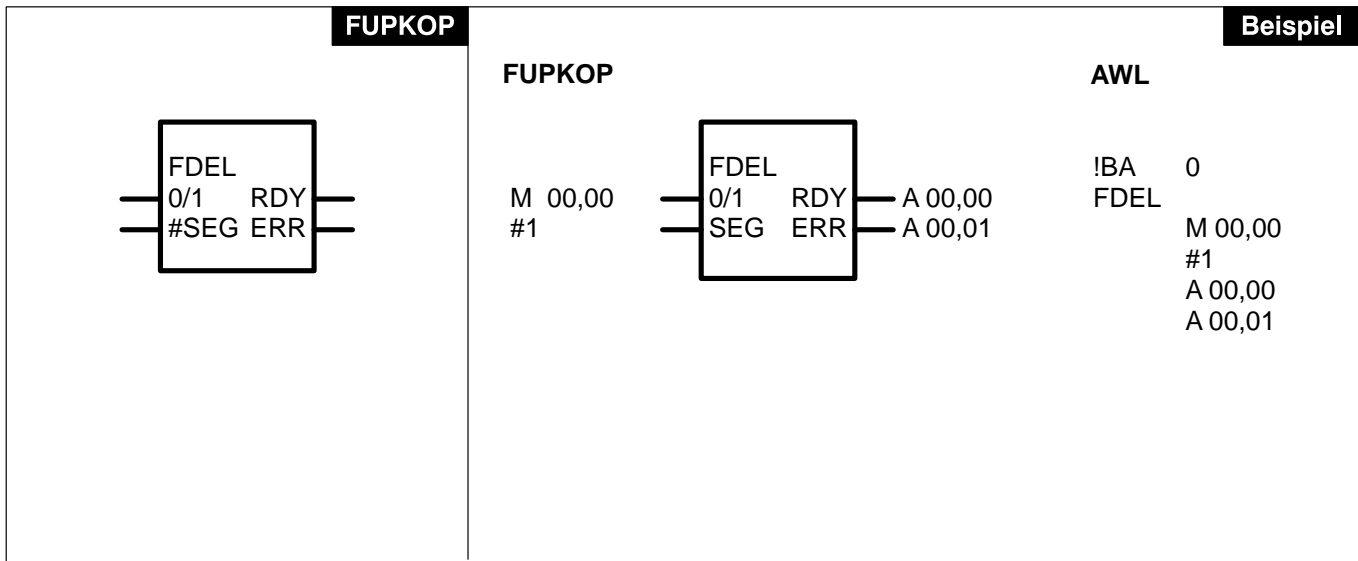
Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr	Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	----	-----	-----	-----	-----	--------------



Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	FCWR		
	00001	PP 00	0/1	
	00002	PP 00	SM	
	00003	PP 00	nB	
	00004	PP 00	SEG	
	00005	PP 00	BNR	
	00006	PP 00	RDY	
	00007	PP 00	ERR	
	00008	PP 00	ERNO	
	00009			



Parameter

0/1	BINÄR	E, M, A, K, S	Löschen <i>eines</i> Datensegments mit 0/1-Flanke
#SEG	DIREKTE KONSTANTE	#, #H	Nummer des Datensegments im FLASH-EPROM
RDY	BINÄR	A, M	0 ... 3
ERR	BINÄR	A, M	Schreibvorgang beendet
			Fehler aufgetreten, Datensegment nicht löschar

Beschreibung

Der Funktionsbaustein löscht ein Datensegment im FLASH-EPROM. Beim Löschvorgang gehen alle Daten in diesem Datensegment verloren.

Wichtiger Hinweis:

Der Zugriff auf das FLASH-EPROM ist nur mit den Funktionsbausteinen FWR und FRD erlaubt. Es ist **verboten**, mit anderen Funktionsbausteinen (WOL, WOS, COPY,...) auf das FLASH-EPROM zuzugreifen.

Der Eingang #SEG legt das Datensegment im FLASH-EPROM fest.

Das Löschen eines Datensegments im FLASH-EPROM kann mehrere SPS-Zyklen dauern.

Mit einer 0/1-Flanke am Eingang 0/1 wird der *einmalige* Löschvorgang des Datensegments ausgelöst. Bis zur Beendigung des Löschvorganges (RDY = 1) wird der Eingang 0/1 nicht mehr ausgewertet.

Nach Beendigung des Löschvorganges werden alle Baustein-Ausgänge aktualisiert. Bei RDY = 1 und ERR = 0 war das Löschen erfolgreich. Sind die Ausgänge RDY = 1 und ERR = 1, so konnte das Datensegment nicht gelöscht werden.

0/1 BINÄR

Über den Eingang 0/1 wird die Bearbeitung des Bausteines gesteuert.

0/1 = 0: Alle Ausgänge werden auf den Wert "0" gesetzt. Dies gilt nicht während eines Löschvorganges.

0/1 = 0/1-Flanke: Es wird der *einmalige* Löschvorgang des Datensegments ausgelöst. Bis zur Beendigung des Löschvorganges (RDY = 1) wird der Eingang 0/1 nicht mehr ausgewertet.

0/1 = 1: Der Baustein wird nicht bearbeitet, d. h. der Baustein verändert seine Ausgänge nicht mehr. Dies gilt nicht während eines Löschvorganges.

#SEG DIREKTE KONSTANTE

Am Eingang SEG wird die Nummer des Datensegments im FLASH-EPROM angegeben. Im FLASH-EPROM stehen 4 Datensegmente zur Verfügung.

Es gilt:

- #0 entspricht Datensegment 0
- #1 entspricht Datensegment 1
- #2 entspricht Datensegment 2
- #3 entspricht Datensegment 3

RDY BINÄR

Am Ausgang RDY wird angezeigt, daß der Löschvorgang des Datensegmentes abgeschlossen ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang ERR betrachtet werden.

Es gilt:

- RDY = 1 **und** ERR = 0: Der Löschvorgang ist abgeschlossen. Das Datensegment wurde gelöscht.
- RDY = 1 **und** ERR = 1: Beim Löschvorgang ist ein Fehler aufgetreten. Das Datensegment konnte nicht gelöscht werden.

ERR BINÄR

Am Ausgang ERR wird angezeigt, ob beim Löschvorgang ein Fehler aufgetreten ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang RDY betrachtet werden.

Kann das Datensegment nicht gelöscht werden, so gilt:
RDY = 1 **und** ERR = 1

Die Eingänge und Ausgänge sind nicht doppelbar und nicht invertierbar.

VE-Daten

Laufzeit:

Grundlaufzeit:

Zusätzliche Laufzeit:

Aktualisierung der Ausgänge:

Anzahl der Vergangenheitswerte:

Verfügbar ab:

07 KR 91 / 07 KT 92 / 07 KT 93 / 07 KP 62

23 – 53 µs

—

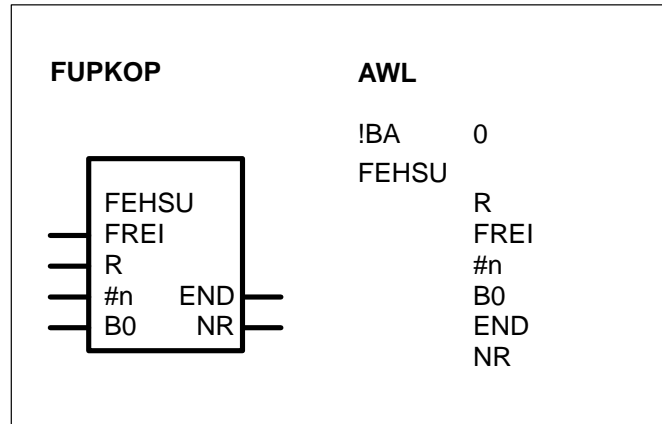
nein

2 Worte

ABB Procontic CS31 07KT92 R202/R262, 07KT93 R171

ABB Procontic T200 07KP62 R202

Der Funktionsbaustein durchsucht eine Liste von Binär-Variablen (E, A, M, S) nach gesetzten Binär-Variablen. Wird eine gesetzte Binär-Variablen gefunden, so wird ihre laufende Nummer, bezogen auf die Liste, angezeigt. Die gesetzte Binär-Variablen am Eingang wird gelöscht.



Parameter

FREI	BINÄR	E, A, M, S, K	Freigabe des Suchlaufes
R	BINÄR	E, A, M, S, K	Reset
#n	DIREKTE KONSTANTE	#, #H	Anzahl Binär-Variablen
B0	BINÄR	E, A, M	Liste der Binär-Variablen, Eingang ist doppelbar
END	BINÄR	A, M	Listenende erreicht
NR	WORT	AW, MW	Listen-Nummer der gefundenen Variablen

VE-Daten

Laufzeit:

Grundlaufzeit: 90 µs

Zusätzliche Laufzeit: 10 µs pro Binär-Variablen an den Eingängen B0...B_{n-1}

Aktualisierung der Ausgänge: ja, falls FREI = 1

Empfehlung: Ausgang direkt an Merker zuweisen oder Zwischenmerker global vergeben; siehe Griff 3, Kapitel 2.5.3

Anzahl der Vergangheitswerte: 4 Worte

Verfügbar ab: ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein durchsucht eine Liste von Binär-Variablen (E, A, M, S) nach gesetzten Binär-Variablen. Wird eine gesetzte Binär-Variablen gefunden, so wird ihre laufende Nummer, bezogen auf die Liste, angezeigt. Die gesetzte Binär-Variablen am Eingang wird gelöscht. Die Ein- und Ausgänge sind nicht negierbar/invertierbar. Der Eingang B0 ist doppelbar.

FREI BINÄR

Mit dem Eingang FREI wird der Suchlauf nach einer gesetzten Binär-Variablen freigegeben.

FREI = 0 Suchlauf ist nicht freigegeben.

FREI = 1 Suchlauf ist freigegeben.

Suchlauf:

Der Suchlauf wird solange fortgesetzt, bis

– eine gesetzte Binär-Variablen gefunden wird, oder

– das Ende der Variablen-Liste erreicht ist.

Wird in der Liste eine Binär-Variablen mit dem Wert 1 gefunden, so wird der Suchlauf beendet, die Variablen gelöscht, und die Listennummer der Variablen am Ausgang NR ausgegeben. Bei erneutem Aufruf des Bausteins (im nächsten Programm-Zyklus) wird der Suchlauf wieder fortgesetzt. Untersucht wird nun als erstes die Binär-Variablen, die direkt auf die zuvor gefundene folgt.

Wird keine gesetzte Binär-Variablen mehr gefunden, so endet der Suchlauf am Ende der Liste.

Dem Ausgang END wird dann der Wert 1 zugewiesen, und am Ausgang NR wird die Nummer der beim Suchlauf zuletzt gefundenen Variablen angezeigt.

Wird bei einem Suchlauf vom Listenanfang an keine gesetzte Variablen gefunden, so wird beim nächsten Aufruf des Bausteins automatisch wieder vom Listenanfang an gesucht. Dies erfolgt solange, bis eine gesetzte Variablen gefunden wird.

R BINÄR

Mit dem Eingang R wird der Funktionsbaustein in einen definierten Zustand (Reset) gebracht, um einen Suchlauf vom Anfang der Variablenliste starten zu können.

- R = 0 → kein Reset des Bausteins
- R = 1 → Reset des Bausteins zur Vorbereitung eines Suchlaufs vom Anfang der Liste an

Hinweis: Auch beim allerersten Aufruf des Bausteins nach dem Start des SPS-Programms *muß* zunächst ein Reset durchgeführt werden, um die korrekten Randbedingungen für den ersten Suchlauf zu schaffen.

Ein Reset des Bausteins bewirkt:

- Ausgang END = 0
- Ausgang NR = 0
- der nächste Suchvorgang erfolgt vom Listenanfang an

Priorität:

Der Eingang R hat höhere Priorität als der Eingang FREI, d. h. solange am Eingang R ein 1-Signal anliegt, wird kein Suchvorgang durchgeführt.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der Binär-Variablen angegeben, die an den Eingängen B0 ... B_{n-1} projiziert sind. Die Angabe erfolgt als direkte Konstante.

B0..B_{n-1} BINÄR

Der Eingang B0 ist doppelbar (B0 ... B_{n-1}). An den Eingängen B0 ... B_{n-1} wird die Liste der zu untersuchenden Binär-Variablen angegeben. Nicht zulässig sind dabei wegen des Rücksetzmechanismus Schrittoperanden (S) und indirekte Konstanten (K).

END BINÄR

Am Ausgang END wird signalisiert, ob beim Suchlauf das Ende der vorgegebenen Variablenliste erreicht wurde, ohne daß beim gerade durchgeführten Suchvorgang eine gesetzte Binär-Variable gefunden wurde.

D. h. wird bei einem Suchvorgang die letzte Binär-Variable der Liste als "gesetzt" erkannt und gemeldet, so wird der Ausgang END noch nicht auf den Wert 1 gesetzt, sondern dies geschieht dann erst im nächsten Bearbeitungszyklus.

- END = 0 → Listenende nicht erreicht
- END = 1 → Listenende erreicht, ohne daß beim aktuellen Suchlauf eine gesetzte Binär-Variable gefunden wurde

NR WORT

Am Ausgang NR wird die Listen-Nummer der zuletzt gefundenen Binär-Variablen ausgegeben. Es gilt die Zuordnung:

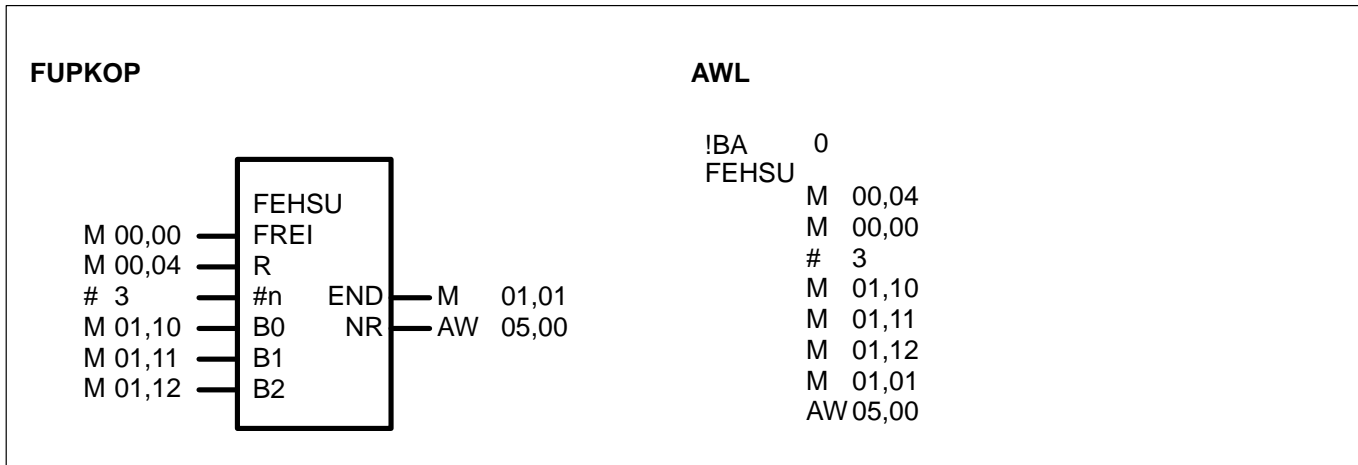
Variable am Eingang	Listennummer
B0	1
B1	2
.	.
.	.
.	.
B _{n-1}	n

Wird beim Suchvorgang das Ende der Liste erreicht, ohne daß eine gesetzte Variable gefunden wurde, so wird am Ausgang NR weiterhin die Nummer der zuletzt gefundenen Variablen ausgegeben.

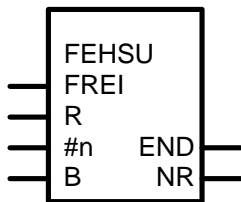
Suchen vom Listenanfang an, ohne daß eine Variable gefunden wird:

Wird bei einem Suchlauf vom Anfang der Liste keine gesetzte Binär-Variable gefunden, so wird beim Erreichen des Listenendes dem Ausgang END der Zustand 1 und dem Ausgang NR der Wert 0 zugewiesen. Beim erneuertem Aufruf des Bausteins im nächsten Programmzyklus erfolgt der Suchvorgang automatisch wieder vom Anfang der Liste an. Das automatische Suchen vom Anfang der Liste an geschieht solange, bis einmal eine gesetzte Binär-Variable gefunden wird.

Beispiel



VE-FUP-Definition

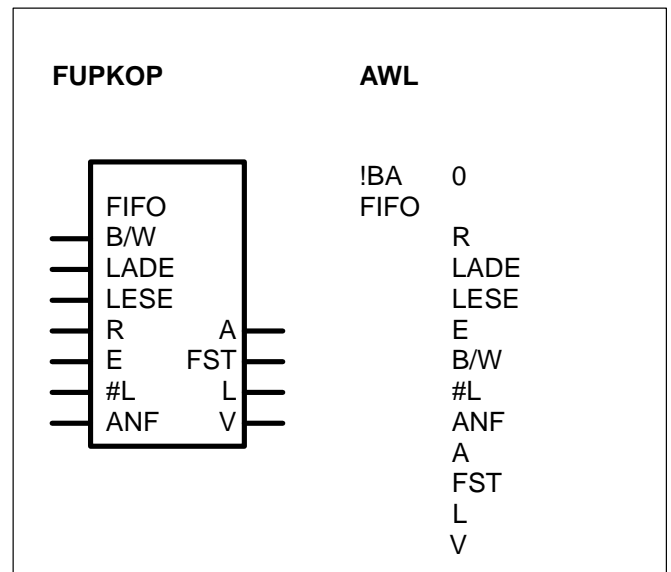


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
R	E	L	N	P	J	0	0
#n	K	W	N	P	J	0	0
B	E	L	N	P	J	1	0
END	A	L	N	P	J	0	0
NR	A	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr (vorbelegt 0)
00001	FEHSU			
00002		PP 0	R	Eingang BINÄR (Reset)
00003		PP 0	FREI	Eingang BINÄR (Freigabe)
00004		PP 0	#n	# KONSTANTE (Anzahl Bit)
	[1		
00005		PP 0	B	Eingang BINÄR (Element Bitfeld)
]	1		
00006		PP 0	END	Ausgang BINÄR (Feldende erreicht)
00007		PP 0	NR	Ausgang WORT (Nummer)

Der Funktionsbaustein realisiert einen Stapelspeicher für Binär- oder Wortdaten, aus dem die zuerst eingeschriebenen Daten als erstes wieder ausgelesen werden (first in/first out).



Parameter

B/W	BINÄR	E, A, M, S, K	Binärdaten/Wortdaten
LADE	BINÄR	E, A, M, S, K	Lade FIFO
LESE	BINÄR	E, A, M, S, K	Lese FIFO
R	BINÄR	E, A, M, S, K	Reset FIFO, 0→1-Flanke
E	BINÄR, WORT	E, A, M, S, K EW, AW, MW, KW	Eingabe der Daten in das FIFO
#L	DIREKTE KONSTANTE	#, #H	Anzahl Speicherplätze (Byte)
ANF	BINÄR WORT	E, A, M, K EW, AW, MW, KW	Anfang des FIFO im Merkerbereich
A	BINÄR, WORT	E, M AW, MW	Ausgabe der Daten aus dem FIFO
FST	WORT	AW, MW	Füllstand des FIFO
L	BINÄR	A, M	FIFO leer
V	BINÄR	A, M	FIFO voll

VE-Daten

Laufzeit:	
Grundlaufzeit:	64 µs
Zusätzliche Laufzeit:	0 ... 114 µs, je nach durchzuführender Funktion
Aktualisierung der Ausgänge:	nein; Empfehlung: Ausgang direkt an Merker zuweisen oder Zwischenmerker global vergeben; siehe Griff 3, Kapitel 2.5.3
Anzahl der Vergangenheitswerte:	4 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein realisiert einen Stapelspeicher für Binär- oder Wortdaten, aus dem die zuerst eingeschriebenen Daten als erstes wieder ausgelesen werden (first in/first out).

Die Eingänge und Ausgänge sind weder doppelbar noch negierbar/invertierbar.

B/W BINÄR

Der Eingang B/W dient dazu Binär- oder Wort-Verarbeitung vorzuzählen (FIFO für Binärdaten oder FIFO für Wortdaten):

B/W = 0 → Wort-Verarbeitung

B/W = 1 → Binär-Verarbeitung

Die Binär- oder Wortverarbeitung wird einmal festgelegt und darf während des Betriebs des FIFO's nicht geändert werden.

LADE BINÄR

Mit einem 1-Signal am Eingang LADE wird der am Eingang E anstehende Wert in den nächsten freien Platz des FIFO's übernommen.

Ist das FIFO "voll" und es liegt ein "Lade"-Signal an, so wird der neue Wert *nicht* eingelesen. Erst, wenn wieder ein Wert ausgelesen wurde, kann ein neuer Wert eingelesen werden. Dieser wird dann in den ersten Platz des FIFO's übernommen.

LESE BINÄR

Ein 1-Signal am Eingang LESE bewirkt eine Ausgabe des aktuellen FIFO-Wertes am Ausgang A. Ist das FIFO "leer", so wird ein anstehender Lese-Auftrag ignoriert und am Ausgang A der Wert 0 ausgegeben. Der Ausgang A wird stets auf 0 gesetzt, wenn kein Lese-Auftrag vorliegt.

LADE und LESE

Liegt gleichzeitig ein Lade- und Lese-Auftrag vor, so wird der zu ladende Wert direkt vom Eingang E an den Ausgang A durchgegeben, falls das FIFO leer ist.

Ist das FIFO nicht leer, so erfolgt die Ausgabe des aktuellen FIFO-Wertes und die neue Ladeinformation wird in das FIFO übernommen.

Das FIFO ändert in dieser Betriebsart seinen Füllstand nicht. Am Ausgang FST bleibt der ausgegebene Wert konstant. Ist das FIFO leer, so bleibt der Ausgang L auf 1 und der Ausgang FST wird auf 0 gesetzt.

R BINÄR

Eine 0→1-Flanke am Eingang R bewirkt ein Rücksetzen (Reset) des Bausteins. Zuvor eingelesene Werte stehen somit nicht mehr zur Verfügung.

Der Ausgang L nimmt den Wert 1, die Ausgänge FST und A nehmen den Wert 0 an.

R und LADE

Steht gleichzeitig ein R- und ein LADE-Signal an, so wird zuerst das Reset und direkt anschließend das Laden ausgeführt. Das FIFO wird also zurückgesetzt und der neu einzulesende Wert wird dann wieder sofort als erster Wert im FIFO abgespeichert.

R und LADE und LESE

Der Wert am Eingang E wird direkt an den Ausgang A gegeben. Der Ausgang L wird permanent auf 1 und der Ausgang FST permanent auf 0 gesetzt.

E BINÄR/WORT

Am Eingang E wird der Wert vorgegeben, der in das FIFO zu übernehmen ist.

#L DIREKTE KONSTANTE

Am Eingang A wird die Anzahl der erforderlichen Speicherplätze (Bytes) des FIFO's vorgegeben. Die Anzahl wird als direkte Konstante angegeben und ergibt sich aus folgender Formel:

BINÄR-Daten:

#L = Anzahl der abzulegenden Binär-Werte

WORT-Daten:

#L = 2 * Anzahl der abzulegenden Wortwerte

Z. B.: BINÄR-Daten: 3 Werte → #L = 3

WORT-Daten: 3 Werte → #L = 6

Eine Plausibilitätsüberwachung überprüft die Angabe der FIFO-Länge auf den Wert 0 und bei Wortverarbeitung auch auf ungerade Byteangabe. Ist die Angabe am Eingang #L unkorrekt, so geht das FIFO in den Initialzustand (wie nach R).

ANF BINÄR/WORT

Am Eingang ANF wird die FIFO-Speicheranfangsadresse als Binär- oder Wort-Merker vorgegeben. Das FIFO beginnt beim angegebenen Merker.

A BINÄR/WORT

Beim Lesen des FIFO's steht am Ausgang A der aktuelle Wert zur Verfügung. Liegt *kein* Lese-Auftrag vor, so wird der Wert 0 ausgegeben.

FST WORT

Der Ausgang FST zeigt zu jedem Zeitpunkt den aktuellen Füllstand des FIFO's an. Der Füllstand ist die Anzahl der im FIFO gespeicherten Binär- oder Wort-Werte.

L BINÄR

Der Ausgang L zeigt an, ob das FIFO leer ist.

L = 0 → FIFO ist nicht leer

L = 1 → FIFO ist leer

V BINÄR

Der Ausgang V zeigt an, ob das FIFO voll ist.

V = 0 → FIFO ist nicht voll

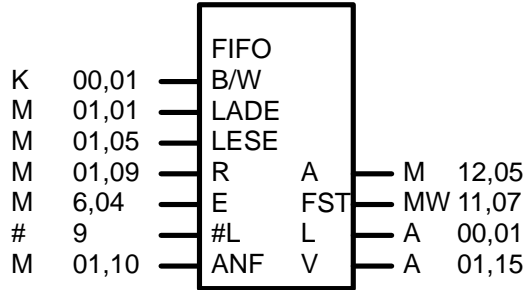
V = 1 → FIFO ist voll

Ist das FIFO voll, so kann kein Wert mehr eingelesen werden. Erst, wenn ein Wert ausgelesen wurde, ist wieder ein Einlesen möglich. Dies geschieht dann wieder ab dem Anfang des FIFO's.

Beispiel

FIFO für 9 Binärdaten, abgelegt ab M 01,10

FUPKOP



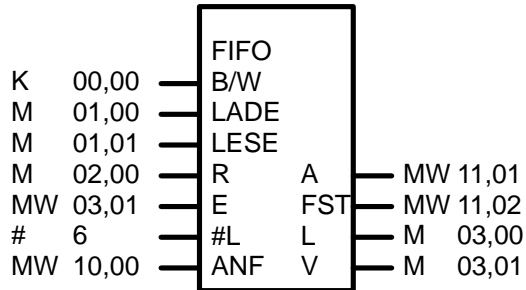
AWL

```

!BA      0
FIFO
M  01,09
M  01,01
M  01,05
M  06,04
K  00,01
#  9
M  01,10
M  12,05
MW 11,07
A  00,01
A  01,15
    
```

FIFO für 3 Wortdaten, abgelegt ab MW 10,00

FUPKOP

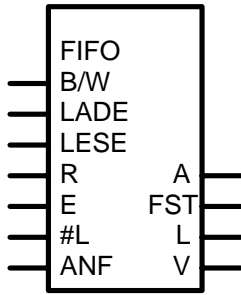


AWL

```

!BA      0
FIFO
M  02,00
M  01,00
M  01,01
MW 03,01
K  00,00
#  6
MW 10,00
MW 11,01
MW 11,02
M  03,00
M  03,01
    
```

VE-FUP-Definition

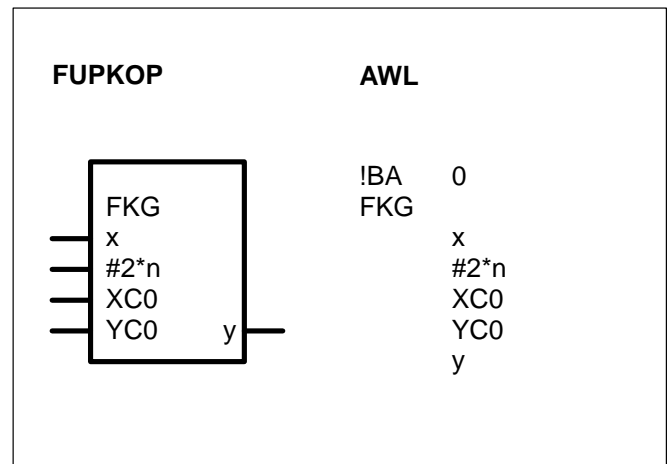


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
B/W	E	L	N	P	J	0	0
LADE	E	L	N	P	J	0	0
LESE	E	L	N	P	J	0	0
R	E	L	N	P	J	0	0
E	E	X	N	P	J	0	0
#L	K	W	N	P	J	0	0
ANF	E	X	N	P	J	0	0
A	A	X	N	P	J	0	0
FST	A	W	N	P	J	0	0
L	A	L	N	P	J	0	0
V	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	FIFO			
00002		PP 0	R	Eingang BINÄR (Reset)
00003		PP 0	LADE	Eingang BINÄR (Lade FIFO)
00004		PP 0	LESE	Eingang BINÄR (Lese FIFO)
00005		PP 0	E	Eingang BINÄR/WORT (Eingabe)
00006		PP 0	B/W	Eingang BINÄR (Binär- Wortdaten)
00007		PP 0	#L	# KONSTANTE (Anzahl Speicherplätze)
00008		PP 0	ANF	Eingang BINÄR/WORT (Anfang Speicherplätze)
00009		PP 0	A	Ausgang BINÄR/WORT (Ausgabe)
00010		PP 0	FST	Ausgang WORT (Füllstand)
00011		PP 0	L	Ausgang BINÄR (FIFO Leer)
00012		PP 0	V	Ausgang BINÄR (FIFO voll)

In einem x/y-Koordinationsystem wird durch n Koordinaten-Punkte $X_0/Y_0...X_{n-1}/Y_{n-1}$ ein Polygonzug festgelegt. Für jeden Wert am Eingang x gibt der Funktionsbaustein am Ausgang y den zugeordneten y-Wert des Polygonzuges aus.



Parameter

x	WORT	EW, AW, MW, KW	Eingang für den x-Wert des Polygonzuges
#2*n	DIREKTE KONSTANTE	#, #H	n: Anzahl der Stützpunkte
XC0	WORT	EW, AW, MW, KW	Eingang für x-Werte der Stützpunkte; Eingang doppelbar
YC0	WORT	EW, AW, MW, KW	Eingang für y-Werte der Stützpunkte; Eingang doppelbar
y	WORT	AW, MW	Ausgang für den y-Wert des Polygonzuges

VE-Daten

Laufzeit:	
Grundlaufzeit:	136 µs
Zusätzliche Laufzeit:	9 µs pro Stützpunkt
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

In einem x/y-Koordinationsystem wird durch n Koordinaten-Punkte $X_0/Y_0...X_{n-1}/Y_{n-1}$ ein Polygonzug festgelegt. Für jeden Wert am Eingang x gibt der Funktionsbaustein am Ausgang y den zugeordneten y-Wert des Polygonzuges aus.

Für die Interpolation zwischen zwei Stützpunkten gilt:

$$y = \frac{(x - X_{i-1}) * (Y_i - Y_{i-1})}{X_i - X_{i-1}} + Y_{i-1}$$

Für die x-Koordinaten gilt:

$$X_0 < X_1 < X_2 \dots < X_{n-1}$$

$$2 \leq n \leq 32767$$

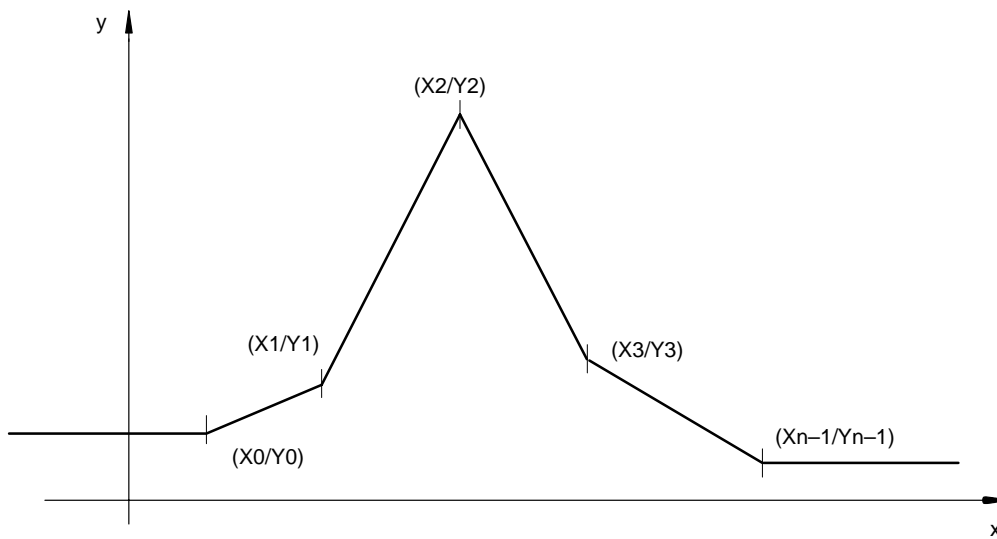
Anmerkung: Bei der Division wird stets abgerundet, d. h. ein Rest bei der Division wird nicht berücksichtigt.

Der Funktionsbaustein interpoliert linear zwischen den Stützpunkten. Der so erhaltene Polygonzug stellt den Zusammenhang zwischen der Eingangsgröße x und der Ausgangsgröße y dar.

Für den Bereich außerhalb der Stützpunkte gilt:

$$\text{für } x \leq X_0 \quad \text{ist } y = Y_0$$

$$\text{für } x \geq X_{n-1} \quad \text{ist } y = Y_{n-1}$$

**x** WORT

Am Eingang x wird die laufende x -Koordinate vorgegeben. Der Baustein bestimmt dann die durch den Polygonzug zugeordnete y -Koordinate.

#2*n DIREKTE KONSTANTE

Am Eingang $\#2*n$ wird die Anzahl der x/y -Koordinaten vorgegeben, die zur Festlegung des Polygonzuges erforderlich sind. Die Angabe erfolgt als direkte Konstante.

- n = Anzahl der Stützpunkte
- $2*n$ = Anzahl der x/y -Koordinaten für n Stützpunkte

XC0...XC_{n-1} WORT

An den Eingängen $XC_0 \dots XC_{n-1}$ werden die x -Koordinaten der n Stützpunkte angegeben. Der Eingang XC_0 ist doppelbar. Bei einer Dopplung dieses Eingangs wird der Eingang YC_0 automatisch mitgedoppelt.

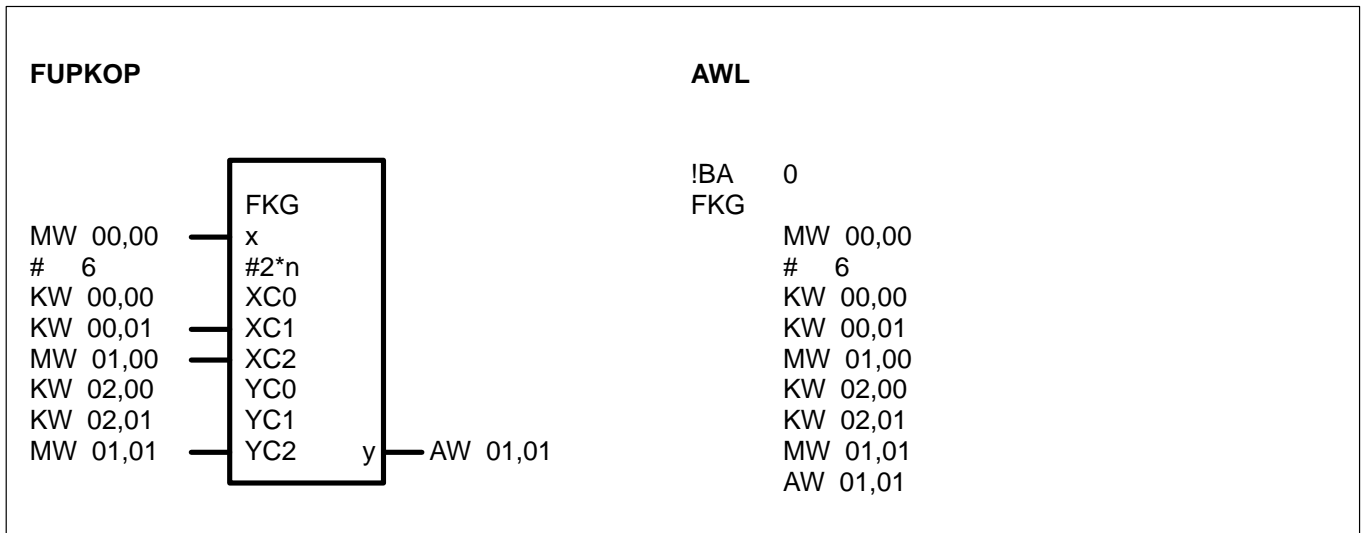
YC0...YC_{n-1} WORT

An den Eingängen $YC_0 \dots YC_{n-1}$ werden die y -Koordinaten der n Stützpunkte angegeben. Der Eingang YC_0 ist doppelbar. Bei einer Dopplung dieses Eingangs wird der Eingang XC_0 automatisch mitgedoppelt.

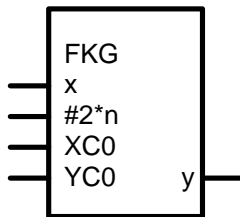
y WORT

Am Ausgang y wird die y -Koordinate ausgegeben, die durch den Polygonzug der vorgegebenen x -Koordinate zugeordnet ist.

Beispiel



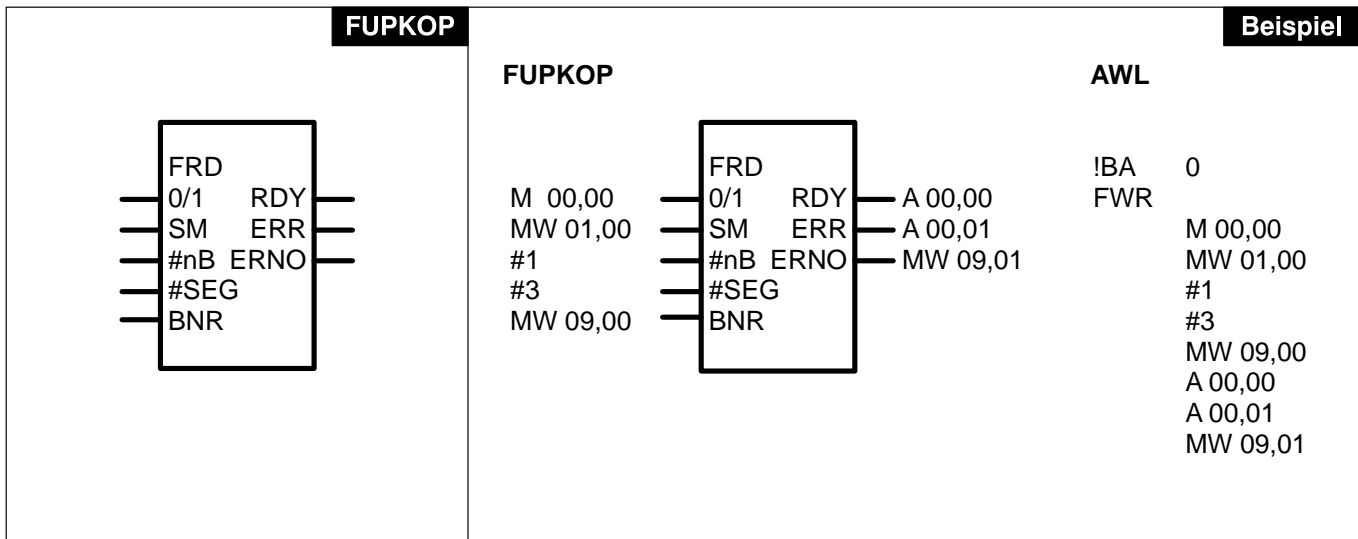
VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
x	E	W	N	P	J	0	0
#2*n	K	W	N	P	J	0	0
XC0	E	W	N	P	J	1	0
YC0	E	W	N	P	J	1	0
y	A	W	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	FKG			
00002		PP 0	x	Eingang WORT
00003		PP 0	#2*n	Eingang # KONSTANTE(n=Anzahl Punkte)
00004	[1		
00005		PP 1	XC0	Eingang WORT(x-Koordinate)
00006		PP 1	YC0	Eingang WORT(y-Koordinate)
00007]	1		
00008		PP 0	y	Ausgang WORT



Parameter

0/1	BINÄR	E, M, A, K, S	Lesen <i>eines</i> Datensatzes mit 0/1-Flanke
SM	BINÄR	E, M, A, K, S	
	WORT	EW, MW, AW, KW	
	DOPPELWORT	MD, KD	Anfangsmerker der Ablage des Datensatzes
#nB	DIREKTE	#, #H	Anzahl der Blöcke des Datensatzes
	KONSTANTE		1 ... 481
#SEG	DIREKTE	#, #H	Nummer des Datensegments im FLASH-EPROM
	KONSTANTE		0 ... 3
BNR	WORT	EW, MW, AW, KW	Nummer des Blocks im Datensegment
			gültige Werte: 0 ... 480
RDY	BINÄR	A, M	Lesevorgang beendet
ERR	BINÄR	A, M	Fehler aufgetreten
ERNO	WORT	AW, MW	Fehlernummer

Beschreibung

Der Funktionsbaustein liest einen Datensatz von einem Datensegment im FLASH-EPROM und legt diesen Datensatz ab dem projektierten Anfangsmerker SM ab. Die Daten des Datensatzes wurden durch den Funktionsbaustein FWR oder durch den Bedienbefehl FWR im FLASH-EPROM abgelegt.

Wichtiger Hinweis:

Der Zugriff auf das FLASH-EPROM ist nur mit den Funktionsbausteinen FWR und FRD erlaubt. Es ist **verboten**, mit anderen Funktionsbausteinen (WOL, WOS, COPY,...) auf das FLASH-EPROM zuzugreifen.

Die Eingänge SM und #nB legen fest, welche Daten aus dem FLASH-EPROM gelesen werden. Der Eingang #SEG legt das Datensegment im FLASH-EPROM fest. Die Anzahl der Daten, die von einem Block gelesen werden, ist abhängig vom Eingang SM.

Es gilt:

Pro Block werden 32 Binär-Daten oder 16 Wort-Daten oder 8 Doppelwort-Daten gelesen. Die Daten jedes Blocks sind mit einer Prüfsumme gesichert.

Der Lesevorgang eines Datensatzes wird durch eine 0/1-Flanke am Eingang 0/1 einmalig durchgeführt. Ist beim Lesen der Daten kein Fehler aufgetreten, werden die Ausgänge RDY zu "1" und ERR und ERNO zu Null gesetzt. Der Datensatz wird ab dem projektierten Anfangsmerker SM abgelegt.

Die Ablage des Datensatzes kann mehrere SPS-Zyklen dauern.

Tritt beim Lesen des Datensatzes ein Fehler auf, wird RDY = 1 und ERR = 1 gesetzt. Die Art des Fehlers wird am Ausgang ERNO signalisiert.

Durch ein Null-Signal am Eingang 0/1 werden die Ausgänge RDY, ERR und ERNO zu Null gesetzt.

0/1 BINÄR

Über den Eingang 0/1 wird die Bearbeitung des Bausteines gesteuert.

0/1 = 0: Die Ausgänge RDY, ERR und ERNO werden auf den Wert "0" gesetzt.

0/1 = 0/1-Flanke: Es wird der *einmalige* Lesevorgang des Datensatzes durchgeführt.

0/1 = 1: Der Baustein wird nicht bearbeitet, d. h. der Baustein verändert seine Ausgänge nicht mehr.

SM BINÄR/WORT/DOPPELWORT

Am Eingang SM wird der 1. Binär/Wort/Doppelwort-Merker für die Ablage des Datensatzes angegeben.

#nB DIREKTE KONSTANTE

Am Eingang #nB wird die Anzahl der Blöcke des Datensatzes angegeben. Die Anzahl der Daten, die von einem Block gelesen werden, ist abhängig vom Eingang SM. Pro Block werden 32 Binär-Daten oder 16 Wort-Daten oder 8 Doppelwort-Daten gelesen.

Beispiele:

– SM = M 01,00 und #nB = 1:

Ablage der Daten von M 01,00 bis M 02,15
(1 Block = 32 Binär-Daten)

– SM = M 01,00 und #nB = 2:

Ablage der Daten von M 01,00 bis M 04,15
(2 Blöcke = 64 Binär-Daten)

– SM = MW 02,00 und #nB = 1:

Ablage der Daten von MW 02,00 bis MW 02,15
(1 Block = 16 Wort-Daten)

– SM = MW 02,00 und #nB = 2:

Ablage der Daten von MW 02,00 bis MW 03,15
(2 Blöcke = 32 Wort-Daten)

– SM = MD 03,00 und #nB = 1:

Ablage der Daten von MD 03,00 bis MD 03,07
(1 Block = 8 Doppelwort-Daten)

– SM = MD 03,00 und #nB = 2:

Ablage der Daten von MD 03,00 bis MD 03,15
(2 Blöcke = 16 Doppelwort-Daten)

#SEG DIREKTE KONSTANTE

Am Eingang SEG wird die Nummer des Datensegments im FLASH-EPROM angegeben. Im FLASH-EPROM stehen 4 Datensegmente zur Verfügung. Es gilt:

– #0 entspricht Datensegment 0

– #1 entspricht Datensegment 1

– #2 entspricht Datensegment 2

– #3 entspricht Datensegment 3

BNR WORT

Am Eingang BNR wird die Nummer des Blocks im Daten-segment angegeben.

Gültige Werte: 0...480

RDY BINÄR

Am Ausgang RDY wird angezeigt, daß der Lesevorgang des Datensatzes abgeschlossen ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang ERR betrachtet werden.

Es gilt:

RDY = 1 **und** ERR = 0: Der Lesevorgang ist abgeschlossen. Der Datensatz wurde ab dem projektierten Eingang SM abgelegt.

RDY = 1 **und** ERR = 1: Beim Lesevorgang ist ein Fehler aufgetreten. Der Ausgang ERNO signalisiert die Fehlernummer.

ERR BINÄR

Am Ausgang ERR wird angezeigt, ob beim Lesevorgang ein Fehler aufgetreten ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang RDY betrachtet werden.

Ist ein Fehler aufgetreten, so gilt:

RDY = 1 **und** ERR = 1

Der Ausgang ERNO signalisiert die Fehlernummer.

ERNO WORT

Am Ausgang ERNO wird eine Fehlernummer ausgegeben. Dieser Ausgang muß immer im Zusammenhang mit den Ausgängen RDY und ERR betrachtet werden. Die Fehlernummer ist binärcodiert.

Es gilt:

ERNO = 0: kein Fehler aufgetreten

ERNO = 1: Blocknummer und Anzahl der Blöcke ist größer als 480
(Bit 0 von ERNO = 1).

ERNO = 2: Datensegment ist größer als 3
(Bit 1 von ERNO = 1).

ERNO = 4: Prüfsummenfehler der gelesenen Daten. Die Daten werden nicht in den Merkerbereich eingetragen
(Bit 2 von ERNO = 1).

Die Eingänge und Ausgänge sind nicht doppelbar und nicht invertierbar.

VE-Daten

Laufzeit:

Grundlaufzeit:

Zusätzliche Laufzeit:

Aktualisierung der Ausgänge:

Anzahl der Vergangenheitswerte:

Verfügbar ab:

07 KR 91 / 07 KT 92 / 07 KT 93 / 07 KP 62

25 – 67 µs

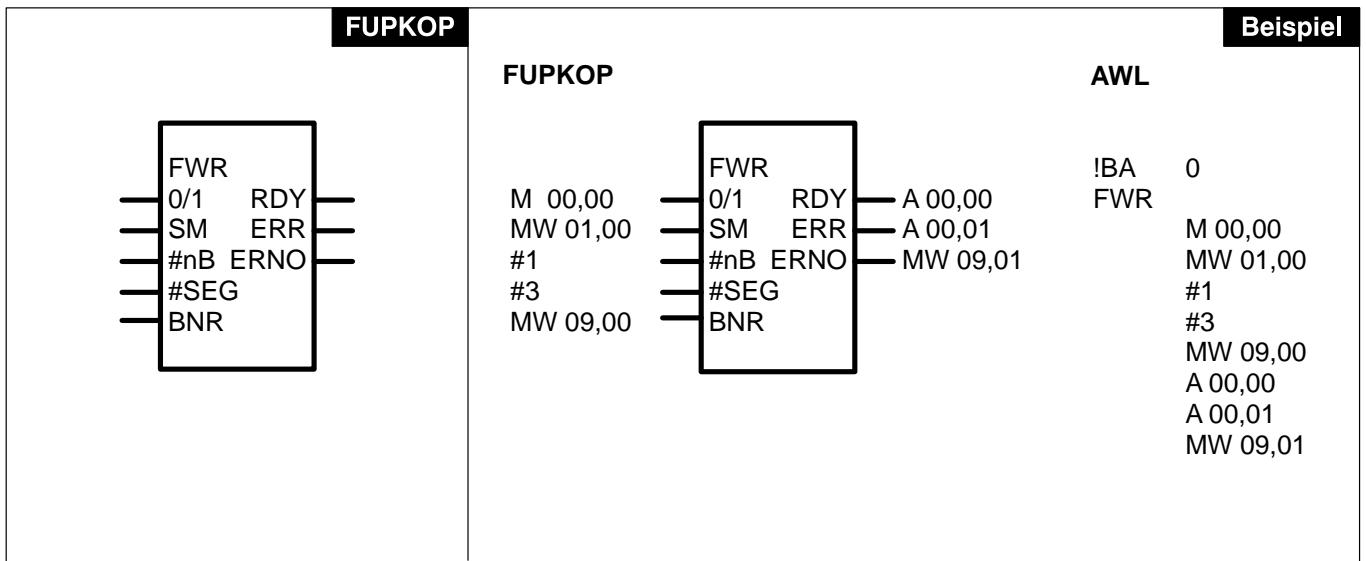
—

nein

2 Worte

ABB Procontic CS31 07KT92 R202/R262,07KT93 R171

ABB Procontic T200 07KP62 R202



Parameter

0/1	BINÄR	E, M, A, K, S	Sichern <i>eines</i> Datensatzes mit 0/1-Flanke
SM	BINÄR	E, M, A, K, S	
	WORT	EW, MW, AW, KW	
#nB	DOPPELWORT	MD, KD	Anfangsmerker des Datensatzes
	DIREKTE	#, #H	Anzahl der Blöcke des Datensatzes
	KONSTANTE		1...481
#SEG	DIREKTE	#, #H	Nummer des Datensegments im FLASH-EPROM
	KONSTANTE		0...3
BNR	WORT	EW, MW, AW, KW	Nummer des Blocks im Datensegment
			gültige Werte: 0...480
RDY	BINÄR	A, M	Schreibvorgang beendet
ERR	BINÄR	A, M	Fehler aufgetreten
ERNO	WORT	AW, MW	Fehlernummer

Beschreibung

Der Funktionsbaustein schreibt einen Datensatz in ein Datensegment im FLASH-EPROM. Dazu stehen im FLASH-EPROM 4 Datensegmente (0...3) zur Verfügung. Ein Löschvorgang (Funktionsbaustein FDEL) löscht immer ein komplettes Datensegment. Jedes Datensegment besteht aus 481 Blöcken (0...480). Nach einem Löschvorgang kann jeder dieser 481 Blöcke eines Datensegments nur **einmal** Daten aufnehmen. Soll ein Block, der Daten enthält, mit neuen Daten überschrieben werden, muß das gesamte Datensegment vorher gelöscht werden. Dadurch gehen alle Daten in diesem Datensegment verloren.

Wichtiger Hinweis:

Der Zugriff auf das FLASH-EPROM ist nur mit den Funktionsbausteinen FWR und FRD erlaubt. Es ist **verboten**, mit anderen Funktionsbausteinen (WOL, WOS, COPY,...) auf das FLASH-EPROM zuzugreifen.

Für jeden Anwender-Datensatz ist im SPS-Programm ein eigener Funktionsbaustein FWR und ein eigener Funktionsbaustein FRD zu projektieren.

Die Eingänge SM und #nB legen fest, welche Daten in das FLASH-EPROM geschrieben werden. Der Eingang #SEG legt das Datensegment im FLASH-EPROM fest. Die Anzahl der Daten, die ein Block aufnehmen kann, ist abhängig vom Eingang SM.

Es gilt:

Pro Block werden 32 Binär-Daten oder 16 Wort-Daten oder 8 Doppelwort-Daten geschrieben. Die Daten jedes Blocks werden mit einer Prüfsumme gesichert.

Wird der Schreibvorgang eines Datensatzes gestartet (0/1-Flanke am Eingang 0/1), dann dürfen die Daten des Datensatzes bis zur Beendigung des Schreibvorgangs (RDY = 1) nicht mehr verändert werden.

Die Ablage des Datensatzes im FLASH-EPROM kann mehrere SPS-Zyklen dauern.

Mit einer 0/1-Flanke am Eingang 0/1 wird der *einmalige* Schreibvorgang des Datensatzes ausgelöst. Bis zur Beendigung der Ablage (RDY = 1) wird der Eingang 0/1 nicht mehr ausgewertet.

Nach Beendigung des Schreibvorgangs werden die Baustein-Ausgänge RDY, ERR und ERNO aktualisiert. Bei RDY = 1 und ERR = 0 war die Sicherung erfolgreich. Sind die Ausgänge RDY = 1 und ERR = 1, ist ein Fehler aufgetreten. Die Art des Fehlers signalisiert der Ausgang ERNO.

Nach der Ablage eines Datensatzes im FLASH-EPROM werden durch ein "0"-Signal am Eingang 0/1 die Baustein-Ausgänge RDY, ERR und ERNO zu Null gesetzt. Eine erneute 0/1-Flanke am Eingang 0/1 startet einen neuen Schreibvorgang. Da ohne vorheriges Löschen des Datensegments keine neuen Daten in Blöcke, die bereits Daten enthalten, geschrieben werden können, muß beim nächsten Schreibvorgang der Eingang BNR auf den nächsten freien Block zeigen.

0/1 BINÄR

Über den Eingang 0/1 wird die Bearbeitung des Bausteines gesteuert.

0/1 = 0: Die Ausgänge RDY, ERR und ERNO werden auf den Wert "0" gesetzt. Dies gilt nicht während eines Schreibvorganges.

0/1 = 0/1-Flanke: Es wird der *einmalige* Schreibvorgang des Datensatzes ausgelöst. Bis zur Beendigung des Schreibvorganges (RDY = 1) wird der Eingang 0/1 nicht mehr ausgewertet.

0/1 = 1: Der Baustein wird nicht bearbeitet, d. h. der Baustein verändert seine Ausgänge nicht mehr. Dies gilt nicht während eines Schreibvorganges.

SM BINÄR/WORT/DOPPELWORT

Am Eingang SM wird der 1. Binär/Wort/Doppelwort-Merker des Datensatzes angegeben. Wird der Schreibvorgang eines Datensatzes gestartet (0/1-Flanke am Eingang 0/1), dann dürfen die Daten des Datensatzes bis zur Beendigung des Schreibvorganges (RDY = 1) nicht mehr verändert werden.

#nB DIREKTE KONSTANTE

Am Eingang #nB wird die Anzahl der Blöcke des Datensatzes angegeben. Die Anzahl der Daten, die ein Block aufnehmen kann, ist abhängig vom Eingang SM. Pro Block werden 32 Binär-Daten oder 16 Wort-Daten oder 8 Doppelwort-Daten abgelegt.

Beispiele:

- SM = M 01,00 und #nB = 1:
Schreiben der Daten von M 01,00 bis M 02,15
(1 Block = 32 Binär-Daten)
- SM = M 01,00 und #nB = 2:
Schreiben der Daten von M 01,00 bis M 04,15
(2 Blöcke = 64 Binär-Daten)
- SM = MW 02,00 und #nB = 1:
Schreiben der Daten von MW 02,00 bis MW 02,15
(1 Block = 16 Wort-Daten)
- SM = MW 02,00 und #nB = 2:
Schreiben der Daten von MW 02,00 bis MW 03,15
(2 Blöcke = 32 Wort-Daten)
- SM = MD 03,00 und #nB = 1:
Schreiben der Daten von MD 03,00 bis MD 03,07
(1 Block = 8 Doppelwort-Daten)
- SM = MD 03,00 und #nB = 2:
Schreiben der Daten von MD 03,00 bis MD 03,15
(2 Blöcke = 16 Doppelwort-Daten)

#SEG DIREKTE KONSTANTE

Am Eingang SEG wird die Nummer des Datensegments im FLASH-EPROM angegeben. Im FLASH-EPROM stehen 4 Datensegmente zur Verfügung.

Es gilt:

- #0 entspricht Datensegment 0
- #1 entspricht Datensegment 1
- #2 entspricht Datensegment 2
- #3 entspricht Datensegment 3

BNR WORT

Am Eingang BNR wird die Nummer des Blocks im Datensegment angegeben.

Gültige Werte: 0...480

RDY BINÄR

Am Ausgang RDY wird angezeigt, daß der Schreibvorgang des Datensatzes abgeschlossen ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang ERR betrachtet werden.

Es gilt:

RDY = 1 **und** ERR = 0: Der Schreibvorgang ist abgeschlossen. Der Datensatz wurde im FLASH-EPROM abgelegt.

RDY = 1 **und** ERR = 1: Beim Schreibvorgang ist ein Fehler aufgetreten. Der Ausgang ERNO signalisiert die Fehlernummer.

ERR BINÄR

Am Ausgang ERR wird angezeigt, ob beim Schreibvorgang ein Fehler aufgetreten ist. Der Ausgang muß immer im Zusammenhang mit dem Ausgang RDY betrachtet werden.

Ist ein Fehler aufgetreten, so gilt:

RDY = 1 **und** ERR = 1

Der Ausgang ERNO signalisiert die Fehlernummer.

ERNO WORT

Am Ausgang ERNO wird eine Fehlernummer ausgegeben. Dieser Ausgang muß immer im Zusammenhang mit den Ausgängen RDY und ERR betrachtet werden.

Die Fehlernummer ist binärcodiert.

Es gilt:

- ERNO = 0: kein Fehler aufgetreten
 ERNO = 1: Blocknummer und Anzahl der Blöcke ist größer als 480 (Bit 0 von ERNO = 1).
 ERNO = 2: Datensegment ist größer als 3 (Bit 1 von ERNO = 1).
 ERNO = 4: FLASH-EPROM nicht programmierbar (Bit 2 von ERNO = 1).
 ERNO = 8: Block enthält bereits Daten (Bit 3 von ERNO = 1).

Die Eingänge und Ausgänge sind nicht doppelbar und nicht invertierbar.

VE-Daten

Laufzeit:

Grundlaufzeit:

Zusätzliche Laufzeit:

Aktualisierung der Ausgänge:

Anzahl der Vergangenheitswerte:

Verfügbar ab:

07 KR 91 / 07 KT 92 / 07 KT 93 / 07 KP 62

22 – 67 µs

—

nein

2 Worte

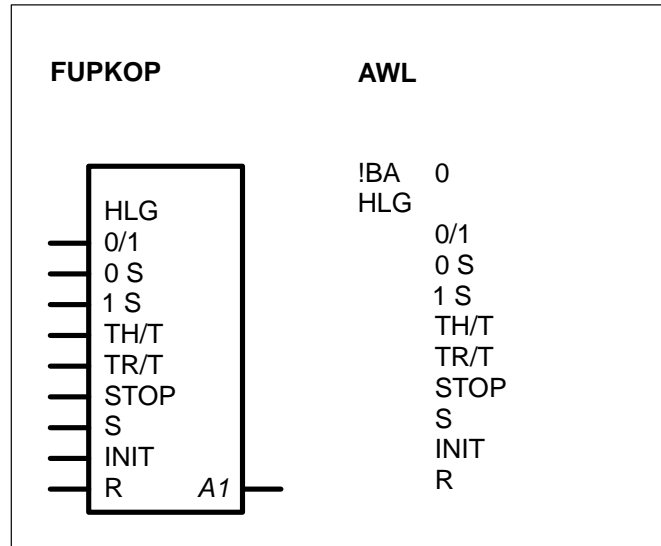
ABB Procontic CS31 07KT92 R202/R262, 07KT93 R171

ABB Procontic T200 07KP62 R202

Der Hochlaufgeber dient zur rampenförmigen Anpassung des momentanen Istwerts am Ausgang an einen vorgegebenen Sollwert.

Der Wert am Ausgang des HLG wird vom momentanen Istwert aus linear mit der Steigung y' an den vorgegebenen Sollwert angepaßt.

Der Wert am Ausgang durchläuft dabei in der Zeit TH bzw. TR genau den Betrag des Sollwertes. Hat der Wert am Ausgang des HLG den Sollwert erreicht, so ändert er sich nicht mehr, es sei denn, es wird ein neuer Sollwert vorgegeben.



Parameter

0/1	BINÄR	E, A, M, S, K	Auswahl des Sollwertes 0 bzw. des Sollwertes 1
0 S	WORT	EW, AW, MW, KW	Sollwert 0
1 S	WORT	EW, AW, MW, KW	Sollwert 1
TH/T	WORT	EW, AW, MW, KW	Hochlaufzeit normiert auf die Zykluszeit
TR/T	WORT	EW, AW, MW, KW	Rücklaufzeit normiert auf die Zykluszeit
STOP	BINÄR	E, A, M, S, K	Festhalten des Ausgangs auf dem momentanen Wert
S	BINÄR	E, A, M, S, K	Setzen des Ausgangs auf den INIT-Wert
INIT	WORT	EW, AW, MW, KW	INIT-Wert, auf den der Ausgang gesetzt werden kann
R	BINÄR	E, A, M, S, K	Rücksetzen des Ausgangs auf den Wert 0
A1	WORT	MW, AW	Ausgang

VE-Daten

Laufzeit:	
Grundlaufzeit:	126 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	2 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Hochlaufgeber dient zur rampenförmigen Anpassung des momentanen Istwerts am Ausgang an einen vorgegebenen Sollwert.

Der Wert am Ausgang des HLG wird vom momentanen Istwert aus linear mit der Steigung y' an den vorgegebenen Sollwert angepaßt.

Der Wert am Ausgang durchläuft dabei in der Zeit TH bzw. TR genau den Betrag des Sollwertes. Hat der Wert am Ausgang des HLG den Sollwert erreicht, so ändert er sich nicht mehr, es sei denn, es wird ein neuer Sollwert vorgegeben.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar/invertierbar.

Die Steigung y' der Rampe ergibt sich aus der vorgegebenen Zeit TH (Hochlaufzeit) bzw. TR (Rücklaufzeit) und dem Betrag des Sollwertes zu:

$$\text{Steigung } y' = \frac{\text{Betrag Sollwert}}{\text{TH oder TR}}$$

Die Steigung ist

- positiv, wenn Sollwert > Istwert
- negativ, wenn Sollwert < Istwert
- 0, wenn Sollwert = 0

Der vorgegebene Sollwert hat also zwei Funktionen:

- sein Betrag bestimmt zusammen mit der vorgegebenen Zeit TH oder TR die Steigung der Rampe;
- er stellt den Wert dar, an den der momentane Istwert rampenförmig anzugleichen ist.

Der Anwender kann getrennt die Hochlaufzeit TH und Rücklaufzeit TR vorgeben. Aufgrund des Sollwertes wird die Richtung der Steigung bestimmt. Aus der Richtung der Steigung ergibt sich dann, ob die Laufzeit TH oder TR verwendet wird.

Steigung y' positiv → TH, d.h. Rampe läuft nach oben.
Steigung y' negativ → TR, d.h. Rampe läuft nach unten.

Die Hochlaufzeit TH und die Rücklaufzeit TR sind auf die Programmzykluszeit TZ zu normieren, d.h. an den entsprechenden Bausteineingängen ist vorzugeben:

- Hochlaufzeit: TH/TZ
- Rücklaufzeit: TR/TZ

Die Zeiten werden in Millisekunden angegeben. Für die Zeitkonstanten TH bzw. TR gilt:

$$0 \leq TH \leq 32767$$

$$0 \leq TR \leq 32767$$

Es können zwei Sollwerte projiziert werden (0 S und 1 S), wobei die Auswahl eines dieser Sollwerte durch den binären Eingang 0/1 (Sollwertauswahl) erfolgt.

Die Sollwerte können folgende Werte annehmen:
 $-32767 \leq \text{Sollwert} \leq +32767$

Der Ausgang des Hochlaufgebers kann jederzeit

- auf dem momentanen Wert festgehalten werden
- auf einen Initialwert gesetzt werden
- zurückgesetzt (Ausgang = 0) werden

Höchste Priorität hat dabei der Eingang STOP und die niedrigste Priorität hat der Eingang R.

Die Werte an den Eingängen des HLG können im Anwenderprogramm jederzeit geändert werden. Auf diese Weise kann aus der linearen Anpassung des Istwertes an den Sollwert eine beliebige (nichtlineare) Anpassung gemacht werden.

Achtung

Sollwert = 0 bedeutet, daß die Steigung der Rampe ebenfalls gleich 0 ist, d.h. der momentane Istwert ändert sich nicht. Soll von Istwert ungleich 0 der Istwert auf 0 gefahren werden, so muß ein Sollwert ungleich 0 vorgegeben werden und der Ausgang des Hochlaufgebers durch einen nachfolgenden Begrenzer auf 0 begrenzt werden. (Bei der Interpolation wird bei den Rundungsübergängen nur mit ganzen Zahlen gerechnet.)

0/1 BINÄR

Mit dem Eingang 0/1 wird einer der beiden Sollwerte ausgewählt.

0/1 = 0 → Sollwert 0 S

0/1 = 1 → Sollwert 1 S

0 S WORT

Am Eingang 0 S wird der Sollwert 0 vorgegeben.

1 S WORT

Am Eingang 1 S wird der Sollwert 1 vorgegeben.

TH/T WORT

Am Eingang TH/T wird die Hochlaufzeit vorgegeben. Die Hochlaufzeit TH muß dabei auf die Zykluszeit T normiert werden.

TR/T WORT

Am Eingang TH/T wird die Hochlaufzeit vorgegeben. Die Hochlaufzeit TH muß dabei auf die Zykluszeit T normiert werden.

STOP BINÄR

Mit dem Eingang STOP kann der Ausgang auf dem momentanen Wert festgehalten werden.

STOP = 0 → Ausgang nicht festgehalten

STOP = 1 → Ausgang wird festgehalten

Der Eingang STOP hat höhere Priorität als die Eingänge S und R.

S BINÄR

Mit dem Eingang S kann der Ausgang auf den am Eingang INIT angegebenen Initialwert gesetzt werden.

S = 0 → Ausgang wird nicht auf Initialwert gesetzt.

S = 1 → Ausgang wird auf Initialwert gesetzt.

INIT WORT

Am Eingang INIT wird der Initialwert vorgegeben, auf den der Ausgang bei Bedarf gesetzt werden soll.

R BINÄR

Mit dem Eingang R kann der Ausgang auf den Wert 0 gesetzt werden.

R = 0 → Ausgang wird nicht zurückgesetzt.

R = 1 → Ausgang wird auf den Wert 0 zurückgesetzt.

A1 WORT

Ausgang des Bausteins.

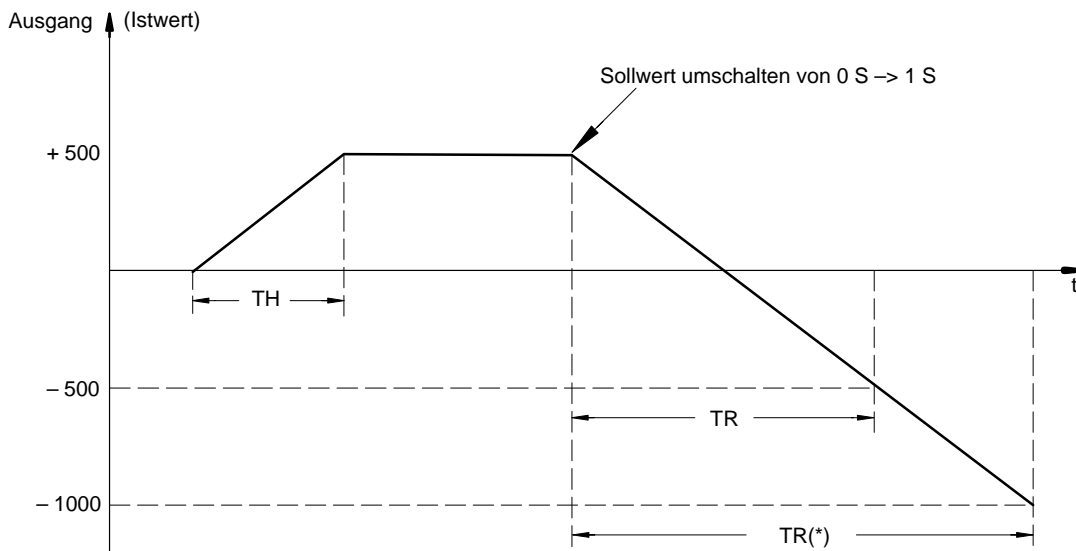
Beispiel 1

Es soll der Istwert von 0 auf den Sollwert +500 (Sollwertbetrag 500) und dann von +500 auf den Sollwert -1000 (Sollwertbetrag 1000) gefahren werden.

Die Hochlaufzeit ist TH, die Rücklaufzeit TR.

0 S: 500

1 S: -1000



TR(*) ist die eigentliche Zeit bis der Istwert -1000 erreicht hat.

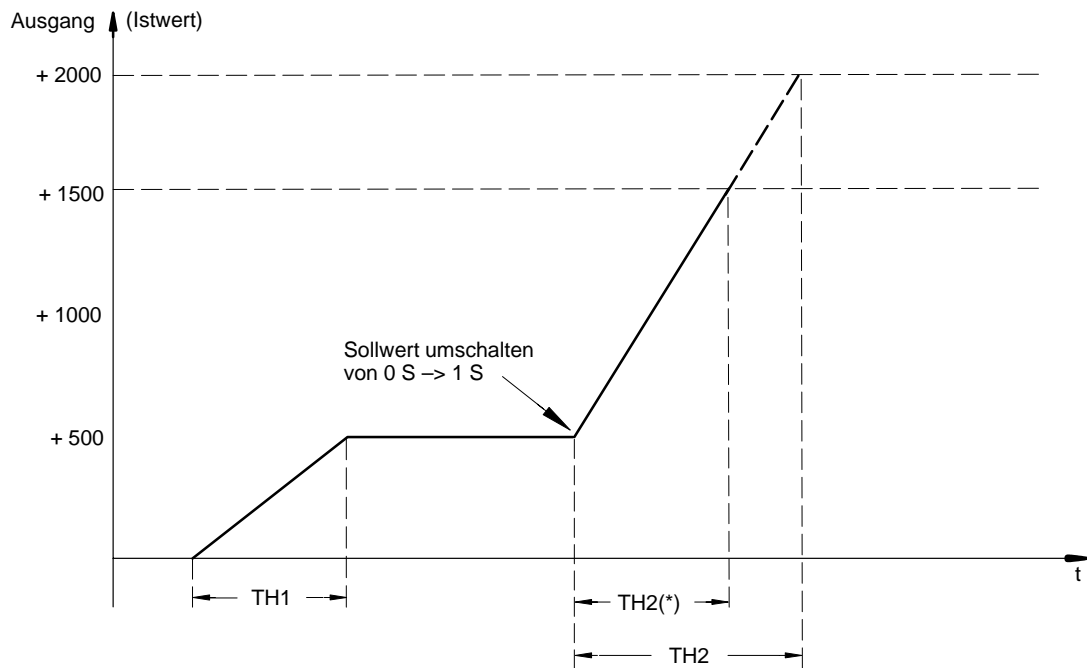
Der Istwert verändert sich in der Zeit TH bzw. TR um den Betrag des angelegten Sollwertes.

Beispiel 2

Es soll der Istwert von 0 auf den Sollwert +500 (Sollwertbetrag 500) und dann von +500 auf den Sollwert 1500 (Sollwertbetrag 1500) gefahren werden.

Die Hochlaufzeit ist TH, die Rücklaufzeit TR.

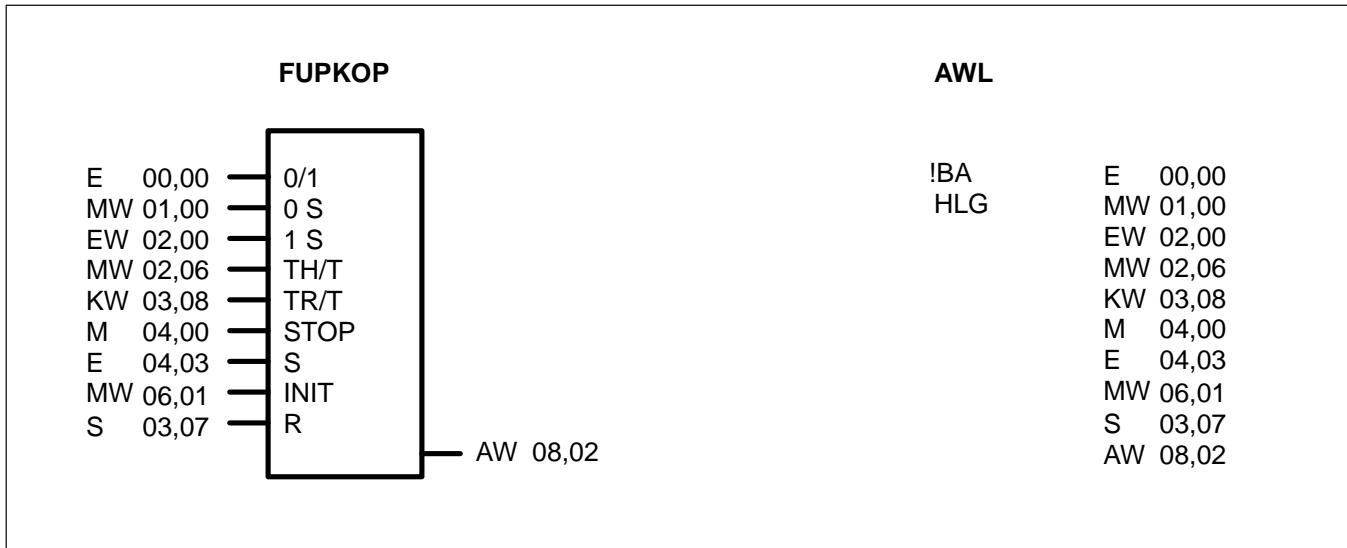
0 S: 500
1 S: 1500



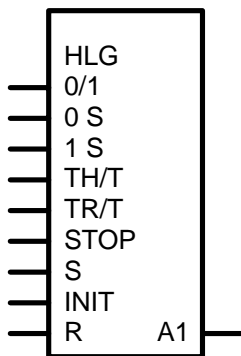
TR(*) ist die eigentliche Zeit bis der Istwert 1500 erreicht hat.

Der Istwert verändert sich in der Zeit TH bzw. TR um den Betrag des angelegten Sollwertes.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0/1	E	L	N	P	J	0	0
0 S	E	W	N	P	J	0	0
1 S	E	W	N	P	J	0	0
TH/T	E	W	N	P	J	0	0
TR/T	E	W	N	P	J	0	0
STOP	E	L	N	P	J	0	0
S	E	L	N	P	J	0	0
INIT	E	W	N	P	J	0	0
R	E	L	N	P	J	0	0
A1	A	W	N	P	N	0	0

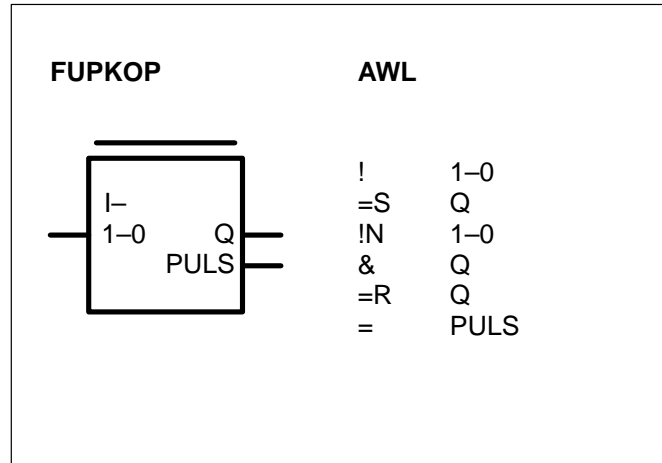
VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	HLG			
00002		PP 0	0/1	Eingang BIT (Sollwertauswahl)
00003		PP 0	0 S	Eingang WORT (Sollwert Nr. 0)
00004		PP 0	1 S	Eingang WORT (Sollwert Nr. 1)
00005		PP 0	TH/T	Eingang WORT (Hochlaufzeit)
00006		PP 0	TR/T	Eingang WORT (Rücklaufzeit)
00007		PP 0	STOP	Eingang BIT (Stop Hochlaufgeber)
00008		PP 0	S	Eingang BIT (Setze auf INIT)
00009		PP 0	INIT	Eingang WORT (Initialwert)
00010		PP 0	R	Eingang BIT (Rücksetze auf 0)
00011		PP 0	A1	Ausgang WORT

Eine negative Flanke (1–0) am Eingang 1–0 erzeugt am Ausgang PULS einen Impuls von der Dauer eines SPS-Programmzyklusses.

ACHTUNG:

Die Funktion dieses Verknüpfungselements entspricht teilweise nicht der Beschreibung. Die Funktionalität ist ab 907 PC 31 Vers. 2.2 geändert und entspricht nun der Beschreibung.

**Parameter**

1–0	BINÄR	E, M, A, S, K	Eingang für 1–0 Flanke
Q	BINÄR	A, M	Ausgang zur Abfrage des direkten Merkers
PULS	BINÄR	A, M	Impuls-Ausgang

VE-Daten**Serie 90, 07 KP 62****Serie 30, 40, 50**

Laufzeit:

Grundlaufzeit:

13,2 µs

45 µs

Zusätzliche Laufzeit:

—

—

Aktualisierung der Ausgänge:

ja

ja

Anzahl der Vergangenheitswerte:

keine

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331
ABB Procontic T200 / 07 KP 62907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51**Beschreibung**

Eine negative Flanke (1–0) am Eingang 1–0 erzeugt am Ausgang PULS einen Impuls von der Dauer eines SPS-Programmzyklusses.

ACHTUNG:

Die Funktion dieses Verknüpfungselements entspricht teilweise nicht der Beschreibung. Die Funktionalität ist ab 907 PC 31 Vers. 2.2 geändert und entspricht nun der Beschreibung.

Beschreibung des Fehlers bei 907 PC 31 Vers. 1.1 und älter:

Beim Einschalten des SPS-Programms (erstmaliges

Durchlaufen des Verknüpfungselements) wird am Ausgang PULS fälschlicherweise ein Impuls abgegeben.

1–0 BINÄR

Der Wert des Operanden an diesem Eingang wird auf eine 1–0-Flanke überwacht.

Q BINÄR

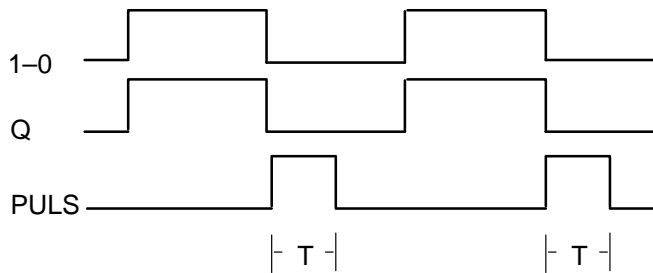
Zur Flankenerkennung ist dieser zusätzliche Hilfsmerker erforderlich, der als direkter Merker oberhalb des Verknüpfungselements angegeben wird. Dieser Merker darf im SPS-Programm nicht nochmals verwendet werden. Der Merker kann über den Ausgang Q abgefragt werden. D.h. der Ausgang Q kann mit einem anderen Verknüpfungselement verbunden werden, oder am Ausgang Q kann auch ein binärer Operand (A, M) projiziert werden.

PULS BINÄR

An diesem Ausgang steht für eine Programmzykluszeit ein Impuls an, falls am Eingang 1-0 eine 1-0-Flanke auftritt.

Dauer des Impulses

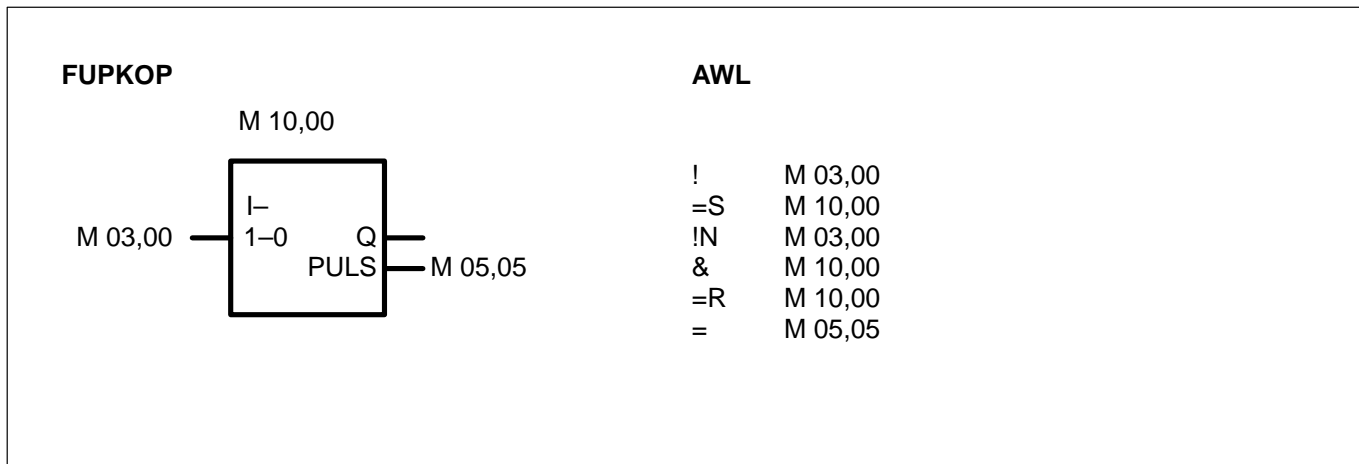
Von der Erkennung der 1-0-Flanke durch das Verknüpfungselement bis zur erneuten Bearbeitung dieses Verknüpfungselements im nächsten Programmzyklus.



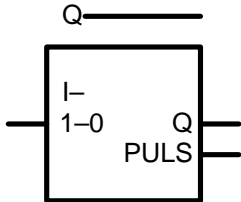
T: Programmzykluszeit der SPS

Der Eingang und die Ausgänge sind weder doppelbar noch invertierbar.

Beispiel



VE-FUP-Definition

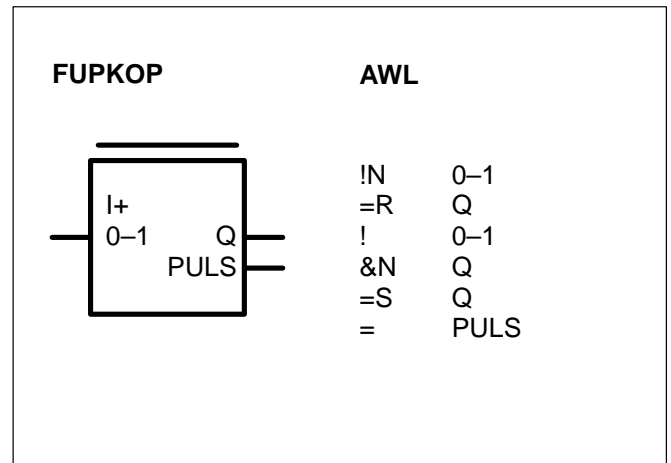


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
1-0	E	L	N	P	J	0	0
Q	A	L	N	P	J	0	0
PULS	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!	PP 0	1-0	Eingang BINÄR (1-0 Flanke)
00002	=S	PP 0	Q	Hilfsmerker BINÄR
00004	!N	PP 0	1-0	Eingang BINÄR (1-0 Flanke)
00006	&	PP 0	Q	Hilfsmerker BINÄR
00008	=R	PP 0	Q	Hilfsmerker BINÄR
00010	=	PP 0	PULS	Ausgang BINÄR (Impuls)

Eine positive Flanke am Eingang 0–1 erzeugt am Ausgang PULS einen Impuls von der Dauer eines SPS–Programmzyklusses.



Parameter

0–1	BINÄR	E, M, A, S, K	Eingang für 0–1–Flanke
Q	BINÄR	A, M	Ausgang zur Abfrage des direkten Merkers
PULS	BINÄR	A, M	Impuls Ausgang

VE–Daten

Serie 90, 07 KP 62

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	13,2 µs	45 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331 ABB Procontic T200 / 07 KP 62	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Eine positive Flanke am Eingang 0–1 erzeugt am Ausgang PULS einen Impuls von der Dauer eines SPS–Programmzyklusses.

0–1 BINÄR

Der Wert des Operanden an diesem Eingang wird auf eine 0–1–Flanke überwacht.

Q BINÄR

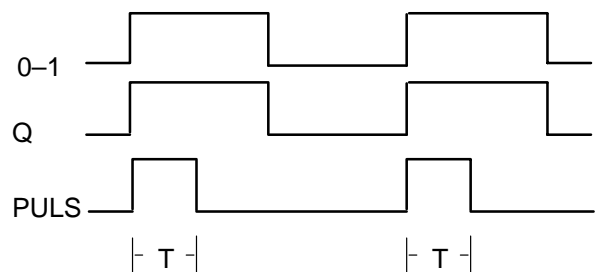
Zur Flankenerkennung ist dieser zusätzliche Hilfsmerker erforderlich, der als direkter Merker oberhalb des Verknüpfungselementes angegeben wird. Dieser Merker darf im SPS–Programm nicht nochmals verwendet werden. Der Merker kann über den Ausgang Q abgefragt werden. D.h. der Ausgang Q kann mit einem anderen Verknüpfungselement verbunden werden, oder am Ausgang Q kann auch ein binärer Operand (A, M) projiziert werden.

PULS BINÄR

An diesem Ausgang steht für eine Programmzykluszeit ein Impuls an, falls am Eingang 0–1 ein 0–1–Flanke auftritt.

Dauer des Impulses

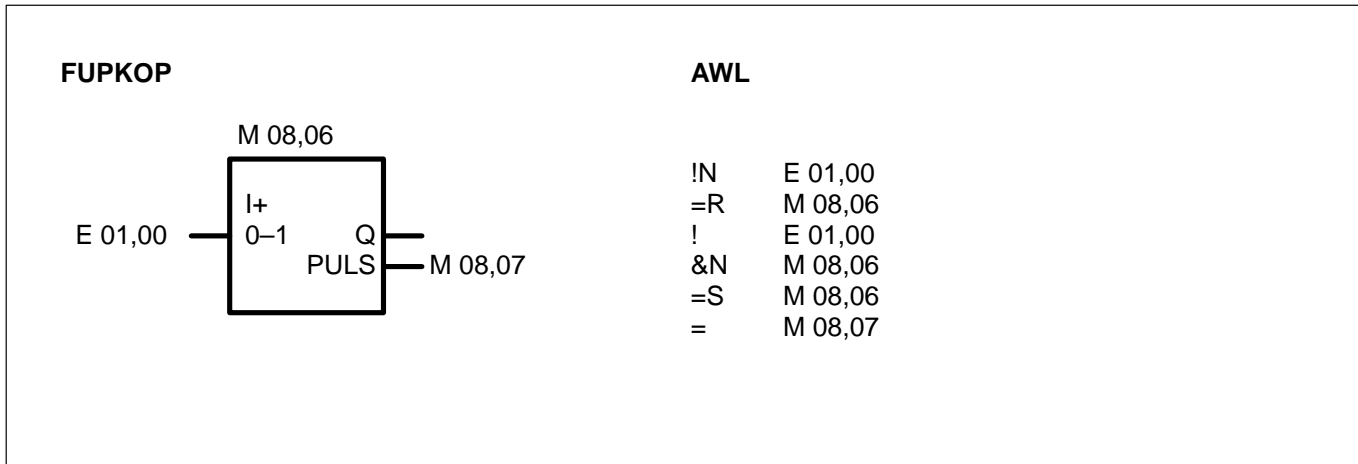
Von der Erkennung der 0–1–Flanke durch das Verknüpfungselement bis zur erneuten Bearbeitung dieses Verknüpfungselementes im nächsten Programmzyklus.



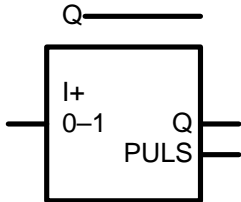
T: Programmzykluszeit.

Der Eingang und die Ausgänge sind weder doppelbar noch invertierbar.

Beispiel



VE-FUP-Definition

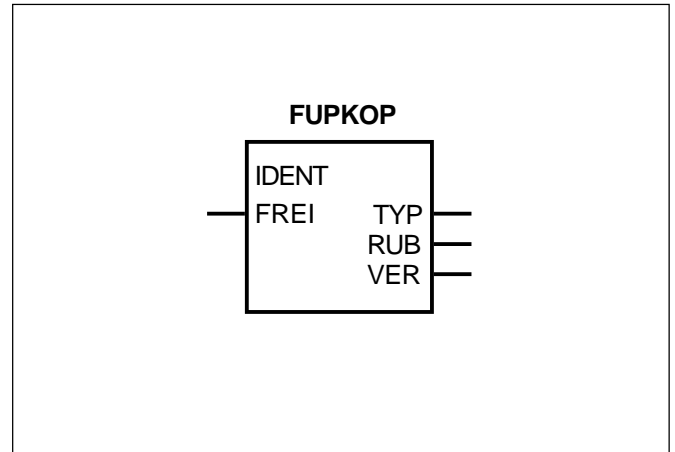


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0-1	E	L	N	P	J	0	0
Q	A	L	N	P	J	0	0
PULS	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!N	PP 0	0-1	Eingang BINÄR (0-1 Flanke)
00002	=R	PP 0	Q	Hilfsmerker BINÄR
00004	!	PP 0	0-1	Eingang BINÄR (0-1 Flanke)
00006	&N	PP 0	Q	Hilfsmerker BINÄR
00008	=S	PP 0	Q	Hilfsmerker BINÄR
00010	=	PP 0	PULS	Ausgang BINÄR (Impuls)

Der Baustein dient zur Identifizierung des Gerätes. Er gibt bei gesetzter Freigabe (FREI = 1) an seinen Ausgängen den Gerätetyp, die Rubrik und die Software-Version aus.



Parameter

FREI	BINÄR	E, K, M, S	Freigabe der Bausteinbearbeitung
TYP	WORT	AW, MW,	Gerätetyp
RUB	WORT	AW, MW	Rubrik
VER	WORT	AW, MW	Software-Version

VE-Daten

Aktualisierung der Ausgänge:	nein
Anzahl der Vergangenheitswerte:	keine

Verfügbar ab:	07 KR 91 R2 Index c, 07 KT 92 R2 Index c, 07 KT 93 R2 Index f, 07 KT 94
---------------	--

Beschreibung

Der Baustein dient zur Identifizierung des Gerätes. Er gibt bei gesetzter Freigabe (FREI = 1) an seinen Ausgängen den Gerätetyp, die Rubrik und die Software-Version aus.

FREI BINÄR

Freigabe des Bausteins mit FREI = 1. Bei FREI = 0 werden die Ausgänge nicht verändert.

TYP WORT

Am Ausgang TYP wird bei FREI = 1 der Gerätetyp ausgegeben.

Beispiel: TYP = 94 → das Gerät ist eine Zentraleinheit 07 KT 94.

RUB WORT

Am Ausgang RUB wird bei FREI = 1 die Rubrik ausgegeben.

Beispiel: RUB = 101 → das Gerät hat die Rubrik R0101.

VER WORT

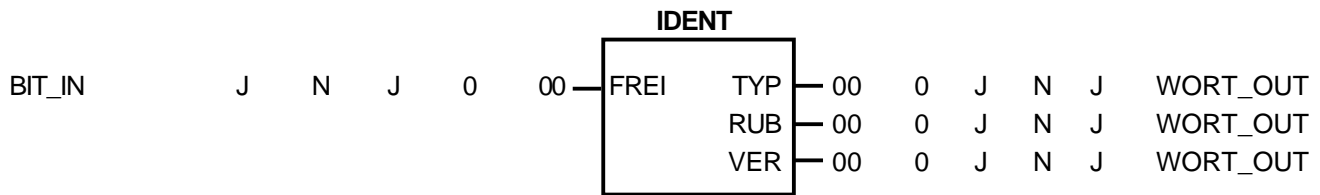
Am Ausgang VER wird bei FREI = 1 die Software-Version ausgegeben.

Beispiel: VER = 105 → das Gerät hat die Softwareversion 1.5.

Bei FREI = 0 wird der Baustein nicht bearbeitet, d. h. die Ausgänge werden nicht verändert.

Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr		Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	--	----	-----	-----	-----	-----	--------------

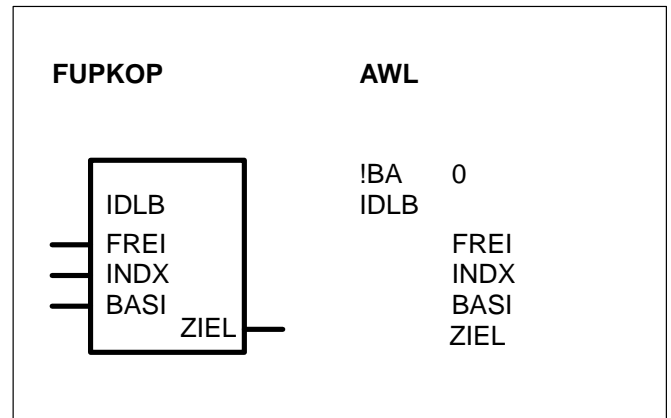


Source

SN	ZN	Befehl / Variable	Symbol	Langtext
----	----	-------------------	--------	----------

00000	00000	IDENT		
	00001	PP 00	FREI	
	00002	PP 00	TYP	
	00003	PP 00	RUB	
	00004	PP 00	VER	
	00005			

Der Funktionsbaustein dient zum indizierten Lesen von Binärvariablen.



Parameter

FREI	BINÄR	E, A, M, K, S	Freigabe für den Baustein Frei = 0: Baustein wird nicht bearbeitet Frei = 1: Der Wert der Quellvariablen wird gelesen und der Zielvariablen zugewiesen.
INDX	WORT	EW, AW, MW, KW	Der Index und die Basisvariable ergeben die Quellvariable
BASI	BINÄR	E, A, M, K	Basisvariable
ZIEL	BINÄR	A, M	Zielvariable

VE-Daten

Laufzeit:		
Grundlaufzeit:		35 µs
Zusätzliche Laufzeit:		keine
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / Kommunikationsprozessor 07 KP 62

Beschreibung

Der Funktionsbaustein dient zum indizierten Lesen von Binärvariablen.

Die zu lesende Quellvariable ergibt sich durch Indizierung der Basisvariablen. Der Wert der gelesenen Quellvariablen wird der Zielvariablen zugewiesen.

Die Ein- und Ausgänge sind weder doppelbar noch invertierbar noch negierbar.

Bestimmung von Gruppen- und Kanal-Nummer des Quell-Merkers (Quellvariable) aus dem Basis-Merker und dem Index INDX.

Der Quell-Merker heißt:
M (G_Basis + A) , (K_Basis + B)

mit:

G_Basis: Gruppen-Nummer des Basis-Merkers
K_Basis: Kanal-Nummer des Basis-Merkers

Formel:

$$\frac{\text{INDX}}{16} = A \quad \text{Rest B}$$

Gruppen-Nr. des Quell-Merkers:
Gruppen-Nr. des Basis-Merkers + A

Kanal-Nr. des Quell-Merkers:
Kanal-Nr. des Basis-Merkers + B

Beispiel:

Basisvariable: M 00,00
INDX = 10 → 10 : 16 = A = 0, Rest B = 10
→ Quellvariable:
M(00+A),(00+B) = M(00+0),(00+10) = M 00,10

Weitere Beispiele

Basisvariable	INDX	Quellvariable
M 00,00	0	M 00,00
M 00,00	2	M 00,02
M 00,00	16	M 01,00
M 00,02	18	M 01,04
E 00,00	3	E 00,03
E 01,01	5	E 01,06
A 05,05	6	A 05,11
K 00,04	8	K 00,12

ZIEL BINÄR

Am Ausgang Ziel wird die Zielvariable angegeben. Der Zielvariablen ZIEL wird der Wert der ausgewählten Quellvariablen zugewiesen.

FREI BINÄR

Freigabe für den Baustein

FREI = 0 → Baustein wird nicht bearbeitet

FREI = 1 → Der Wert der Quellvariablen wird gelesen und der Zielvariablen ZIEL zugewiesen.

INDX WORT

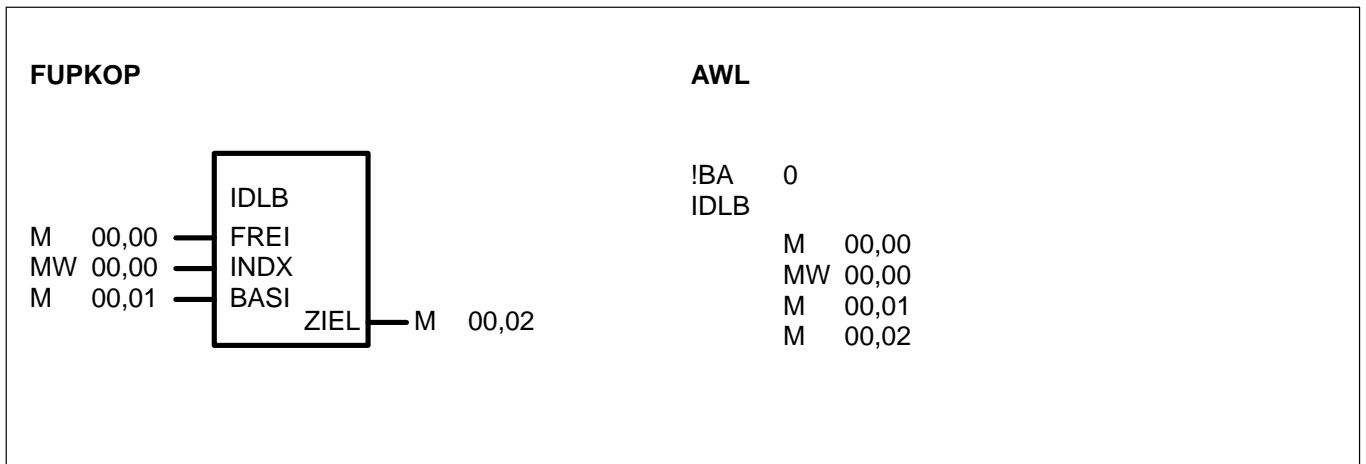
Am Eingang INDX wird der Indexwert angegeben. Aus dem Index INDX und der Basisvariablen ergibt sich die Quellvariable (Berechnung siehe oben).

Wertebereich: $-16383 \leq \text{INDX} \leq +16383$

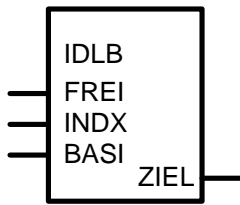
BASI BINÄR

Am Eingang BASI wird die Basisvariable angegeben. Aus dem Index INDX und der Basisvariablen ergibt sich die Quellvariable (Berechnung siehe oben).

Beispiel



VE-FUP-Definition

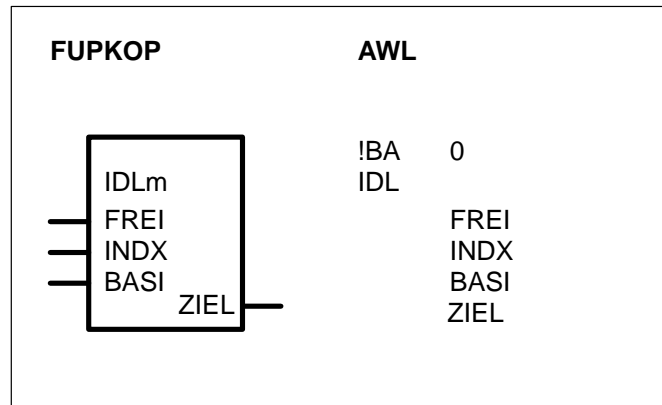


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
INDX	E	W	N	P	J	0	0
BASI	E	L	N	P	J	0	0
ZIEL	E	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	IDLB			
00002		PP 0	FREI	Freigabe (BINÄR)
00003		PP 0	INDX	Index bezogen auf Basis (WORT)
00004		PP 0	BASI	Basisvariable (BINÄR)
00005		PP 0	ZIEL	Zielvariable (BINÄR)

Der Funktionsbaustein dient zum indizierten Lesen von Wortvariablen.



Parameter

FREI	BINÄR	E, A, M, K, S	Freigabe für den Baustein Frei = 0: Baustein wird nicht bearbeitet Frei = 1: Der Wert der Quellvariablen wird gelesen und der Zielvariablen zugewiesen.
INDX	WORT	EW, AW, MW, KW	Der Index und die Basisvariable ergeben die Quellvariable
BASI	WORT	EW, AW, MW, KW	Basisvariable
ZIEL	WORT	AW, MW	Zielvariable

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	35 µs	63 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein dient zum indizierten Lesen von Wortvariablen.

Die zu lesende Quellvariable ergibt sich durch Indizierung der Basisvariablen. Der Wert der gelesenen Quellvariablen wird der Zielvariablen zugewiesen.

Die Ein- und Ausgänge sind weder doppelbar noch invertierbar noch negierbar.

Bestimmung von Gruppen- und Kanal-Nummer des Quell-Merkers (Quellvariable) aus dem Basis-Merker und dem Index INDX.

Der Quell-Merker heißt:
MW (G_Basis + A) , (K_Basis + B)

mit:

G_Basis: Gruppen-Nummer des Basis-Merkers
K_Basis: Kanal-Nummer des Basis-Merkers

Formel:

$$\frac{INDX}{16} = A \quad \text{Rest } B$$

Gruppen-Nr. des Quell-Merkers:
Gruppen-Nr. des Basis-Merkers + A

Kanal-Nr. des Quell-Merkers:
Kanal-Nr. des Basis-Merkers + B

Beispiel:

Basisvariable: MW 00,00
INDX = 10 → 10 : 16 = A = 0, Rest B = 10
→ Quellvariable:
MW(00+A),(00+B) = MW(00+0),(00+10) = MW 00,10

Weitere Beispiele

Basisvariable	INDX	Quellvariable
MW 00,00	0	MW 00,00
MW 00,00	2	MW 00,02
MW 00,00	16	MW 01,00
MW 00,02	18	MW 01,04
EW 00,00	3	EW 00,03
EW 01,01	5	EW 01,06
AW 05,05	6	AW 05,11
KW 00,04	8	KW 00,12

ZIEL WORT

Am Ausgang Ziel wird die Zielvariable angegeben. Der Zielvariablen ZIEL wird der Wert der ausgewählten Quellvariablen zugewiesen.

FREI BINÄR

Freigabe für den Baustein

FREI = 0 → Baustein wird nicht bearbeitet

FREI = 1 → Der Wert der Quellvariablen wird gelesen und der Zielvariablen ZIEL zugewiesen.

INDX WORT

Am Eingang INDX wird der Indexwert angegeben. Aus dem Index INDX und der Basisvariablen ergibt sich die Quellvariable (Berechnung siehe oben).

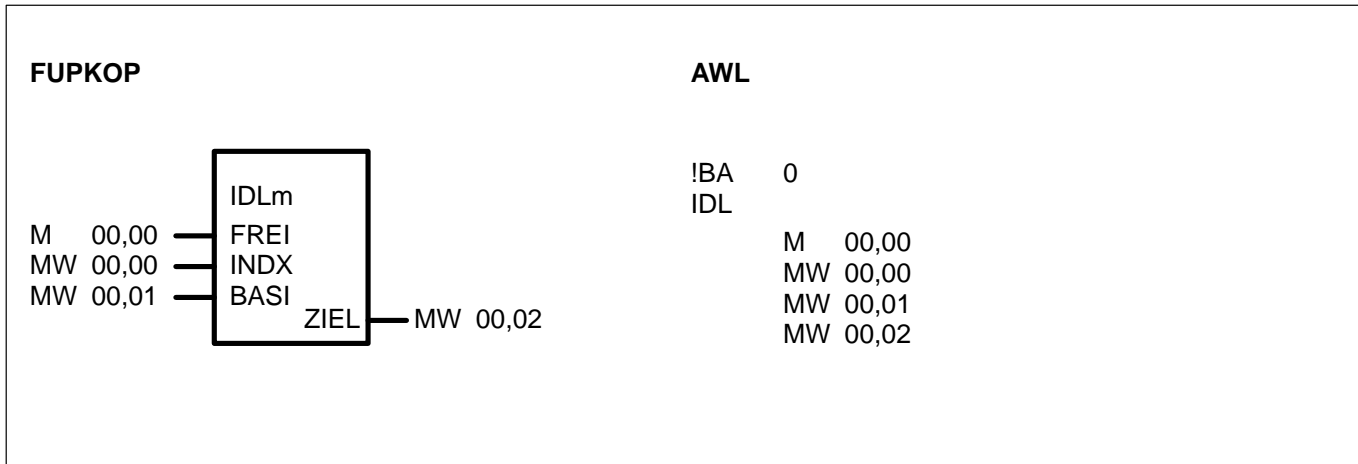
Wertebereich: $-16383 \leq \text{INDX} \leq +16383$

Falls INDX außerhalb des Bereiches liegt, wird der Funktionsblock nicht bearbeitet.

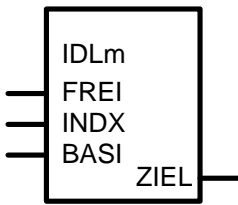
BASI WORT

Am Eingang BASI wird die Basisvariable angegeben. Aus dem Index INDX und der Basisvariablen ergibt sich die Quellvariable (Berechnung siehe oben).

Beispiel



VE-FUP-Definition

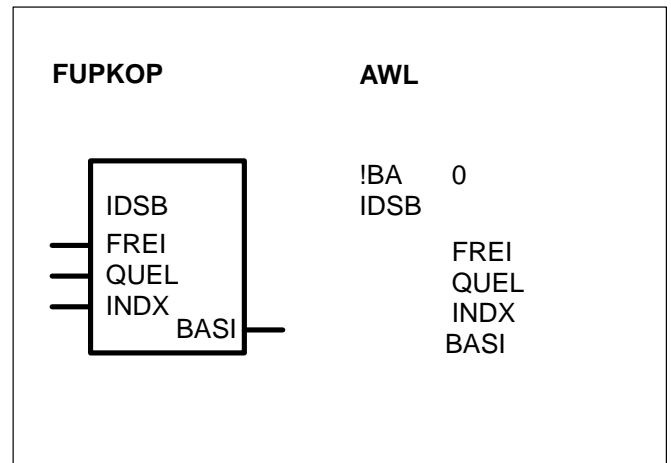


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
INDX	E	W	N	P	J	0	0
BASI	E	W	N	P	J	0	0
ZIEL	E	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr (vorbelegt 0)
00001	IDL			
00002		PP 0	FREI	Freigabe (BINÄR)
00003		PP 0	INDX	Index bezogen auf Basis (WORT)
00004		PP 0	BASI	Basisvariable (WORT)
00005		PP 0	ZIEL	Zielvariable (WORT)

Der Funktionsbaustein dient zum indizierten Schreiben von Binärvariablen.



Parameter

FREI	BINÄR	E, A, M, K, S	Freigabe für den Baustein FREI = 0: Baustein wird nicht bearbeitet FREI = 1: Der Wert der Quellvariablen wird gelesen und der Zielvariablen zugewiesen
QUEL	BINÄR	E, A, M, K	Quellvariable
INDX	WORT	EW, AW, MW, KW	Aus dem Index und der Basisvariablen ergibt sich die aktuelle Zielvariable
BASI	BINÄR	A, M	Basisvariable

VE-Daten

Laufzeit:		
Grundlaufzeit:		35 µs
Zusätzliche Laufzeit:		keine
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / Kommunikationsprozessor 07 KP 62

Beschreibung

Der Funktionsbaustein dient zum indizierten Schreiben von Binärvariablen.

Bei Freigabe des Bausteins wird der Wert der Quellvariablen gelesen und der Zielvariablen zugewiesen. Die Zielvariable wird durch Indizierung der Basisvariablen bestimmt.

Die Ein- und Ausgänge sind weder doppelbar noch negierbar noch invertierbar.

Bestimmung von Gruppen- und Kanal-Nummer des Ziel-Merkers (Zielvariable) aus dem Basis-Merker und dem Index INDX.

Der Ziel-Merker heißt: $M(G_Basis+A), (K_Basis+B)$ mit:

- G_Basis: Gruppen-Nummer des Basis-Merkers
- K_Basis: Kanal-Nummer des Basis-Merkers

Formel:

$$INDX \\ \text{---} = A \quad \text{Rest } B \\ 16$$

Gruppen-Nr. des Ziel-Merkers:
Gruppen-Nr. des Basis-Merkers + A

Kanal-Nr. des Ziel-Merkers:
Kanal-Nr. des Basis-Merkers + B

Beispiel:

Basisvariable: M 00,00
 -> INDX = 10 -> 10 : 16 = A = 0, Rest B = 10
 -> Zielvariable:
 $M(00+A), (00+B) = M(00+0), (00+10) = M 00,10$
 -> Zielvariable: M 00,10

Weitere Beispiele:

Basisvariable	INDX	Zielvariable
M 00,00	0	M 00,00
M 00,00	2	M 00,02
M 00,00	16	M 01,00
M 00,02	18	M 01,04
A 00,00	3	A 00,03
A 00,15	1	A 01,00
A 05,05	6	A 05,11
A 05,05	12	A 06,01

BASI BINÄR

Am Eingang BASI wird die Basisvariable angegeben. Aus dem Index INDX und der Basisvariablen ergibt sich die Zielvariable (Berechnung siehe oben).

FREI BINÄR

Freigabe für den Baustein

FREI = 0: Baustein wird nicht bearbeitet.

FREI = 1: Der Wert der Quellvariablen wird gelesen und der Zielvariablen ZIEL zugewiesen.

QUEL BINÄR

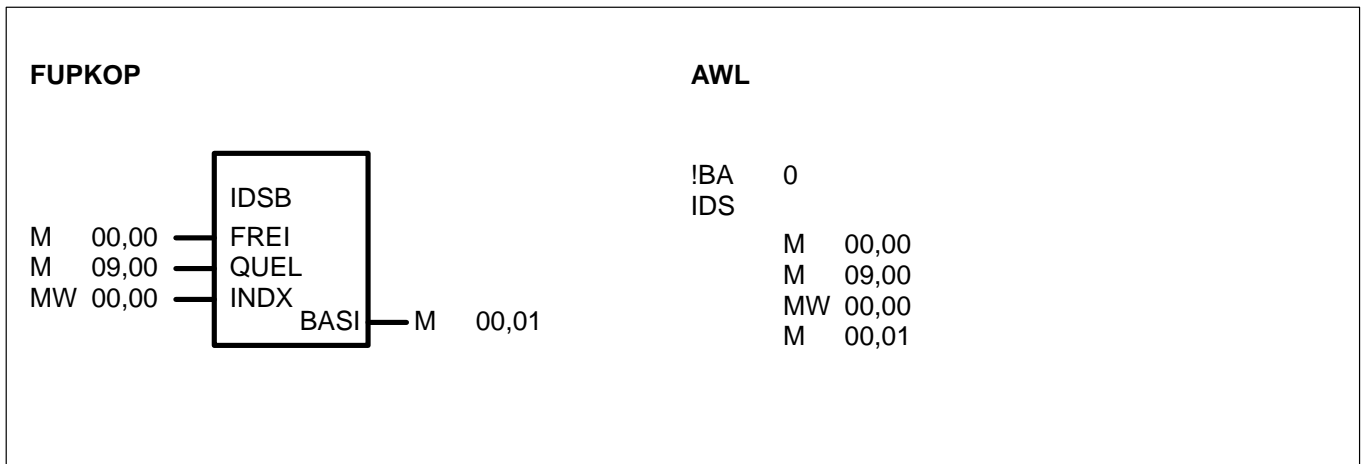
Am Eingang QUEL wird die Quellvariable angegeben. Der Wert dieser Variablen wird gelesen und der Zielvariablen zugewiesen.

INDX WORT

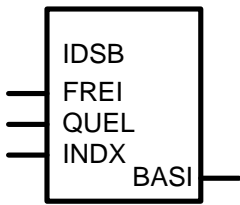
Am Eingang INDX wird der Indexwert angegeben. Aus dem Index INDX und der Basisvariablen ergibt sich die Zielvariable (Berechnung siehe oben).

Wertebereich: $-16383 \leq \text{INDX} \leq +16383$

Beispiel



VE-FUP-Definition

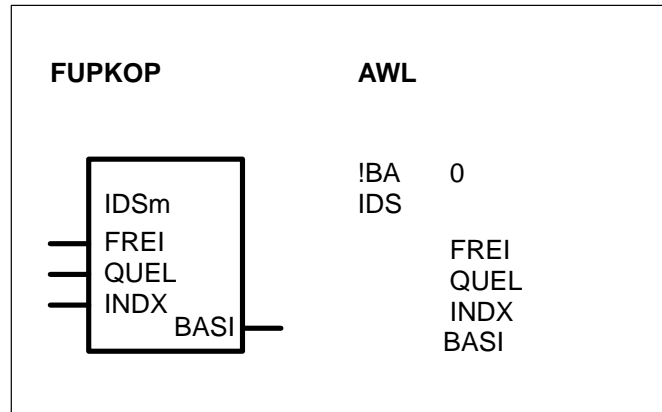


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
QUEL	E	L	N	P	J	0	0
INDX	E	W	N	P	J	0	0
BASI	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	IDSB			
00002		PP 0	FREI	Freigabe (BINÄR)
00003		PP 0	QUEL	Quellvariable (BINÄR)
00004		PP 0	INDX	Index bezogen auf Basis (WORT)
00005		PP 0	BASI	Basisvariable für Ziel (BINÄR)

Der Funktionsbaustein dient zum indizierten Schreiben von Wortvariablen.



Parameter

FREI	BINÄR	E, A, M, K, S	Freigabe für den Baustein FREI = 0: Baustein wird nicht bearbeitet FREI = 1: Der Wert der Quellvariablen wird gelesen und der Zielvariablen zugewiesen
QUEL	WORT	EW, AW, MW, KW	Quellvariable
INDX	WORT	EW, AW, MW, KW	Aus dem Index und der Basisvariablen ergibt sich die aktuelle Zielvariable
BASI	WORT	AW, MW	Basisvariable

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	35 µs	99 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein dient zum indizierten Schreiben von Wortvariablen.

Bei Freigabe des Bausteins wird der Wert der Quellvariablen gelesen und der Zielvariablen zugewiesen. Die Zielvariable wird durch Indizierung der Basisvariablen bestimmt.

Die Ein- und Ausgänge sind weder doppelbar noch negierbar noch invertierbar.

Bestimmung von Gruppen- und Kanal-Nummer des Ziel-Merkers (Zielvariable) aus dem Basis-Merker und dem Index INDX.

Der Ziel-Merker heißt: MW (G_Basis+A),(K_Basis+B) mit:

- G_Basis: Gruppen-Nummer des Basis-Merkers
- K_Basis: Kanal-Nummer des Basis-Merkers

Formel:

$$\text{INDX} \\ \text{—} = A \quad \text{Rest } B \\ 16$$

Gruppen-Nr. des Ziel-Merkers:
Gruppen-Nr. des Basis-Merkers + A

Kanal-Nr. des Ziel-Merkers:
Kanal-Nr. des Basis-Merkers + B

Beispiel:

Basisvariable: MW 00,00
 -> INDX = 10 -> 10 : 16 = A = 0, Rest B = 10
 -> Zielvariable:
 MW(00+A),(00+B) = MW (00+0),(00+10) =
 MW 00,10
 ->Zielvariable: MW 00,10

Weitere Beispiele:

Basisvariable	INDX	Zielvariable
MW 00,00	0	MW 00,00
MW 00,00	2	MW 00,02
MW 00,00	16	MW 01,00
MW 00,02	18	MW 01,04
AW 00,00	3	AW 00,03
AW 00,15	1	AW 01,00
AW 05,05	6	AW 05,11
AW 05,05	12	AW 06,01

BASI WORT

Am Eingang BASI wird die Basisvariable angegeben. Aus dem Index INDX und der Basisvariablen ergibt sich die Zielvariable (Berechnung siehe oben).

FREI BINÄR

Freigabe für den Baustein

FREI = 0: Baustein wird nicht bearbeitet.

FREI = 1: Der Wert der Quellvariablen wird gelesen und der Zielvariablen ZIEL zugewiesen.

QUEL WORT

Am Eingang QUEL wird die Quellvariable angegeben. Der Wert dieser Variablen wird gelesen und der Zielvariablen zugewiesen.

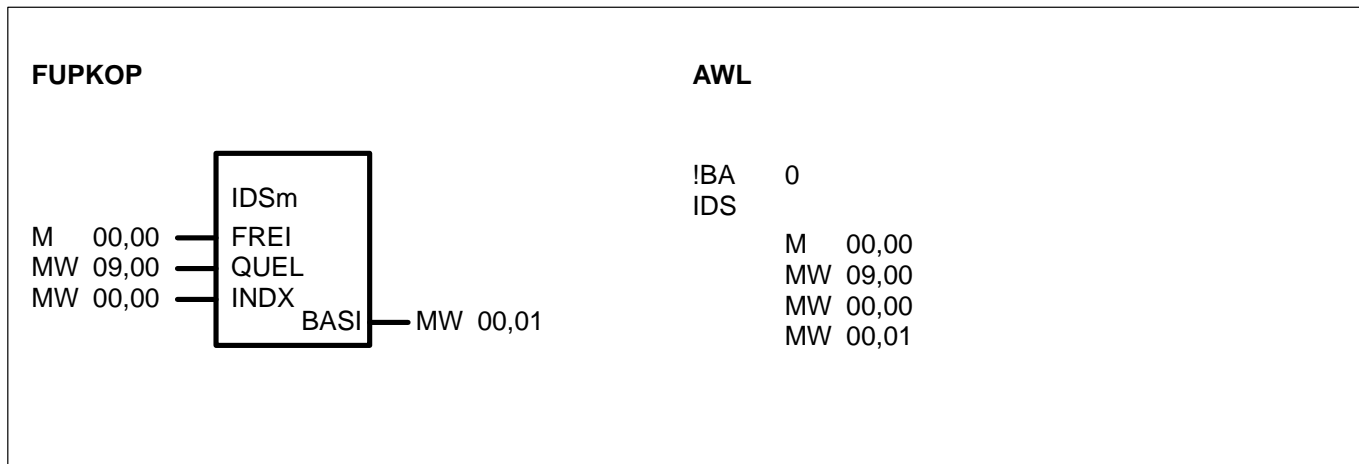
INDX WORT

Am Eingang INDX wird der Indexwert angegeben. Aus dem Index INDX und der Basisvariablen ergibt sich die Zielvariable (Berechnung siehe oben).

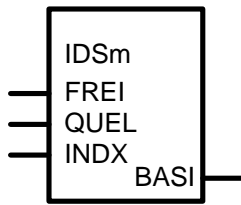
Wertebereich: $-16383 \leq \text{INDX} \leq +16383$

Falls INDX außerhalb des Bereiches liegt, wird der Funktionsblock nicht bearbeitet.

Beispiel



VE-FUP-Definition

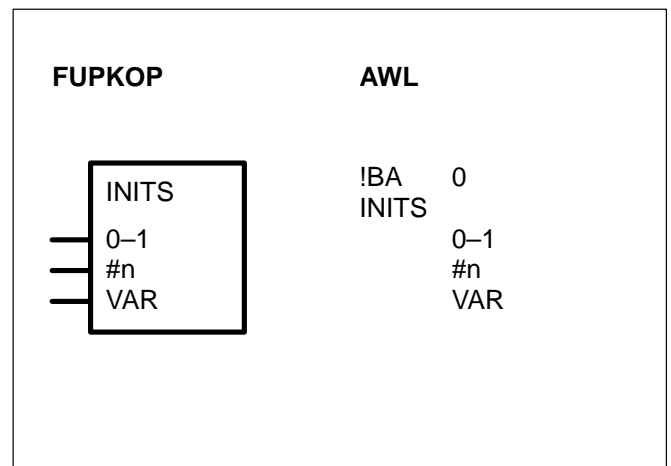


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
QUEL	E	W	N	P	J	0	0
INDX	E	W	N	P	J	0	0
BASI	A	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	IDS			
00002		PP 0	FREI	Freigabe (BINÄR)
00003		PP 0	QUEL	Quellvariable (WORT)
00004		PP 0	INDX	Index bezogen auf Basis (WORT)
00005		PP 0	BASI	Basisvariable für Ziel (WORT)

Der Funktionsbaustein dient zur wortweisen Initialisierung von Speicherbereichen im Operandenspeicher mit dem Wert 0.



Parameter

0-1	BINÄR	E, A, M, S	Freigabe für die einmalige Bearbeitung des Bausteins (0/1-Flanke)
#n	DIREKTE KONSTANTE	#, #H	Anzahl (n) der zu initialisierenden Speicher-Worte
VAR	BINÄR, WORT, DOPPELWORT	E, A, M, K, S, EW, AM, MW, KW, MD, KD	Ab dieser Variablen werden die am Eingang #n angegebenen n Speicherworte initialisiert.

VE-Daten

Laufzeit:	
Grundlaufzeit:	40 µs
Zusätzliche Laufzeit:	2,6 µs pro zu initialisierendem Speicher-Wort
Aktualisierung der Ausgänge:	entfällt
Anzahl der Vergangenheitswerte:	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein dient zur wortweisen Initialisierung von Speicherbereichen im Operandenspeicher mit dem Wert 0.

Der Baustein wird bei einer 0/1-Flanke am Eingang 0-1 genau einmal durchlaufen. Dabei werden ab der am Eingang VAR angegebenen Variablen *n Speicherworte* mit dem Wert 0 initialisiert. Die am Eingang VAR angegebene Variable kann ein BINÄR-, WORT-, oder DOPPELWORT-Operand sein.

Dabei ist zu beachten:

- BINÄR-Operand:
Belegt im Operandenspeicher 1 BYTE.
- WORT-Operand:
Belegt im Operandenspeicher 1 WORT.
- DOPPELWORT-Operand:
Belegt im Operandenspeicher 2 WORTE.

0-1 BINÄR

Freigabe für den Baustein.

Der Baustein wird bei einer 0/1-Flanke an diesem Eingang genau *einmal* durchlaufen.

#n DIREKTE KONSTANTE

Anzahl (n) der zu initialisierenden Worte.

An diesem Eingang wird die Anzahl der zu initialisierenden *Speicherworte* als DIREKTE KONSTANTE angegeben.

Wertebereich: $0 \leq n \leq 65535$

Bei Vorgabe des Wertes 0 erfolgt keine Initialisierung.

VAR BINÄR, WORT, DOPPELWORT

Ab dieser Variablen (einschließlich) werden n Speicherworte mit dem Wert 0 initialisiert.

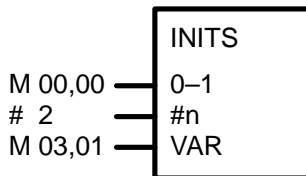
Beispiel

Die 4 Binär-Merker

M 03,01
M 03,02
M 03,03
M 03,04

sind mit dem Wert 0 zu initialisieren. Ein Binär-Merker belegt im Speicher 1 Byte. Es sind also insgesamt 4 Byte = 2 Worte zu initialisieren. Am Eingang #n ist deshalb der Wert 2 als DIREKTE KONSTANTE und am Eingang VAR der Merker M 03,01 anzugeben.

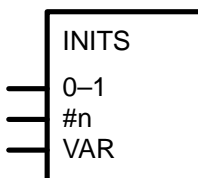
FUPKOP



AWL

```
!BA 0
INITS
M 00,00
# 2
M 03,01
```

VE-FUP-Definition

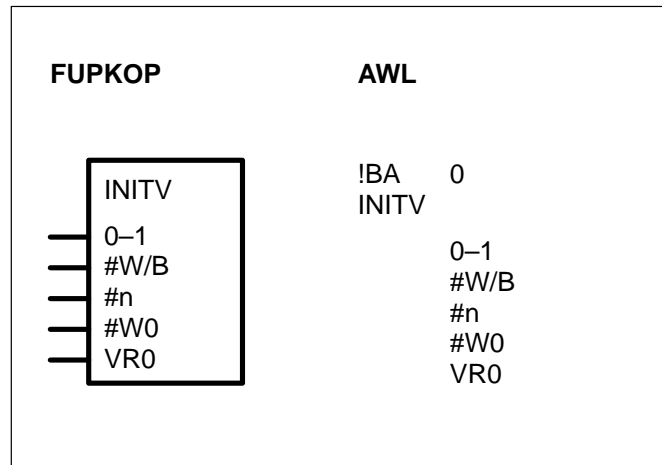


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0-1	E	L	N	P	J	0	0
#n	K	W	N	P	J	0	0
VAR	E	X	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	INITS			
00001		PP 0	0-1	Freigabe-Flanke (BINÄR)
00002		PP 0	#n	Anzahl (DIREKTE KONSTANTE)
00003		PP 0	VAR	Anfangsvariable (BINÄR, WORT, DWORT)

Der Funktionsbaustein dient zur Initialisierung von BINÄR- und WORT-Variablen mit Zahlenwerten.



Parameter

0-1	BINÄR	E, A, M, S	Freigabe für die einmalige Bearbeitung des Bausteins (0/1-Flanke)
#W/B	DIREKTE KONSTANTE	#, #H	Format-Vorgabe #W/B = 1: Initialisierung von WORT-Variablen #W/B = 0: Initialisierung von BINÄR-Variablen
#n	DIREKTE KONSTANTE	#, #H	Anzahl der zu initialisierenden Variablen
#W0	DIREKTE KONSTANTE	#, #H	Initialisierungswert für die nachfolgende Variable; die Eingänge #W0 und VR0 sind paarweise doppelbar
VR0	BINÄR, WORT	E, A, M, EW, AW, MW	die Eingänge VR0 und #W0 sind paarweise doppelbar

VE-Daten

Laufzeit:	
Grundlaufzeit:	34 µs
Zusätzliche Laufzeit:	11 µs pro zusätzlich zu initialisierender Variablen VR1 ... VRn-1
Aktualisierung der Ausgänge:	entfällt
Anzahl der Vergangenheitswerte:	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein dient zur Initialisierung von BINÄR- und WORT-Variablen mit Zahlenwerten.

Die Initialisierungswerte werden als direkte Konstanten vorgegeben. Der Baustein wird bei einer 0/1-Flanke am Eingang 0-1 genau einmal durchlaufen.

#W/B DIREKTE KONSTANTE
Am Eingang #W/B wird das Format der zu initialisierenden Variablen als direkte Konstante vorgegeben. Dabei gilt:

- #W/B = 1: Wort-Variable (EW, AW, MW)
- #W/B = 0: Binär-Variable (E, A, M)

0-1 BINÄR
Bei einer 0/1-Flanke am Eingang 0-1 wird der Baustein genau *einmal* durchlaufen.

#n DIREKTE KONSTANTE
Die Anzahl n der zu initialisierenden Variablen wird am Eingang #n als direkte Konstante vorgegeben.

#W0 DIREKTE KONSTANTE

Am Eingang #W0 wird der Initialisierungswert für die nachfolgende Variable als direkte Konstante vorgegeben.

Die Eingänge #W0 und VR0 sind paarweise doppelbar.

Dabei gilt folgende Zuordnung:

Wert 0	→	Variable 0
:		:
Wert n-1	→	Variable n-1

VR0 BINÄR/WORT

Am Eingang VR0 wird die erste zu initialisierende Variable vorgegeben.

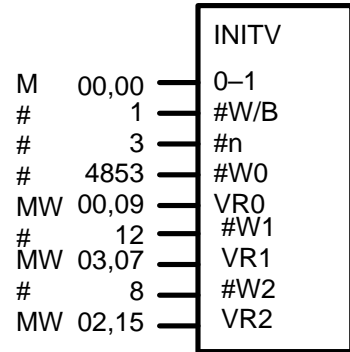
Die Eingänge VR0 und #W0 sind paarweise doppelbar.

Beispiel

Es ist folgende Initialisierung von Variablen durchzuführen:

MW 00,09 = 4853
 MW 03,07 = 12
 MW 02,15 = 8

FUPKOP

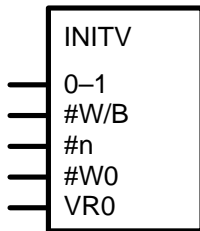


AWL

```

IBA 0
INITV
M 00,00
# 1
# 3
# 4853
MW 00,09
# 12
MW 03,07
# 8
MW 02,15
    
```

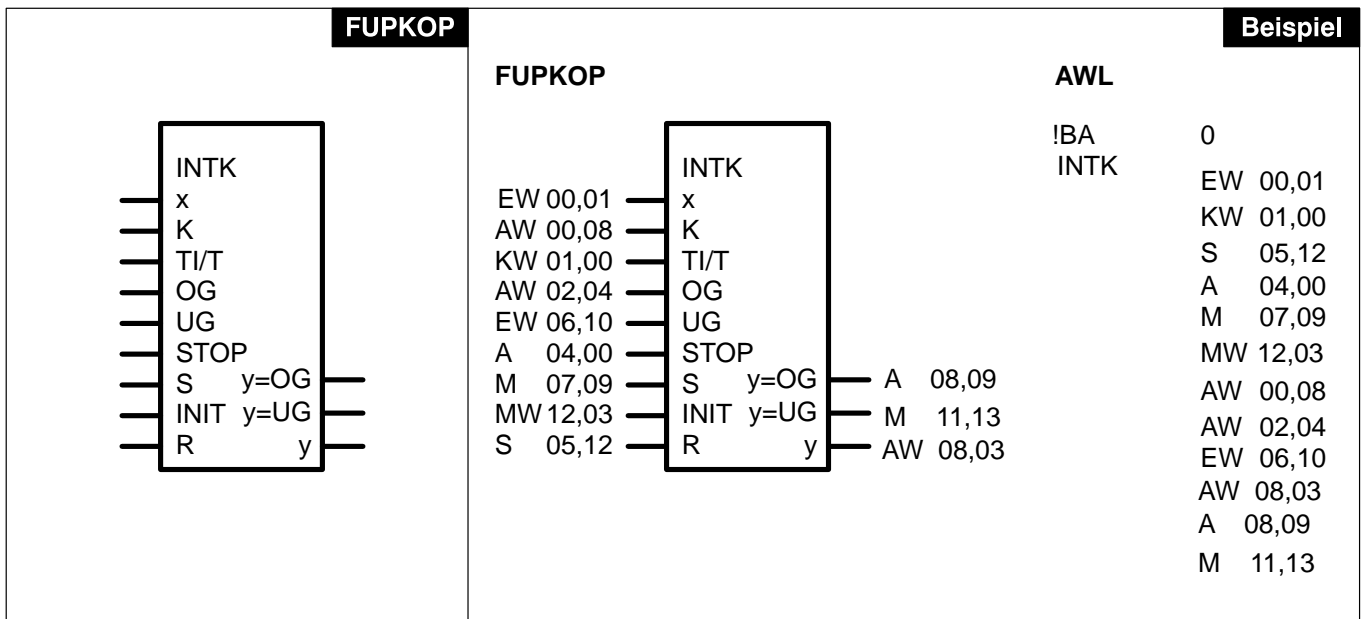
VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0-1	E	L	N	P	J	0	0
#W/B	K	W	N	P	J	0	0
#n	K	W	N	P	J	0	0
#W	K	W	N	P	J	1	1
VR	E	X	N	P	J	1	1

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	INITV			
00002		PP 0	0-1	Freigabe-Flanke (BINÄR)
00003		PP 0	#W/B	Format für Variable (#)
00004		PP 0	#n	Anzahl (#)
	[1		
00005		PP 1	#W	Initwert für Variable (#)
00006		PP 1	VR	zu initialisierende Variable (BINÄR/WORT)
]	1		



Parameter

x	WORT	EW, AW, MW, KW	Regelgröße
K	WORT	EW, AW, MW, KW	Proportionalbeiwert, Ausgabe in Prozent
TI/T	WORT	EW, AW, MW, KW	Integrierzeit normiert auf Zykluszeit
OG	WORT	EW, AW, MW, KW	obere Grenze für die Stellgröße y
UG	WORT	EW, AW, MW, KW	untere Grenze für die Stellgröße y
STOP	BINÄR	E, A, M, S, K	Integrierer Stop
S	BINÄR	E, A, M, S, K	Ausgang auf INIT-Wert setzen
INIT	WORT	EW, AW, MW, KW	Initialwert
R	BINÄR	E, A, M, S, K	Rücksetzen des Ausgangs y auf den Wert 0
y=OG	BINÄR	A, M	Ausgang y hat obere Grenze erreicht
y=UG	BINÄR	A, M	Ausgang y hat untere Grenze erreicht
y	WORT	AW, MW	Stellgröße

Beschreibung

Der Baustein bildet das Integral der mit dem Proportionalbeiwert K multiplizierten Regelgröße x.

Der Ausgang y des Integrierers kann durch:

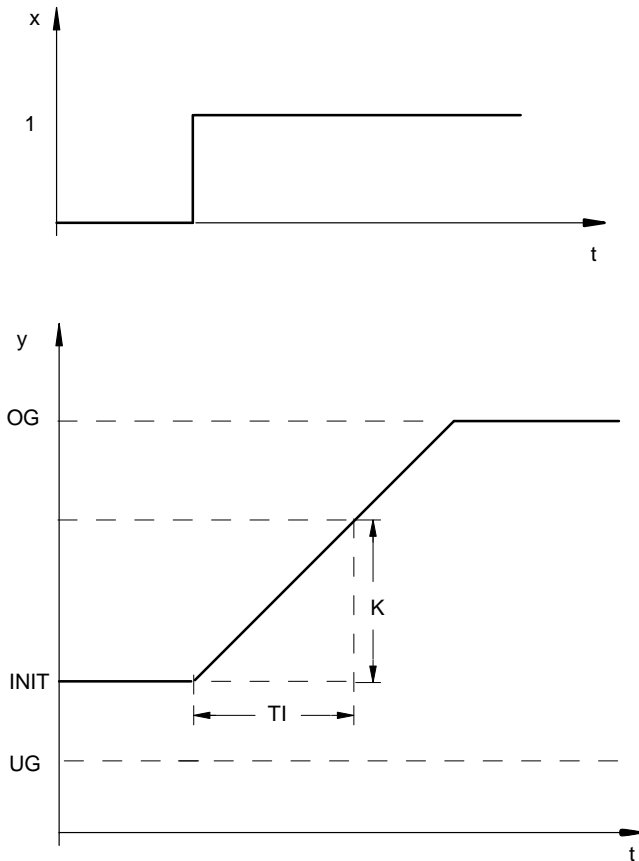
- 1-Signal am Eingang R (Reset) auf den Wert 0 gesetzt werden
- 1-Signal am Eingang STOP auf dem momentanen Wert festgehalten werden
- 1-Signal am Eingang S (Set) auf den Initialwert am Eingang INIT gesetzt werden
- Vorgabe am Eingang OG (Obere Grenze) auf einen Maximalwert begrenzt werden
- Vorgabe am Eingang UG (Untere Grenze) auf einen Minimalwert begrenzt werden

Die Eingänge und Ausgänge sind weder doppelbar noch negierbar/invertierbar.

Übertragungsfunktion:

$$F(s) = \frac{K}{s \cdot T}$$

Übergangsfunktion:



x WORT

Am Eingang x wird der Operand für die Regelgröße vorgegeben.

K WORT

Am Eingang K wird der Proportionalbeiwert angegeben. Er dient zur Gewichtung der Regelgröße am Eingang x. Die Gewichtung erfolgt durch die Multiplikation der Regelgröße mit dem Proportionalbeiwert. Die Angabe des Proportionalbeiwerts erfolgt seit Version V6 (GJP 122900 R301) in Prozent.

Beispiel:

1	=	1	Prozent
55	=	55	Prozent
100	=	100	Prozent
1000	=	1000	Prozent
-100	=	-100	Prozent

– 1 Prozent heißt, daß der Baustein den Wert am Eingang x mit dem Faktor 0,01 multipliziert.

– 100 Prozent heißt, daß der Baustein den Wert am Eingang x mit dem Faktor 1 multipliziert.

– 1000 Prozent heißt, daß der Baustein den Wert am Eingang x mit dem Faktor 10 multipliziert.

TI/T WORT

Am Eingang TI/T wird die Integrierzeit vorgegeben. Die Integrierzeit ist dabei auf die Zykluszeit zu normieren. In der Zeit TI ändert sich der Ausgang y des Integrierers um den Wert $K \cdot x$.

Wertebereich: $0 \leq TI/T \leq 328$

– Werden Werte vorgegeben, die außerhalb des zulässigen Wertebereichs liegen, so arbeitet die SPS grundsätzlich mit dem Wert 328.

– Eine große Integrierzeit (TI) kann dadurch erreicht werden, daß auch die Zykluszeit T groß gewählt wird. Befindet sich der Baustein innerhalb eines Laufzahlblockes, so ist für ihn die Zykluszeit des Laufzahlblockes und nicht die Zykluszeit (KD 0,0) des SPS-Programmes maßgebend.

OG WORT

Die Stellgröße y kann auf einen Wertebereich begrenzt werden. Am Eingang OG wird der obere Grenzwert für die Stellgröße y vorgegeben.

UG WORT

Die Stellgröße y kann auf einen Wertebereich begrenzt werden. Am Eingang UG wird die untere Grenze für die Stellgröße y vorgegeben.

STOP BINÄR *)

Mit dem Eingang STOP kann der Integriervorgang gestoppt werden.

STOP = 0 → Integriervorgang wird nicht gestoppt

STOP = 1 → Integriervorgang wird gestoppt. D. h. der Ausgang y ändert sich nicht mehr.

S BINÄR *)

Mit dem Eingang S kann die Stellgröße y auf den am Eingang INIT angegebenen Initialwert gesetzt werden. Der Integriervorgang beginnt danach wieder ab dem Initialwert.

S = 0 → kein Setzvorgang

S = 1 → Ausgang y wird auf den angegebenen Initialwert gesetzt.

INIT WORT

Am Eingang INIT wird der Initialwert angegeben, auf den der Ausgang y bei Bedarf zu setzen ist.

R BINÄR *)

Mit dem Eingang R kann der Ausgang y auf den Wert 0 zurückgesetzt werden. Der Integriervorgang beginnt danach wieder ab dem Wert 0.

*) Prioritätsreihenfolge für die Eingänge STOP, S und R:

R	höchste Priorität
STOP	
S	niedrigste Priorität

y=OG BINÄR

Am Ausgang y=OG wird signalisiert, ob der Wert am Ausgang y die vorgegebene obere Grenze erreicht hat. Beim Erreichen des Grenzwerts wird der Integriervorgang automatisch gestoppt.

y=OG = 0 → y hat den Grenzwert nicht erreicht
y=OG = 1 → y hat den Grenzwert erreicht

y=UG BINÄR

Am Ausgang y=UG wird signalisiert, ob der Wert am Ausgang y die vorgegebene untere Grenze erreicht hat. Beim Erreichen des Grenzwerts wird der Integriervorgang automatisch gestoppt.

y=UG = 0 → y hat den Grenzwert nicht erreicht
y=UG = 1 → y hat den Grenzwert erreicht

y WORT

Am Ausgang y wird Stellgröße (Ausgangswert des Integrierers) ausgegeben.

VE-Daten

Laufzeit:

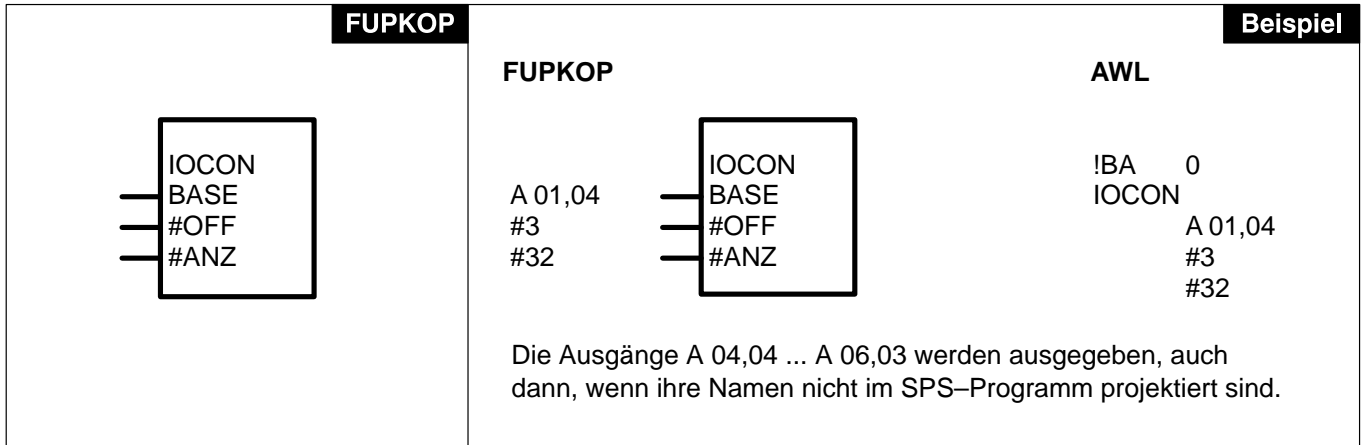
Grundlaufzeit: 189–197 μ s

Zusätzliche Laufzeit: —

Aktualisierung der Ausgänge: ja

Anzahl der Vergangenheitswerte: 3 Worte

Verfügbar ab: ABB Procontic CS31 / 907 PC 331



Parameter

BASE	BINÄR	E, A	Basis Ein- oder Ausgangsbezeichner
#OFF	DIREKTE KONSTANTE	#, #H	Offset der gewünschten Gruppennummer (bezogen auf BASE)
#ANZ	DIREKTE KONSTANTE	#, #H	Anzahl der zu konfigurierenden Kanäle

VE-Daten

Laufzeit:	
Grundlaufzeit:	ca. 0 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	—
Anzahl der Vergangenheitswerte:	0 Worte
Verfügbar ab:	ABB Procontic CS31, 07 KR 91 g, 07 KT 92 g, 07 KT 93

Beschreibung

Bei *binären* Ein- und Ausgängen bedient der SPS-Verarbeiter nur diejenigen Ein- und Ausgänge, deren Bezeichner explizit im SPS-Programm projiziert sind.

Der Funktionsbaustein dient dazu, einen vorgebbaren Bereich von Ein- oder Ausgängen zur Bedienung freizugeben, deren Bezeichner im SPS-Programm *nicht* vorkommen, und die deshalb auch nicht automatisch be-

dient werden. Dies ist z. B. dann erforderlich, wenn mit dem Baustein IDLB binäre Eingänge gelesen werden sollen oder mit dem Baustein IDSB binäre Ausgänge geschrieben werden sollen.

Der am Baustein angegebene E- bzw. A-Bereich wird beim Programmstart im Rahmen der Programmaufbereitung konfiguriert. Im Programmzyklus selbst hat dieser Baustein keine Funktion.

BASE BINÄR

Am Eingang BASE wird der Bezeichner eines binären Ein- oder Ausgangs angegeben. Dieser Bezeichner stellt die Basis dar, auf die sich #OFF und #ANZ beziehen.

#OFF DIREKTE KONSTANTE

Wertebereich: #OFF \geq 0

Am Eingang #OFF wird angegeben, wo der zusätzlich zu bedienende Ein- oder Ausgabe-Bereich anfängt.

Der Anfang dieses zusätzlich zu konfigurierenden Ein- oder Ausgabe-Bereichs ergibt sich wie folgt:

Der am Eingang #OFF angegebene Wert wird zur Gruppennummer des am Eingang BASE projektierten Ein- bzw. Ausgangsbezeichners addiert. Der so errechnete neue Ein- bzw. Ausgangsbezeichner kennzeichnet den Anfang des zusätzlich zu konfigurierenden Ein- bzw. Ausgabebereichs.

Der Bereichsanfang errechnet sich wie folgt:

GruppNr_Bereichsanfang = GruppNr_BASE + #OFF
Kanalnummer_Bereichsanfang = Kanalnummer_BASE

Dabei ist:

GruppNr_Bereichsanfang:
erste Gruppennummer des Bereiches

Kanalnummer_Bereichsanfang:
erste Kanalnummer des Bereiches

#OFF: Wert am Eingang #OFF

GruppNr_BASE:
Gruppennummer des Bezeichners an BASE

Kanalnummer_BASE:
Kanalnummer des Bezeichners an BASE

Beispiel:

- Eingang BASE: A 01,04
d. h. Gruppennummer = 01 und Kanalnummer = 04
- Eingang #OFF: 3
d. h. der zu konfigurierende Bereich fängt an bei:
A (01+3),04 = A 04,04

#ANZ DIREKTE KONSTANTE

Wertebereich: #ANZ \geq 1

Am Eingang #ANZ wird die Anzahl der Ein- bzw. Ausgabe-Kanäle des zusätzlich zu konfigurierenden Ein- bzw. Ausgabebereichs angegeben.

Das Bereichsende errechnet sich wie folgt:

Formel:

$(\#ANZ - 1 + \text{Kanalnummer_BASE}) : 16 = \Delta G$, Rest = Kanalnummer_Bereichsende

GruppNr_Bereichsende =
GruppNr_Bereichsanfang + ΔG

dabei ist:

GruppNr_Bereichsende:
letzte Gruppennummer des Bereiches

Kanalnummer_Bereichsende:
letzte Kanalnummer des Bereiches

GruppNr_Bereichsanfang:
erste Gruppennummer des Bereiches

Kanalnummer_BASE:
Kanalnummer des Bezeichners an BASE

Beispiel:

projektiert:

BASE: A 01,04
#OFF: 3
#ANZ: 32

daraus ergibt sich:

Bereichsanfang: A (01+3),04 = A 04,04

Bereichsende: Formel: $(32 - 1 + 4) : 16 = 2$, Rest 3
—>A (04+2),03 = A 06,03

Bereich: A 04,04 ... A 06,03

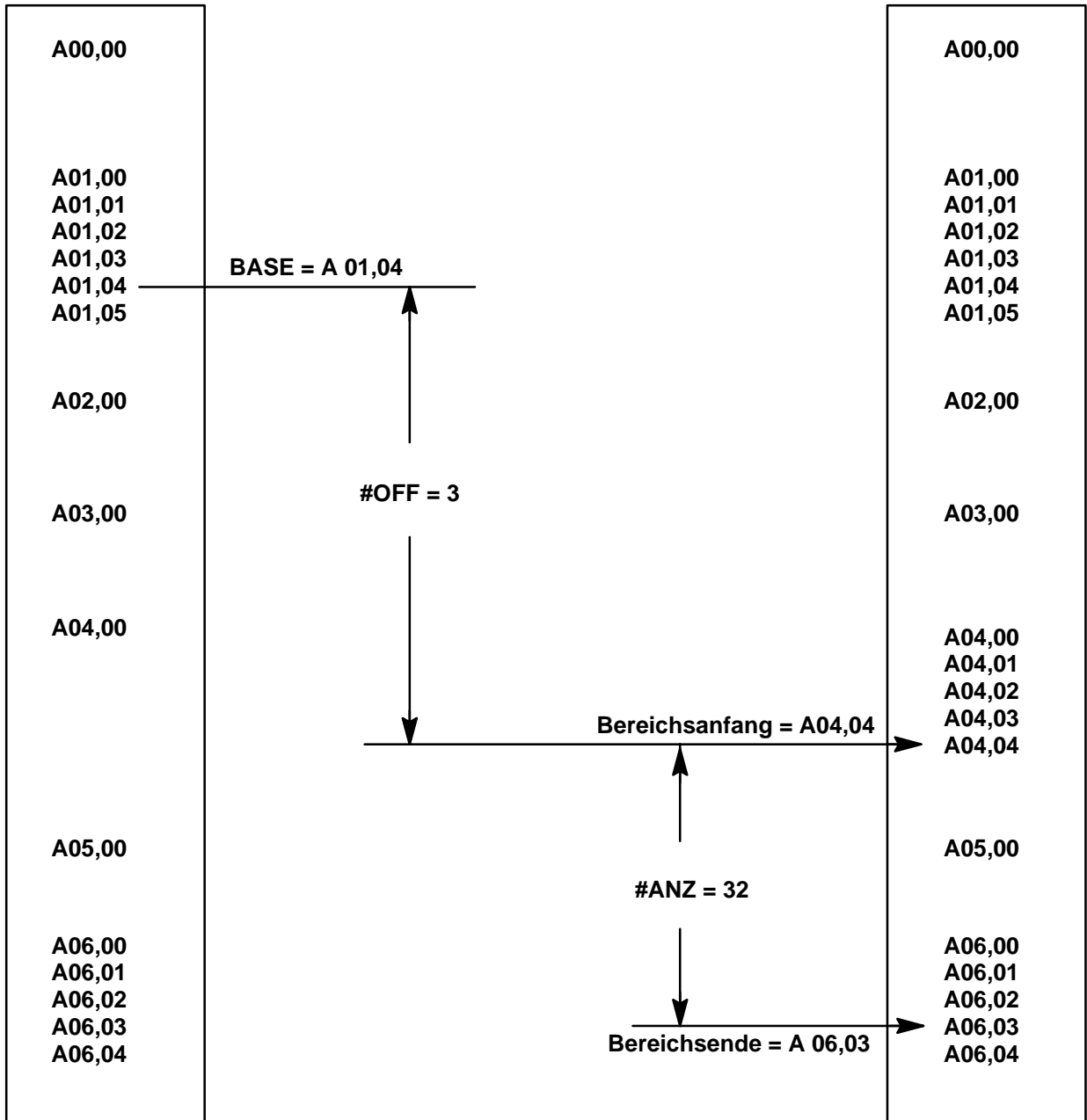
Beispiel

projektiert:

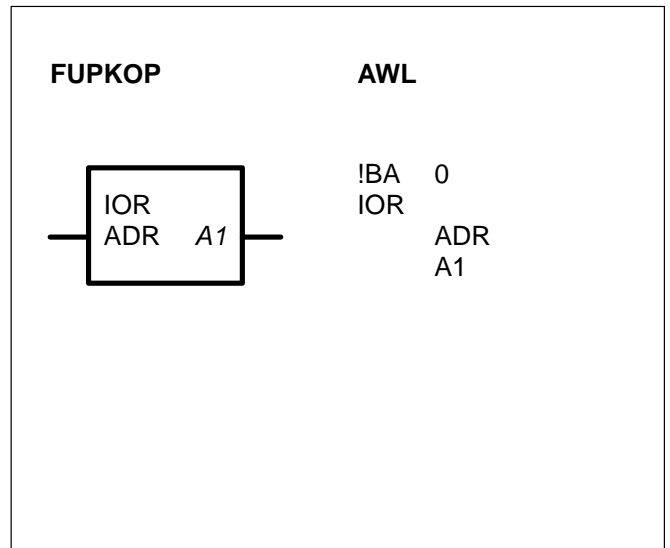
BASE : A 01,04
 #OFF : 3
 #ANZ : 32

Damit erhält man:

Bereichs-Anfang: A (01+#OFF),04 = A 04,04
 Bereichs-Ende: A 06,03



Der Funktionsbaustein liest ein Byte aus dem I/O-Bereich und weist es dem Operanden am Ausgang zu.



Parameter

ADR	WORT	EW, AW, MW, KW	Adresse aus dem I/O-Bereich, deren BYTE-Wert zu lesen ist.
A1	WORT	AW, MW	Der gelesene BYTE-Wert wird dem Ausgang A1 zugewiesen.

VE-Daten

Laufzeit:		
Grundlaufzeit:		24 µs
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

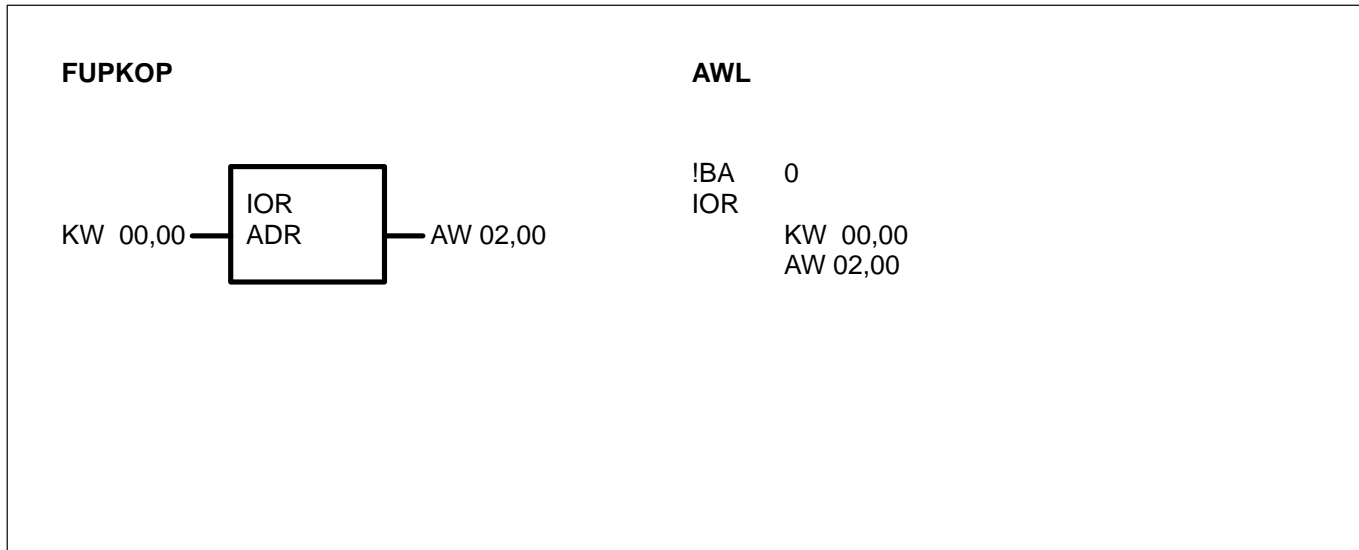
Der Funktionsbaustein liest ein Byte aus dem I/O-Bereich und weist es dem Operanden am Ausgang zu.

Der Eingang und der Ausgang ist weder doppelbar noch negierbar.

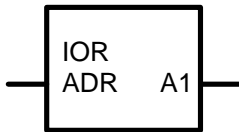
ADR WORT
Der Wert des Operanden am Eingang ADR stellt die zu lesende I/O-Adresse dar.

A1 WORT
Das aus dem I/O-Bereich gelesene Byte wird dem LOW BYTE des Operanden am Ausgang A1 zugewiesen.

Beispiel



VE-FUP-Definition

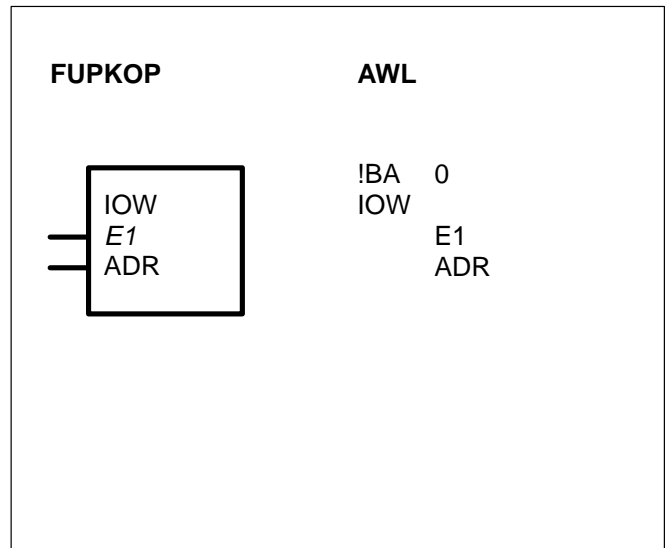


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
ADR	E	W	N	P	J	0	0
A1	A	W	N	P	N	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	IOR			
00002		PP 0	ADR	Eingang WORT (I/O-Adresse)
00003		PP 0	A1	Ausgang WORT (Wert)

Der Funktionsbaustein schreibt das am Eingang E1 angegebene Byte in den I/O-Bereich.



Parameter

E1	WORT	EW, AW, MW, KW	Operand, dessen LOW-Byte gelesen und in den I/O-Bereich geschrieben wird
ADR	WORT	EW, AW, MW, KW	

I/O-Adresse, auf die der Wert des Operanden geschrieben wird.

VE-Daten

Laufzeit:	
Grundlaufzeit:	24 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

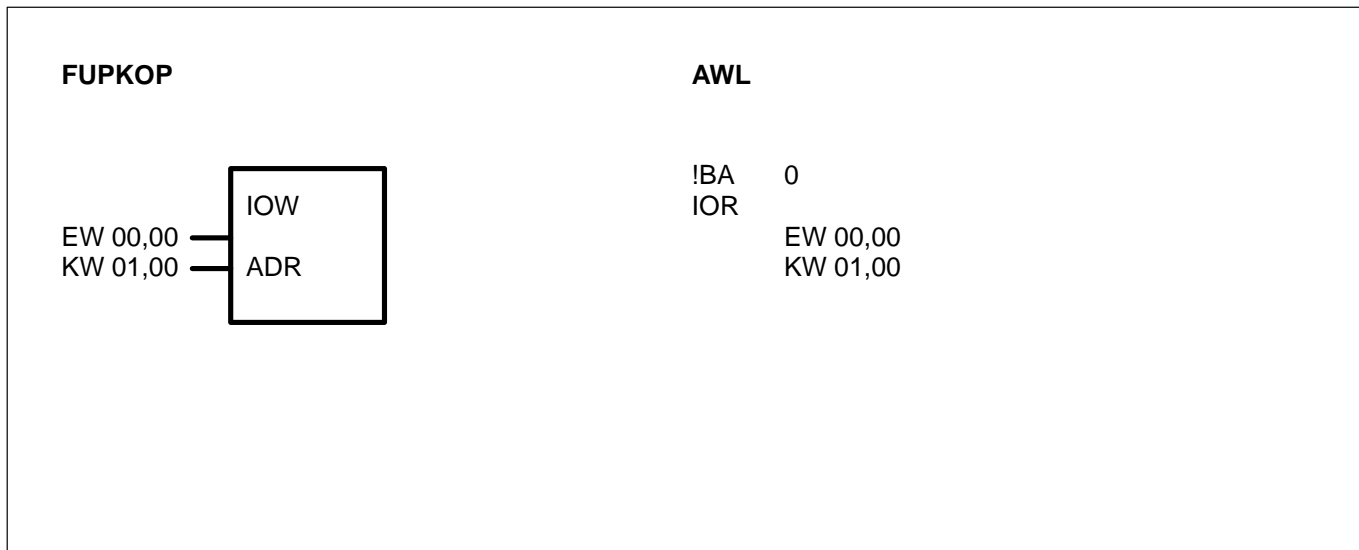
Beschreibung

Der Funktionsbaustein schreibt das am Eingang E1 angegebene Byte in den I/O-Bereich.

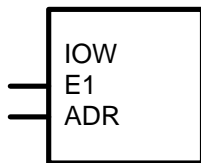
E1 WORT
 Das LOW BYTE des Operanden am Eingang E1 wird in den I/O-Bereich geschrieben.

ADR WORT
 Der Wert des Operanden am Eingang ADR stellt die I/O-Adresse dar, auf die geschrieben wird.

Beispiel



VE-FUP-Definition

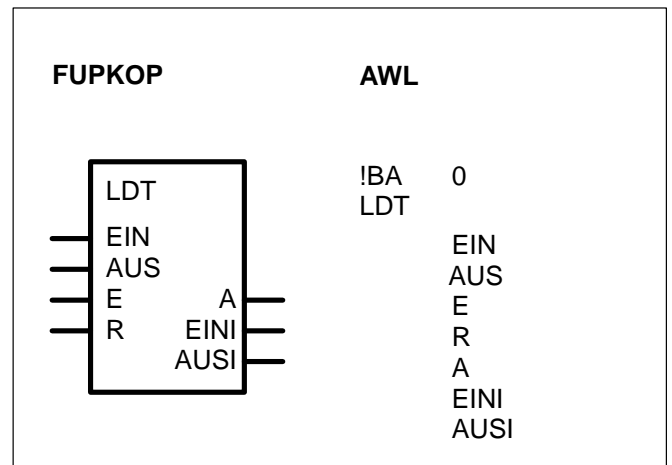


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
ADR	E	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr (vorbelegt 0)
00001	IOW			
00002		PP 0	E1	Eingang WORT (Wert)
00003		PP 0	ADR	Eingang WORT (I/O-Adresse)

Der Funktionsbaustein wechselt bei jeder 0/1-Flanke am Eingang E die Polarität am Ausgang A (Prinzip Stromstoßrelais).



Parameter

EIN	BINÄR	E, A, M, K, S	Dynamischer Eingang zum Einschalten des Ausgangs A
AUS	BINÄR	E, A, M, K, S	Dynamischer Eingang zum Ausschalten des Ausgangs A
E	BINÄR	E, A, M, K, S	Dynamischer Eingang zum Ein- und Ausschalten des Ausgangs A
R	BINÄR	E, A, M, K, S	Statischer Eingang zum Ausschalten des Ausgangs A
A	BINÄR	A, M	Ausgang
EINI	BINÄR	A, M	Impuls für die Dauer eines SPS-Programmzyklusses beim Einschalten des Ausgangs A
AUSI	BINÄR	A, M	Impuls für die Dauer eines SPS-Programmzyklusses beim Ausschalten des Ausgangs A

VE-Daten

Laufzeit:	
Grundlaufzeit:	79...83 µs
Zusätzliche Laufzeit:	keine
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein wechselt bei jeder 0/1-Flanke am Eingang E die Polarität am Ausgang A (Prinzip Stromstoßrelais). Dies gilt aber nur dann, wenn der Eingang R inaktiv ist. Mit den Eingängen EIN, AUS und R kann der Ausgang A zusätzlich beeinflusst werden.

Alle Eingänge mit Ausnahme von R sind flankengesteuert, d. h. sie reagieren nur auf eine 0/1-Flanke. Der Eingang R ist solange wirksam, wie an ihm ein statisches 1-Signal anliegt. Am Ausgang EINI und AUSI stehen bei einer Polaritätsänderung des Ausgangs A kurze Impulse zur Verfügung.

EIN	AUS	E	R	A	
X	X	X	H	L	NA: Invertieren des vorherigen Zustandes von A
X	X	>	L	NA	X: Zustand wird ignoriert
X	>	<	L	L	>: 0/1-Flanke
>	<	<	L	H	<: keine 0/1-Flanke
					L: Zustand 0
					H: Zustand 1

EIN BINÄR

Bei jeder 0/1-Flanke am Eingang EIN wird der Ausgang A gesetzt. Dies gilt aber nur dann, wenn die Eingänge AUS, E und R inaktiv sind. Der Eingang EIN hat also von allen Eingängen die niedrigste Priorität.

AUS BINÄR

Bei jeder 0/1-Flanke am Eingang AUS wird der Ausgang A zurückgesetzt. Der Eingang AUS hat also höhere Priorität als der Eingang EIN, aber niedrigere Priorität als die Eingänge E und R.

E BINÄR

Der Baustein wechselt bei jeder 0/1-Flanke am Eingang E die Polarität am Ausgang A. Dies gilt aber nur dann, wenn der Eingang R inaktiv ist. Der Eingang E hat also höhere Priorität als die Eingänge EIN und AUS, aber niedrigere Priorität als der Eingang R.

R BINÄR

Wenn am Eingang R ein statisches 1-Signal anliegt, wird der Ausgang A zurückgesetzt.

Der Eingang R hat eine höhere Priorität als die Eingänge E und EIN. Ein Prioritätskonflikt mit dem Eingang AUS ist nicht möglich, da beide Eingänge die gleiche Wirkung auf die Ausgänge haben.

A BINÄR

Der Ausgang A wird entsprechend der Definition der Eingänge verändert.

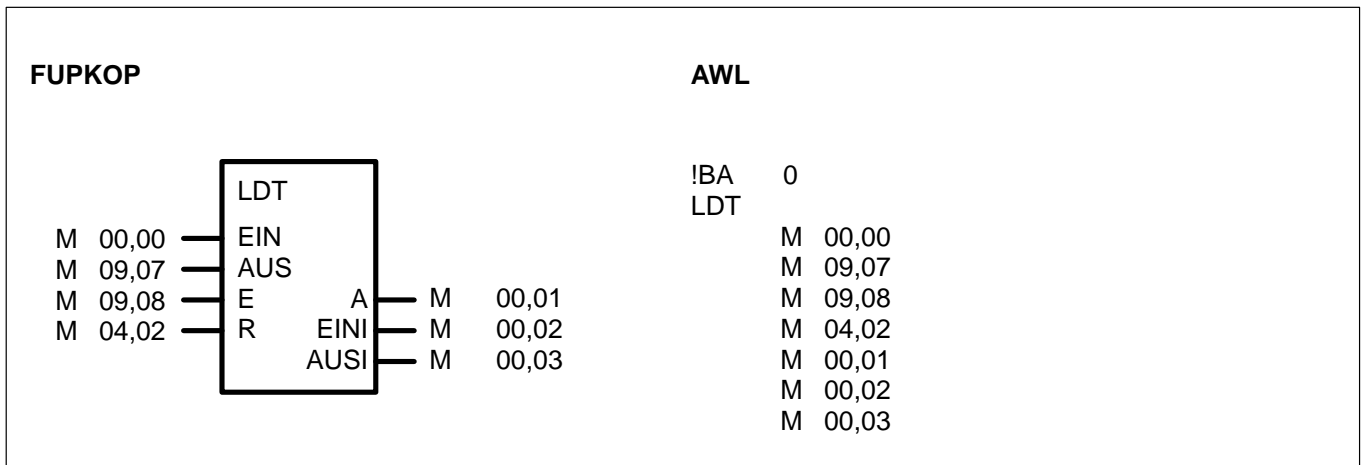
EINI BINÄR

Bei jeder 0/1-Flanke des Ausgangs A wird am Ausgang EINI ein Impuls erzeugt. Die Länge des Impulses beträgt einen Programmzyklus.

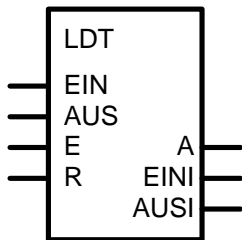
AUSI BINÄR:

Bei jeder 1/0-Flanke des Ausgangs A wird am Ausgang AUSI ein Impuls erzeugt. Die Länge des Impulses beträgt einen Programmzyklus.

Beispiel



VE-FUP-Definition

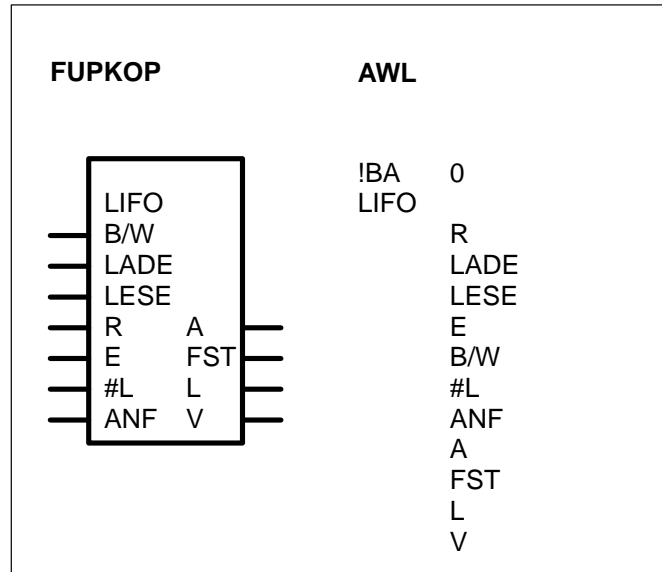


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
EIN	E	L	N	P	J	0	0
AUS	E	L	N	P	J	0	0
E	E	L	N	P	J	0	0
R	E	L	N	P	J	0	0
A	A	L	N	P	J	0	0
EINI	A	L	N	P	J	0	0
AUSI	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)			
00001	LDT						
00002		PP 0	EIN	0/1:	Einschalten von A (BINÄR)		
00003		PP 0	AUS	0/1:	Ausschalten von A (BINÄR)		
00004		PP 0	E	0/1:	Ein-/Ausschalten von A (BINÄR)		
00005		PP 0	R	Statisch:	Ausschalten von A (BINÄR)		
00006		PP 0	A	Ausgang (BINÄR)			
00007		PP 0	EINI	Einschaltimpuls (BINÄR)			
00008		PP 0	AUSI	Ausschaltimpuls (BINÄR)			

Der Funktionsbaustein realisiert einen Stapelspeicher für Binär- oder Wortdaten, aus dem die zuletzt eingeschriebenen Daten als erstes ausgelesen werden (last in/first out).



Parameter

B/W	BINÄR	E, A, M, S, K	Binärdaten/ Wortdaten
LADE	BINÄR	E, A, M, S, K	Lade LIFO
LESE	BINÄR	E, A, M, S, K	Lese LIFO
R	BINÄR	E, A, M, S, K	Reset LIFO, 0/1-Flanke
E	BINÄR	E, A, M, S, K	Eingabe der Daten in das LIFO
#L	DIREKTE KONSTANTE	#, #H	Anzahl Speicherplätze (Byte)
ANF	BINÄR	E, A, M, K	Anfang des LIFO im Merkerbereich
A	BINÄR	E, M	Ausgabe der Daten aus dem LIFO
FST	WORT	AW, MW	Füllstand des LIFO
L	BINÄR	A, M	LIFO leer
V	BINÄR	A, M	LIFO voll

VE-Daten

Laufzeit:

Grundlaufzeit: 64 µs

Zusätzliche Laufzeit: 0–114 µs je nach durchzuführender Funktion

Aktualisierung der Ausgänge: nein; Empfehlung: Ausgang direkt an Merker zuweisen oder Zwischenmerker global vergeben; siehe Griff 3, Kapitel 2.5.3

Anzahl der Vergangenheitswerte: 3 Worte

Verfügbar ab: ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein realisiert einen Stapelspeicher für Binär- oder Wortdaten, aus dem die zuletzt eingeschriebenen Daten als erstes ausgelesen werden (last in/first out).

Die Ein- und Ausgänge sind weder doppelbar noch negierbar/invertierbar.

B/W BINÄR

Der Eingang B/W dient dazu, Binär- oder Wort-Verarbeitung vorzuzählen (LIFO für Binärdaten oder Wortdaten):

B/W = 0 → Wort-Verarbeitung
B/W = 1 → Binär-Verarbeitung

Die Binär- oder Wortverarbeitung wird einmal festgelegt und darf während des Betriebs des LIFOs nicht geändert werden.

LADE BINÄR

Mit einem 1-Signal am Eingang LADE wird der am Eingang E anstehende Wert in den nächsten freien Platz des LIFOs übernommen. Ist das LIFO "voll" und es liegt ein "Lade"-Signal an, so wird der neue Wert *nicht* eingelesen. Erst, wenn wieder ein Wert ausgelesen wurde, kann ein neuer Wert eingelesen werden. Dieser wird dann in den ersten Platz des LIFOs übernommen.

LESE BINÄR

Ein 1-Signal am Eingang LESE bewirkt eine Ausgabe des aktuellen LIFO-Wertes am Ausgang A. Ist das LIFO "leer", so wird ein anstehender Lese-Auftrag ignoriert und am Ausgang A der Wert 0 ausgegeben. Der Ausgang A wird stets auf 0 gesetzt, wenn kein Lese-Auftrag vorliegt.

LADE und **LESE**

Liegt gleichzeitig ein Lade- und Lese-Auftrag vor, so wird der zu ladende Wert direkt vom Eingang E an den Ausgang A durchgegeben. Der Wert wird dabei nicht im LIFO eingespeichert. Das LIFO ändert in dieser Betriebsart seinen Füllstand nicht. Am Ausgang FST bleibt der ausgegebene Wert konstant. Ist das LIFO leer, so bleibt der Ausgang L auf 1 und der Ausgang FST wird auf 0 gesetzt.

R BINÄR

Eine 0→1-Flanke am R-Eingang bewirkt ein Rücksetzen (Reset) des Bausteins. Zuvor eingelesene Werte stehen somit nicht mehr zur Verfügung. Der Ausgang L nimmt den Wert 1, der Ausgang FST und der Ausgang A den Wert 0 an.

R und **LADE**

Steht gleichzeitig ein R- und ein LADE-Signal an, so wird zuerst das Reset und direkt anschließend das Laden ausgeführt. Das LIFO wird also zurückgesetzt und der neueinzulesende Wert wird dann wieder sofort als erster Wert im LIFO abgespeichert.

R und **LADE** und **LESE**

Der Wert am Eingang E wird direkt an den Ausgang A gegeben. Der Ausgang L wird permanent auf 1 und der Ausgang FST permanent auf 0 gesetzt.

E BINÄR/WORT

Am Eingang E wird der Wert vorgegeben, der in das LIFO zu übernehmen ist.

#L DIREKTE KONSTANTE

Am Eingang #L wird die Anzahl der erforderlichen Speicherplätze (Bytes) des LIFOs vorgegeben. Die Anzahl wird als direkte Konstante angegeben und ergibt sich aus folgender Formel:

BINÄR-Daten:

#L = Anzahl der abzulegenden Binär-Werte

WORT-Daten:

#L = 2 * Anzahl der abzulegenden Wortwerte

Z. B.: BINÄR-Daten: 3 Werte → #L = 3

WORT-Daten: 3 Werte → #L = 6

Eine Plausibilitätsüberwachung überprüft die Angabe der LIFO-Länge auf den Wert 0 und bei Wortverarbeitung auch auf ungerade Byteangabe. Ist die Angabe am Eingang #L unkorrekt, so geht das LIFO in den Initialzustand (wie nach R).

ANF BINÄR/WORT

Am Eingang ANF wird die LIFO-Speicheranfangsadresse als Binär- oder Wort-Merker vorgegeben. Das LIFO beginnt beim angegebenen Merker.

A BINÄR/WORT

Beim Lesen des LIFOs steht am Ausgang A der aktuelle Wert zur Verfügung. Liegt *kein* Lese-Auftrag vor, so wird der Wert 0 ausgegeben.

FST WORT

Der Ausgang FST zeigt zu jedem Zeitpunkt den aktuellen Füllstand des LIFOs an. Der Füllstand ist die Anzahl der im LIFO gespeicherten Binär- oder Wort-Werte.

L BINÄR

Der Ausgang L zeigt an, ob das LIFO leer ist.
L = 0 → LIFO ist nicht leer
L = 1 → LIFO ist leer

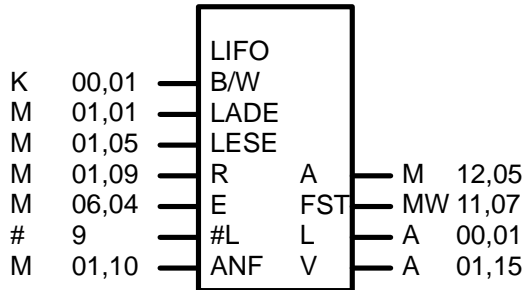
V BINÄR

Der Ausgang V zeigt an, ob das LIFO voll ist.
V = 0 → LIFO ist nicht voll
V = 1 → LIFO ist voll
Ist das LIFO voll, so kann kein Wert mehr eingelesen werden. Erst, wenn ein Wert ausgelesen wurde, ist wieder ein Einlesen möglich. Dies geschieht dann wieder ab dem Anfang des LIFOs.

Beispiel

LIFO für 9 Binär-Daten, abgelegt ab M 01,10

FUPKOP



AWL

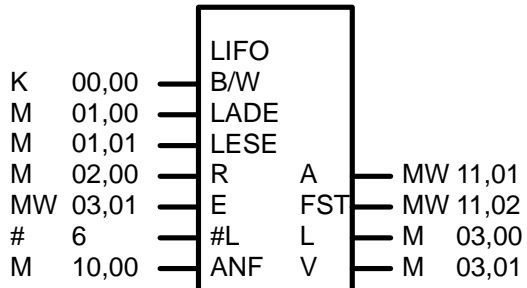
```

IBA      0
LIFO
M 01,09
M 01,01
M 01,05
M 06,04
K 00,01
# 9
M 01,10
M 12,05
MW 11,07
A 00,01
A 01,15

```

LIFO für 3 Wort-Daten, abgelegt ab MW 10,00

FUPKOP



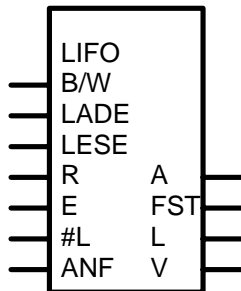
AWL

```

IBA      0
LIFO
M 02,00
M 01,00
M 01,01
MW 03,01
K 00,00
# 6
MW 10,00
MW 11,01
MW 11,02
M 03,00
M 03,01

```

VE-FUP-Definition

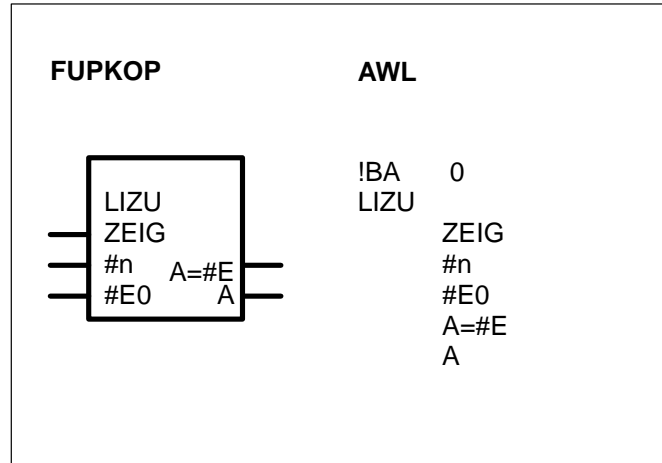


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
B/W	E	L	N	P	J	0	0
LADE	E	L	N	P	J	0	0
LESE	E	L	N	P	J	0	0
R	E	L	N	P	J	0	0
E	E	X	N	P	J	0	0
#L	K	W	N	P	J	0	0
ANF	E	X	N	P	J	0	0
A	A	X	N	P	J	0	0
FST	A	W	N	P	J	0	0
L	A	L	N	P	J	0	0
V	A	L	N	P	J	0	0

VE-AWL-Definition

Nr	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00000	LIFO			
00001				
00002		PP 0	R	Eingang BINÄR (R)
00003		PP 0	LADE	Eingang BINÄR (Lade LIFO)
00004		PP 0	LESE	Eingang BINÄR (Lese LIFO)
00005		PP 0	E	Eingang BINÄR/WORT (Eingabe)
00006		PP 0	B/W	Eingang BINÄR (Binär- Wortdaten)
00007		PP 0	#L	# KONSTANTE (Anzahl Speicherplätze)
00008		PP 0	ANF	Eingang BINÄR/WORT (Anfang Speicherplätze)
00009		PP 0	A	Ausgang BINÄR/WORT (Ausgabe)
00010		PP 0	FST	Ausgang WORT (Füllstand)
00011		PP 0	L	Ausgang BINÄR (LIFO Leer)
00012		PP 0	V	Ausgang BINÄR (LIFO voll)

Der Funktionsbaustein hat an seinen Eingängen eine Liste von direkten Konstanten. Mit einem Listenzeiger wählt er aus dieser Liste eine Konstante aus und gibt sie an seinem Ausgang aus.



Parameter

ZEIG	WORT	AW, EW, MW, KW	Zeiger auf die Liste der direkten Konstanten
#n	DIREKTE KONSTANTE	#, #H	Anzahl der direkten Konstanten in der Liste
#E0	DIREKTE KONSTANTE	#, #H	Liste der direkten Konstanten; Eingang ist doppelbar
A	WORT	AW, MW	ausgewählte direkte Konstante
A=#E	BINÄR	A, M	$0 \leq \text{ZEIG} < \#n$, d.h. Zeiger im zulässigen Bereich

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	19 µs	127 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein hat an seinen Eingängen eine Liste von direkten Konstanten. Mit einem Listenzeiger wählt er aus dieser Liste eine Konstante aus und gibt sie an seinem Ausgang aus.

Die Ein- und Ausgänge sind nicht negierbar/invertierbar. Der Eingang #E0 ist doppelbar.

ZEIG WORT

Am Eingang ZEIG wird der Zeiger auf die aus der Liste auszuwählende direkte Konstante angegeben.

Es gilt die Zuordnung:

ZEIG = 0 → direkte Konstante an E0

ZEIG = 1 → direkte Konstante an E1

ZEIG = n-1 → direkte Konstante an E_{n-1}

Der Wert am Eingang ZEIG wird auf Zulässigkeit überwacht. Das Ergebnis dieser Bereichsüberprüfung wird am Ausgang A=#E signalisiert.

Erlaubter Bereich:

$0 \leq \text{ZEIG} \leq n-1$

mit n: Anzahl der Eingänge #E0 ... #E_{n-1}.

Befindet sich der Wert am Eingang ZEIG außerhalb des erlaubten Bereichs, so erfolgt auch keine Zuweisung an den Ausgang A.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl n der an den Eingängen #E0 ... #E_{n-1} projektierten direkten Konstanten angegeben. Die Angabe erfolgt als direkte Konstante.

#E0...#E_{n-1} DIREKTE KONSTANTE

Der Eingang #E0 ist doppelbar (#E0...#E_{n-1}).

An den Eingängen #E0 ... #E_{n-1} werden die direkten Konstanten angegeben, aus denen mit dem Wert am Eingang ZEIG eine ausgewählt und dem Ausgang A zugewiesen wird.

A=#E BINÄR

Am Ausgang A=#E wird angezeigt, ob der Wert des Listen-zeigers (Eingang ZEIG) sich im erlaubten Bereich befindet.

Erlaubter Bereich:

$$0 \leq \text{ZEIG} \leq n-1$$

mit n: Anzahl der direkten Konstanten an #E0...#E_{n-1}.

Es gilt:

ZEIG im erlaubten Bereich $\rightarrow A=\#E = 1$

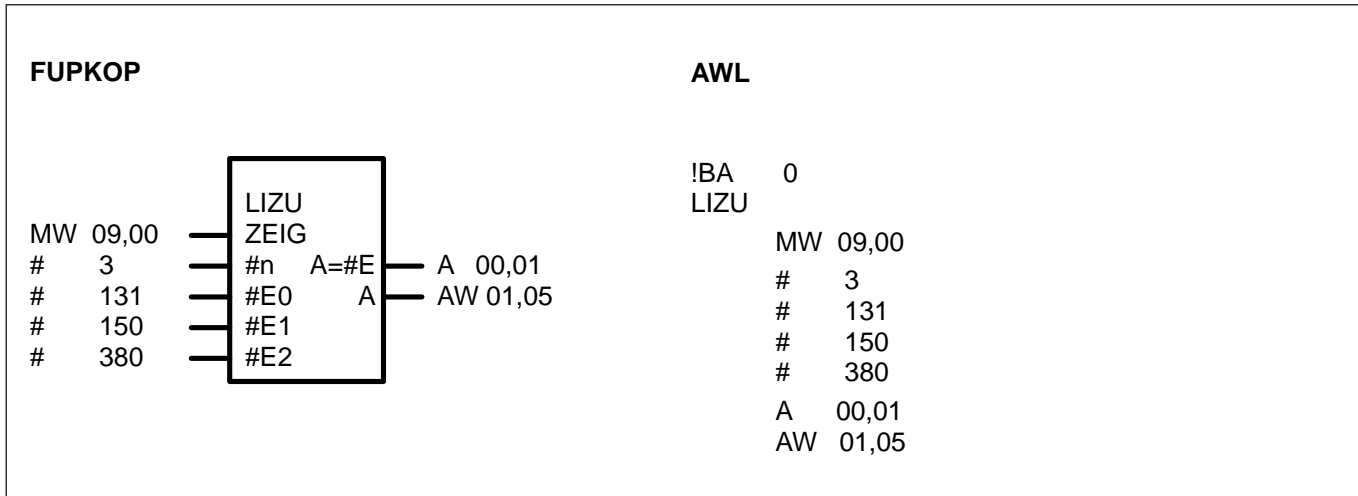
ZEIG im verbotenen Bereich $\rightarrow A=\#E = 0$

Hat der Listenzeiger einen verbotenen Wert, so erfolgt auch keine Zuweisung einer Konstanten an den Ausgang A, weil keine Konstante ausgewählt werden kann. Der Ausgang A wird in diesem Fall *nicht* aktualisiert.

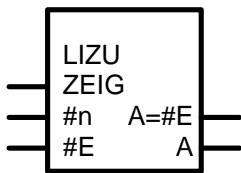
A WORT

Der Wert der ausgewählten direkten Konstanten wird dem Operanden am Ausgang A zugewiesen.

Beispiel



VE-FUP-Definition



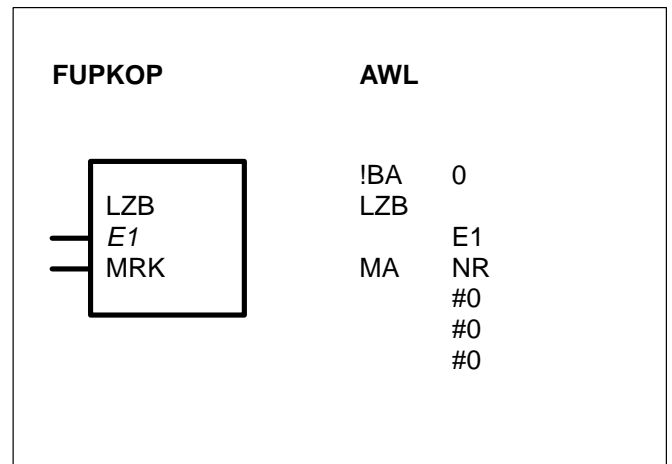
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
ZEIG	E	W	N	P	J	0	0
#n	K	W	N	P	J	0	0
#E	K	W	N	P	J	1	0
A=#E	A	L	N	P	J	0	0
A	A	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	LIZU			
00002		PP 0	ZEIG	Eingang WORT (Zeiger)
00003		PP 0	#n	# KONSTANTE (Anzahl #E's)
	[1		
00004		PP 1	#E	# KONSTANTE (Eingangswerte)
]	1		
00005		PP 0	A=#E0	Ausgang BINÄR (0 ≤ ZEIG < #n)
00006		PP 0	A	Ausgang WORT (Ausgangswert)

Der Funktionsbaustein steuert die Bearbeitung eines Programmteiles. Dieses Programmteil heißt Laufzahlblock und beginnt mit dem Funktionsbaustein LZB und endet bei der zugehörigen Zielmarke, die am Eingang MRK angegeben wird. In Abhängigkeit des Wertes des Operanden am Eingang E1 wird dieses Programmteil wie folgt bearbeitet:

- E1 = 0: Programmteil wird nicht bearbeitet
- E1 = 1: Programmteil wird bei jedem Zyklus bearbeitet
- E1 = 2: Programmteil wird bei jedem 2-ten Zyklus bearbeitet.
- E1 = n: Programmteil wird bei jedem n-ten Zyklus bearbeitet.



Parameter

E1	WORT	EW, MW, AW, KW	Laufzahl
MRK	SPEZIAL	MRK	Zielmarke

VE-Daten

Laufzeit:	
Grundlaufzeit:	29 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	entfällt
Anzahl der Vergangenheitswerte:	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein steuert die Bearbeitung eines Programmteiles. Dieses Programmteil heißt Laufzahlblock und beginnt mit dem Funktionsbaustein LZB und endet bei der zugehörigen Zielmarke, die am Eingang MRK angegeben wird. In Abhängigkeit des Wertes des Operanden am Eingang E1 wird dieses Programmteil wie folgt bearbeitet:

- E1 = 0: Programmteil wird nicht bearbeitet
- E1 = 1: Programmteil wird bei jedem Zyklus bearbeitet
- E1 = 2: Programmteil wird bei jedem 2-ten Zyklus bearbeitet.
- E1 = n: Programmteil wird bei jedem n-ten Zyklus bearbeitet.

FUPKOP

Bezeichnung der Zielmarke: MRK Nr
 Dabei ist: MRK Schlüsselwort
 0 ≤ Nr ≤ 999 Nummer der Marke

AWL

Bezeichnung der Zielmarke: MA Nr
 Dabei ist: MA Schlüsselwort
 0 ≤ Nr ≤ 999 Nummer der Marke

Die SPS errechnet automatisch :

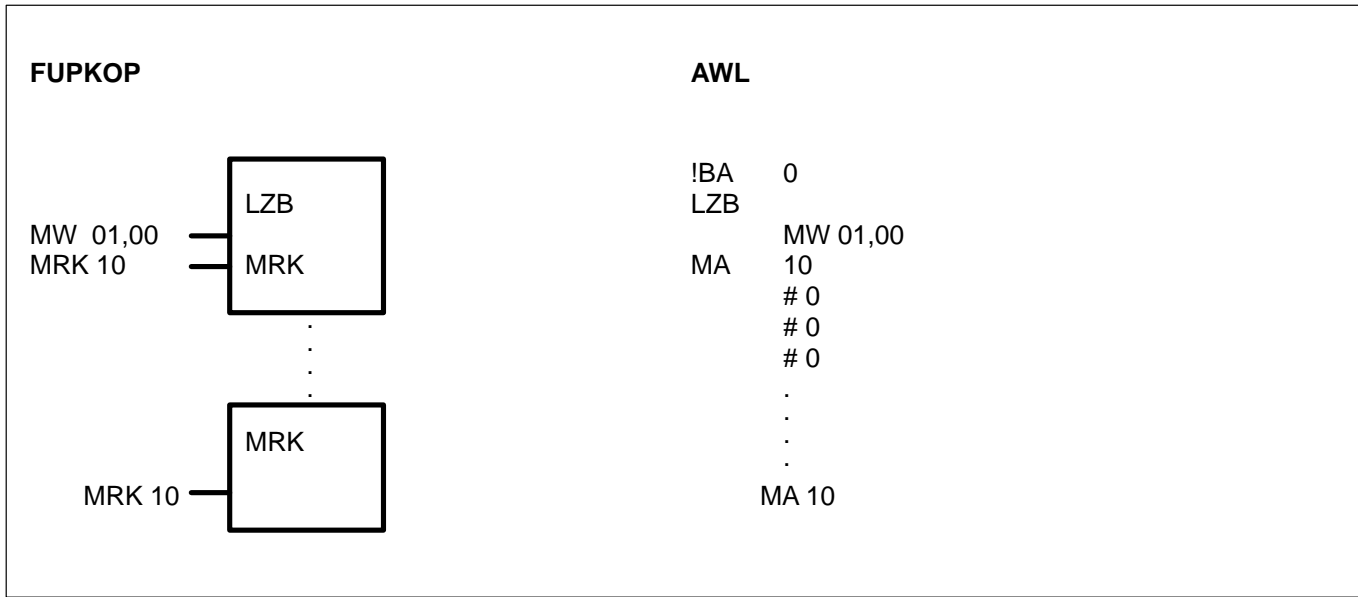
- Die Anzahl der zu überspringenden Vergangenheitswerte
- Die Adresse der Zielmarke
- Den Zeiger auf den eigenen Vergangenheitswert

Die drei errechneten Werte werden von der SPS an den dafür vorgesehenen Stellen des LZB-Bausteins eingetragen. Bei Projektierung in AWL kann der Anwender diese drei Programmworte prinzipiell mit beliebigem Inhalt füllen. Es wird aber vorgeschlagen hier den Zahlenwert 0 einzutragen.

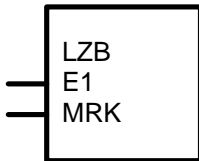
Vorsicht:

Die drei Programmworte sollten nicht mit NOP's beschrieben werden, da diese beim Optimieren des Anwenderprogramms entfernt werden.

Beispiel



VE-FUP-Definition

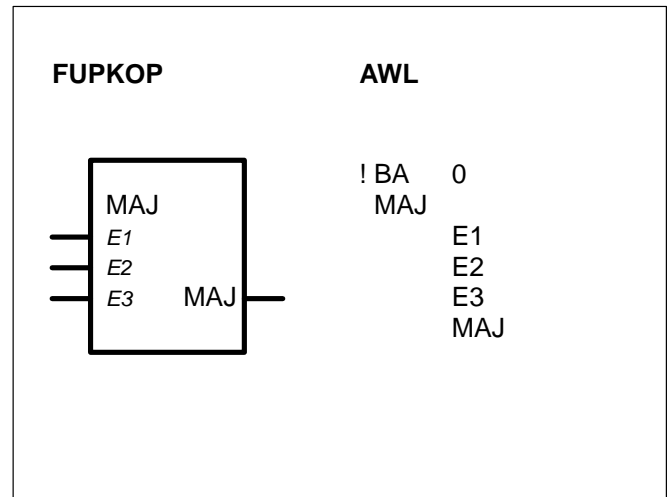


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
MRK	S	S	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	LZB			
00002		PP 0	E1	Eingang WORT (Laufzahl)
00003	MA	PP 0	MRK	Zielmarke
00004		# 0		
00005		# 0		
00006		# 0		

Das Verknüpfungselement realisiert ein MAJORITÄTS-Glied.



Parameter

E1	BINÄR	E, M, A, S, K	Operand 1
E2	BINÄR	E, M, A, S, K	Operand 2
E3	BINÄR	E, M, A, S, K	Operand 3
MAJ	BINÄR	M, A	Ergebnis

VE-Daten

Laufzeit:		
Grundlaufzeit:		36 µs
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

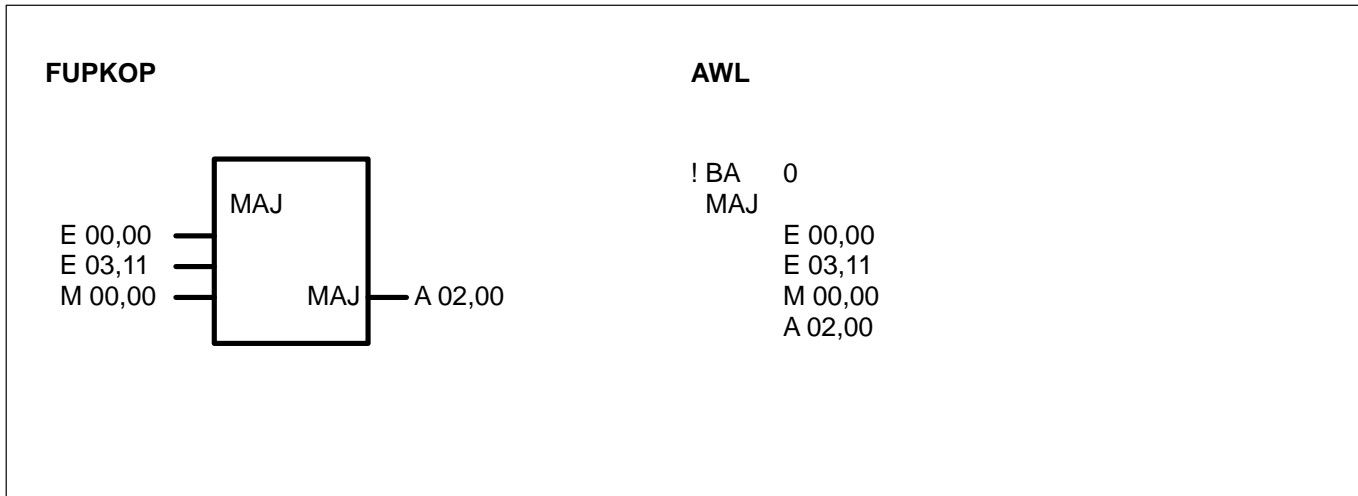
Beschreibung

Das Verknüpfungselement realisiert ein MAJORITÄTS-Glied.

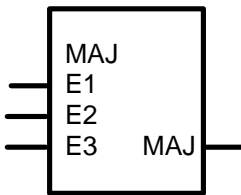
Haben mindestens 2 der 3 binären Operanden am Eingang E1, E2 und E3 den Zustand 1, so wird dem binären Operanden am Ausgang MAJ der Zustand 1 zugewiesen. Ist dies nicht der Fall, wird Zustand 0 zugewiesen.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar.

Beispiel



VE-FUP-Definition



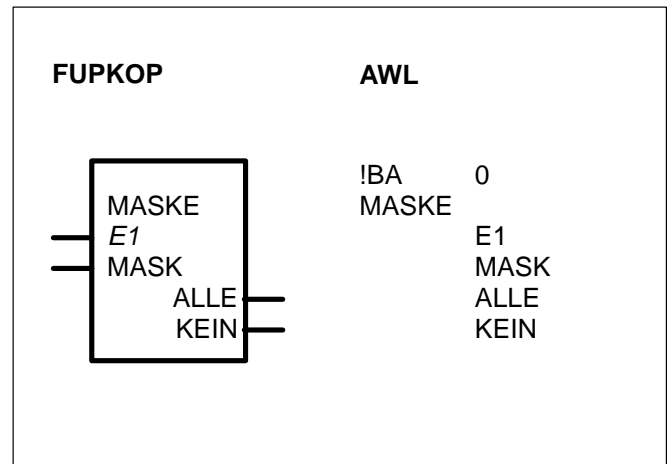
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	L	N	P	N	0	0
E2	E	L	N	P	N	0	0
E3	E	L	N	P	N	0	0
MAJ	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MAJ			
00002		PP 0	E1	Eingang BINÄR
00003		PP 0	E2	Eingang BINÄR
00004		PP 0	E3	Eingang BINÄR
00005		PP 0	MAJ	Ausgang BINÄR (Majorität)

Die einzelnen Bits des Operanden am Eingang E1 werden mit den Bits des Operanden am Eingang MASK verglichen.

Das Ergebnis des Vergleichs wird an den Ausgängen ALLE und KEIN signalisiert.



Parameter

E1	WORT	EW, AW, MW, KW	Eingangswert
MASK	WORT	EW, AW, MW, KW	Maske
ALLE	BINÄR	A, M	Alle Bits stimmen überein
KEIN	BINÄR	A, M	Kein Bit stimmt überein

VE-Daten

Laufzeit:	
Grundlaufzeit:	35 – 38 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Die einzelnen Bits des Operanden am Eingang E1 werden mit den Bits des Operanden am Eingang MASK verglichen.

Das Ergebnis des Vergleichs wird an den Ausgängen ALLE und KEIN signalisiert.

Sind beim Operanden am Eingang E1 mindestens **alle** Bits gesetzt, die beim Operanden am Eingang MASK gesetzt sind, so gilt für die Ausgänge:

ALLE = 1
KEIN = 0

Ist beim Operanden am Eingang E1 **keines** der Bits gesetzt, die beim Operanden am Eingang MASK gesetzt sind, so gilt für die Ausgänge:

ALLE = 0
KEIN = 1

Ist beim Operanden am Eingang E1 nur **ein Teil** der Bits gesetzt, die beim Operanden am Eingang MASK gesetzt sind, so gilt für die Ausgänge:

ALLE = 0
KEIN = 0

Die Ein- und Ausgänge sind weder doppelbar noch negierbar noch invertierbar.

Beispiel

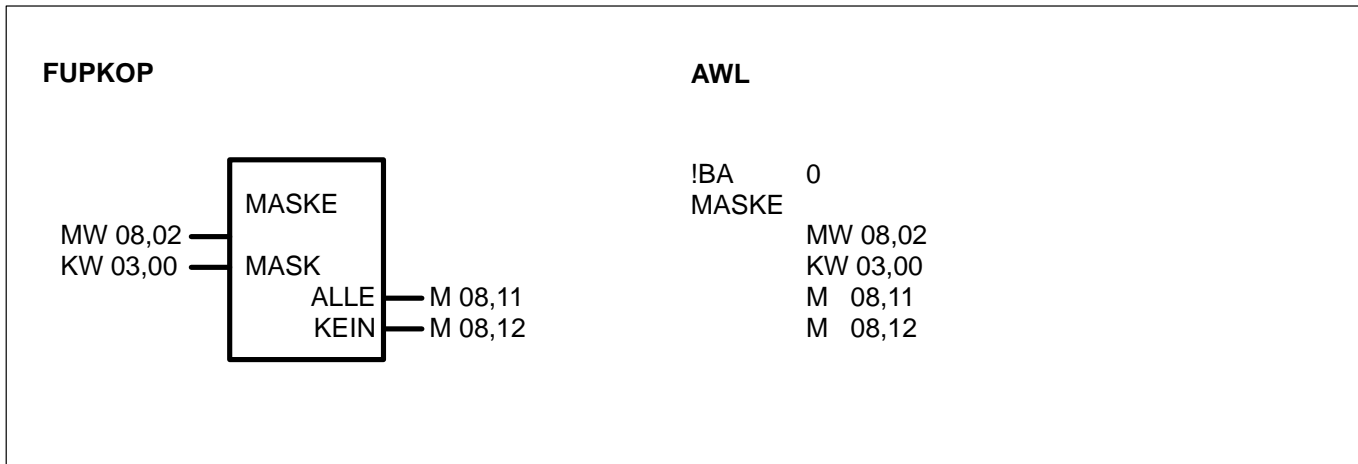
E1 : x1111xx11xxxx11x ALLE = 1
MASK : 0111100110000110 KEIN = 0

E1 : 00xxxxx0xxx0xxxx ALLE = 0
MASK : 1100000100010000 KEIN = 1

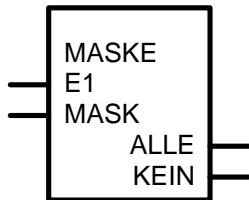
E1 : 1x1x1xxx10xxxx10 ALLE = 0
MASK : 0010100011000011 KEIN = 0

X: Diese Bits können beliebige Werte haben (don't care)

Beispiel



VE-FUP-Definition



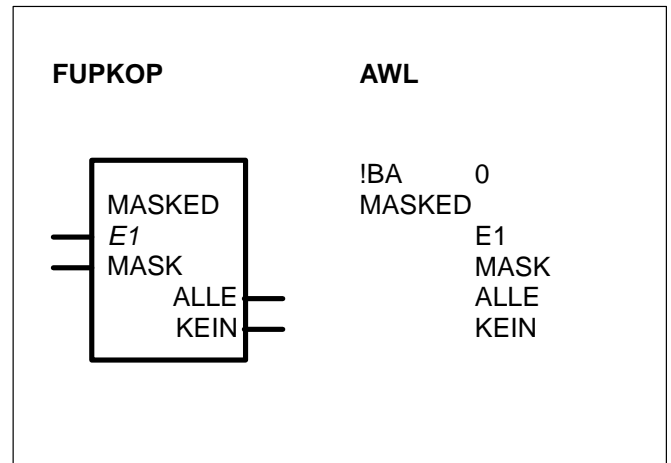
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
MASK	E	W	N	P	J	0	0
ALLE	A	L	N	P	J	0	0
KEIN	A	L	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MASKE			
00002		PP 0	E1	Eingang WORT (Eingangswert)
00003		PP 0	MASK	Eingang WORT (Maske)
00004		PP 0	ALLE	Ausgang BINÄR (ALLE Bit stimmen)
00005		PP 0	KEIN	Ausgang BINÄR (KEIN Bit stimmt)

Die einzelnen Bits des Operanden am Eingang E1 werden mit den Bits des Operanden am Eingang MASK verglichen.

Das Ergebnis des Vergleichs wird an den Ausgängen ALLE und KEIN signalisiert.



Parameter

E1	DOPPELWORT	MD, KD	Eingangswert
MASK	DOPPELWORT	MD, KD	Maske
ALLE	BINÄR	A, M	Alle Bits stimmen überein
KEIN	BINÄR	A, M	Kein Bit stimmt überein

VE-Daten

Laufzeit:	
Grundlaufzeit:	42 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Die einzelnen Bits des Operanden am Eingang E1 werden mit den Bits des Operanden am Eingang MASK verglichen.

Das Ergebnis des Vergleichs wird an den Ausgängen ALLE und KEIN signalisiert.

Sind beim Operanden am Eingang E1 mindestens *alle* Bits gesetzt, die beim Operanden am Eingang MASK gesetzt sind, so gilt für die Ausgänge:

ALLE = 1
KEIN = 0

Ist beim Operanden am Eingang E1 *keines* der Bits gesetzt, die beim Operanden am Eingang MASK gesetzt sind, so gilt für die Ausgänge:

ALLE = 0
KEIN = 1

Ist beim Operanden am Eingang E1 nur *ein Teil* der Bits gesetzt, die beim Operanden am Eingang MASK gesetzt sind, so gilt für die Ausgänge:

ALLE = 0
KEIN = 0

Die Ein- und Ausgänge sind weder doppelbar noch negierbar/invertierbar.

Beispiel

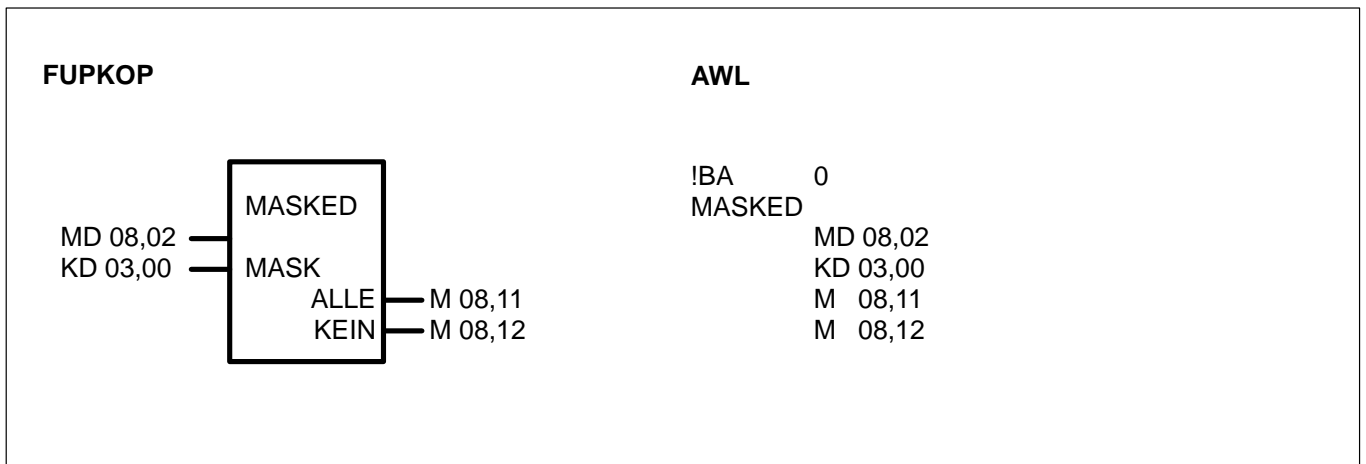
```
E1      : X1111XX11XXXX11X  X1XXXXX1XXX1XXXX  ALLE = 1
MASK    : 0111100110000110  0100000100010000  KEIN = 0
```

```
E1      : X0000XX00XXXX00X  X0XXXXX0XXX0XXXX  ALLE = 0
MASK    : 0111100110000110  0100000100010000  KEIN = 1
```

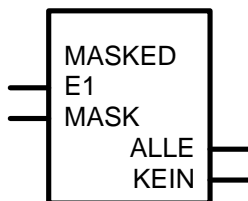
```
E1      : X1011XX10XXXX11X  X1XXXXX0XXX1XXXX  ALLE = 0
MASK    : 0111100110000110  0100000100010000  KEIN = 0
```

X: Diese Bits können beliebige Werte haben (don't care).

Beispiel



VE-FUP-Definition

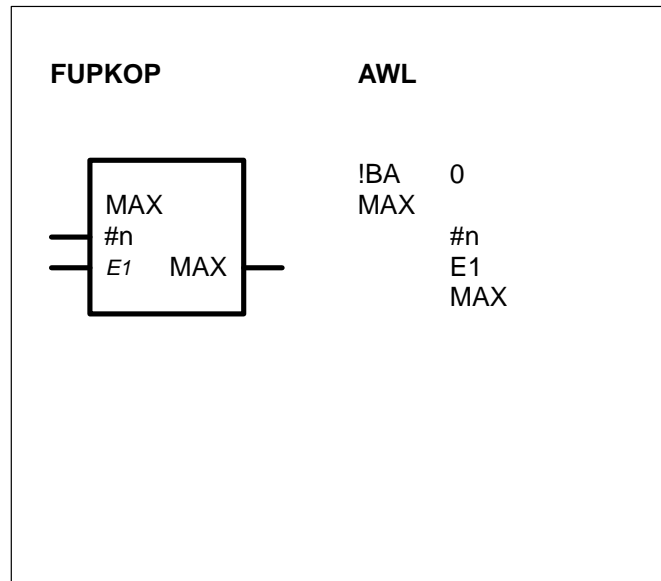


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
MASK	E	D	N	P	J	0	0
ALLE	A	L	N	P	J	0	0
KEIN	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MASKED			
00002		PP 0	E1	Eingang DOPPELWORT (Eingangswert)
00003		PP 0	MASK	Eingang DOPPELWORT (Maske)
00004		PP 0	ALLE	Ausgang BINÄR (ALLE Bit stimmen)
00005		PP 0	KEIN	Ausgang BINÄR (KEIN Bit stimmt)

Der Funktionsbaustein ermittelt aus n Operanden den Maximalwert und weist diesen dem Ausgang zu.



Parameter

#n	DIREKTE KONSTANTE	#, #H	Anzahl der Eingänge
E1	WORT	EW, MW, AW, KW	1. Eingangswert; Eingang ist doppelbar
MAX	WORT	AW, MW	Ausgang (Maximalwert)

VE–Daten

Serie 90

Serie 30, 40, 50

Laufzeit:			
Grundlaufzeit:		22 µs	74 µs
Zusätzliche Laufzeit:		9 µs, pro projektiertem Eingang E1	64 µs, pro projektiertem Eingang E1
Aktualisierung der Ausgänge:	ja	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0,	07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein ermittelt aus n Operanden den Maximalwert und weist diesen dem Ausgang zu.

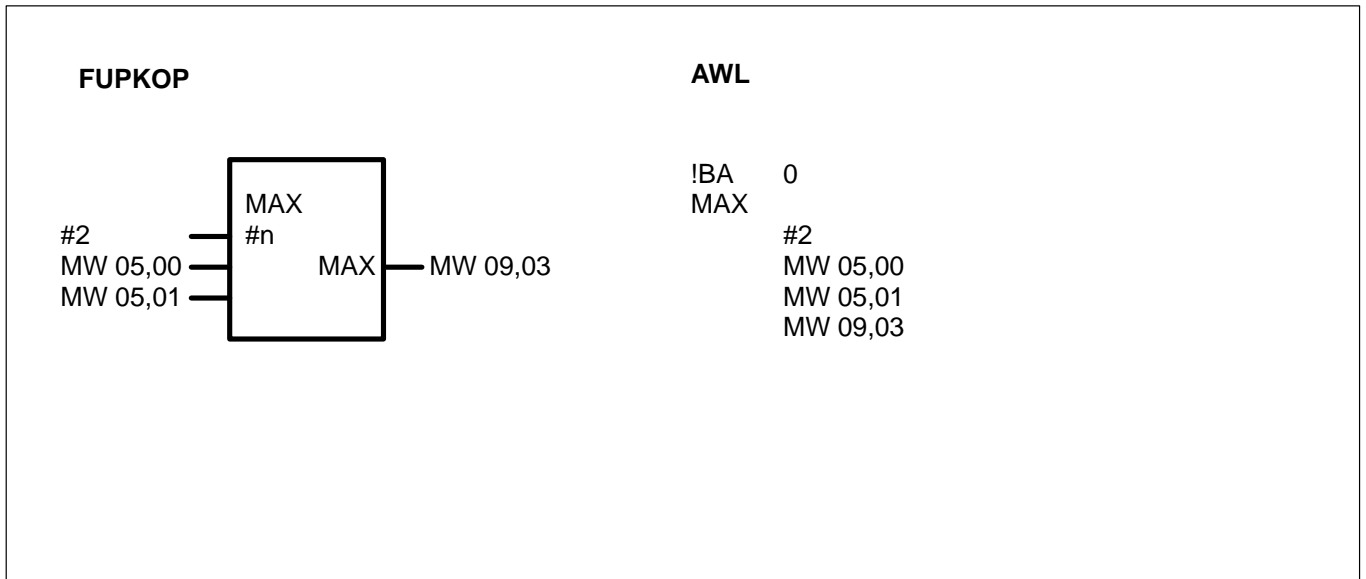
#n DIREKTE KONSTANTE (#, #H)
Am Eingang #n wird die Anzahl der Operanden angegeben, aus denen der Maximalwert ermittelt werden soll. Die Angabe erfolgt als direkte Konstante. Es gilt n > 0.

E1 WORT
Der Eingang E1 ist so oft zu doppelnd, bis die am Eingang #n angegebene Zahl von Eingängen vorhanden ist. Aus den Werten der Operanden an diesen Eingängen wird der Maximalwert ermittelt und dem Ausgang MAX zugewiesen.

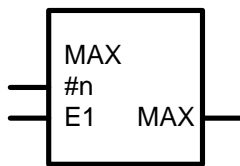
MAX WORT
Am Ausgang MAX steht der Maximalwert aus den n Eingangswerten zur Verfügung.

Der Eingang E1 ist doppelbar.
Der Ausgang ist weder doppelbar noch negierbar.

Beispiel



VE-FUP-Definition

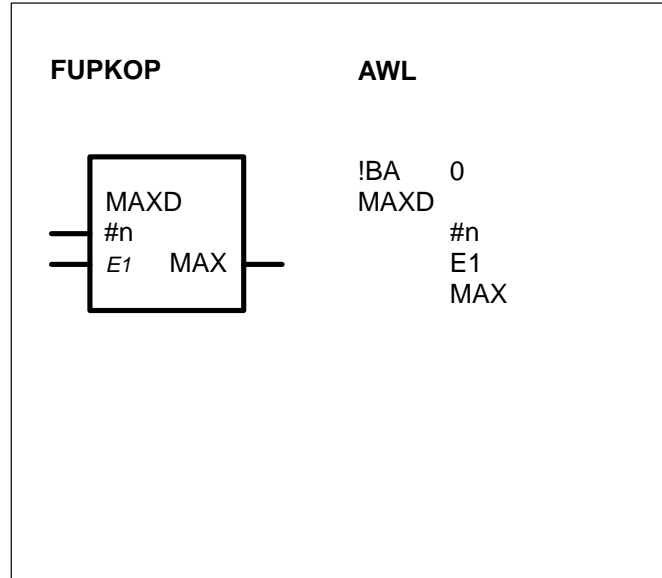


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
E1	E	W	N	P	N	1	0
MAX	A	W	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MAX			
00002	[PP 0	#n	# KONSTANTE (Anzahl Eingänge)
		1		
00003	E1	PP 1	E1	Eingang WORT
]	1		
00004		PP 0	MAX	Ausgang WORT (Maximalwert)

Der Funktionsbaustein ermittelt aus n Operanden den Maximalwert und weist diesen dem Ausgang zu.



Parameter

#n	DIREKTE #, #H KONSTANTE	Anzahl der Eingänge
E1	DOPPELWORT MD, KD	1. Eingangswert; Eingang ist doppelbar
MAX	DOPPELWORT MD	Ausgang (Maximalwert)

VE-Daten

Laufzeit:	
Grundlaufzeit:	29 µs
Zusätzliche Laufzeit:	13 µs pro projektiertem Eingang E1
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

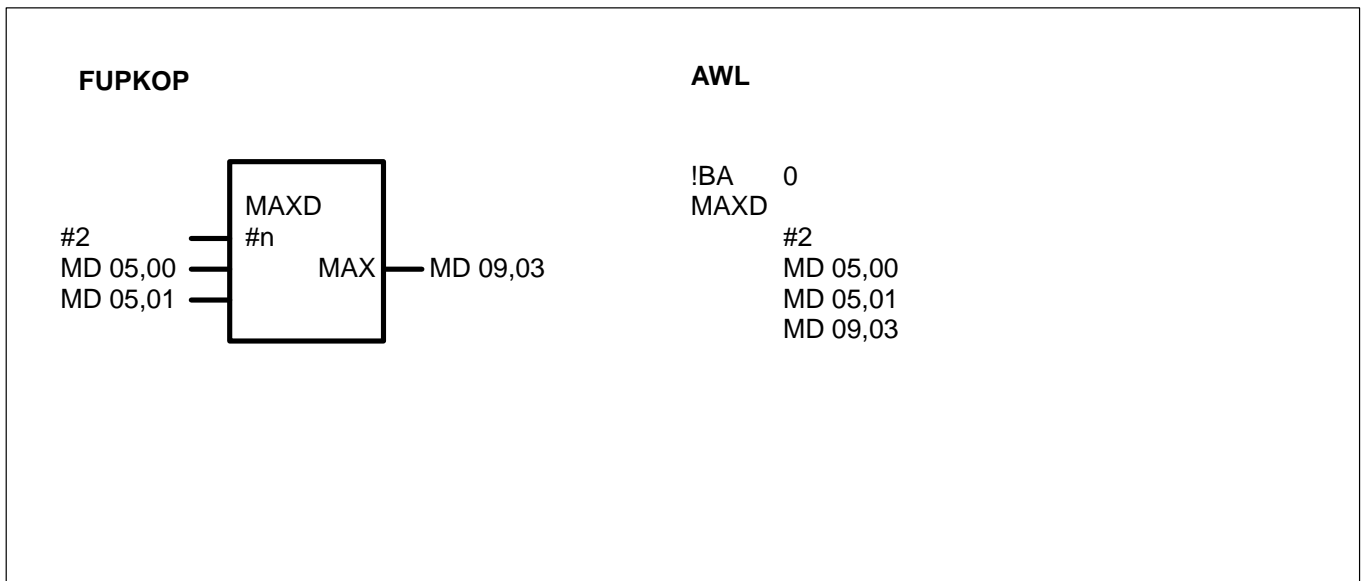
Der Funktionsbaustein ermittelt aus n Operanden den Maximalwert und weist diesen dem Ausgang zu.

#n DIREKTE KONSTANTE (#, #H)
Am Eingang #n wird die Anzahl der Operanden angegeben, aus denen der Maximalwert ermittelt werden soll. Die Angabe erfolgt als direkte Konstante. Es gilt n > 0.

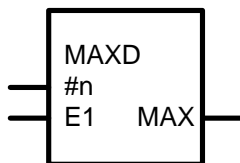
E1 DOPPELWORT
Der Eingang E1 ist so oft zu doppelnd, bis die am Eingang #n angegebene Zahl von Eingängen vorhanden ist. Aus den Werten der Operanden an diesen Eingängen wird der Maximalwert ermittelt und dem Ausgang MAX zugewiesen.

MAX DOPPELWORT
Am Ausgang MAX steht der Maximalwert aus den n Eingangswerten zur Verfügung.
Der Eingang E1 ist doppelbar.
Der Ausgang ist weder doppelbar noch negierbar.

Beispiel



VE-FUP-Definition

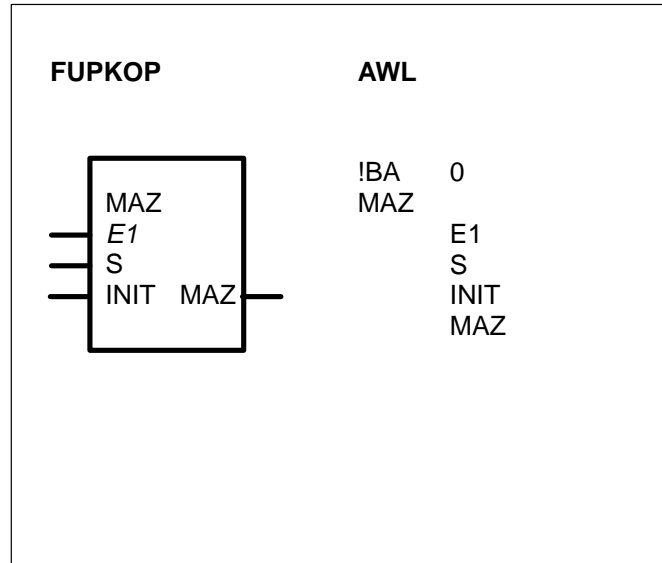


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
E1	E	D	N	P	N	1	0
MAX	A	D	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MAXD			
00002	[PP 0	#n	# KONSTANTE (Anzahl Eingänge)
		1		
00003]	PP 1	E1	Eingang DOPPELWORT
		1		
00004		PP 0	MAX	Ausgang DOPPELWORT (Maximalwert)

Der Funktionsbaustein ermittelt aus dem zeitlichen Verlauf eines Signals dessen bis zum aktuellen Zeitpunkt aufgetretenen Maximalwert.



Parameter

E1	WORT	EW, MW, AW, KW	Eingangswert, dessen zeitliches Maximum ermittelt wird
S	BINÄR	E, M, A, S, K	Setzeingang
INIT	WORT	EW, MW, AW, KW	Initialwert
MAZ	WORT	AW, MW	Maximalwert

VE-DATEN

Laufzeit:	
Grundlaufzeit:	36 ... 40 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	2 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein ermittelt aus dem zeitlichen Verlauf eines Signals dessen bis zum aktuellen Zeitpunkt aufgetretenen Maximalwert.

Der Wert des Operanden am Eingang E1 wird mit dem bisher aufgetretenen Maximalwert verglichen. Ist der Eingangswert E1 größer als das bisher aufgetretenen Maximum, so ist der Eingangswert der neue Maximalwert und wird dem Operanden am Ausgang MAZ zugewiesen.

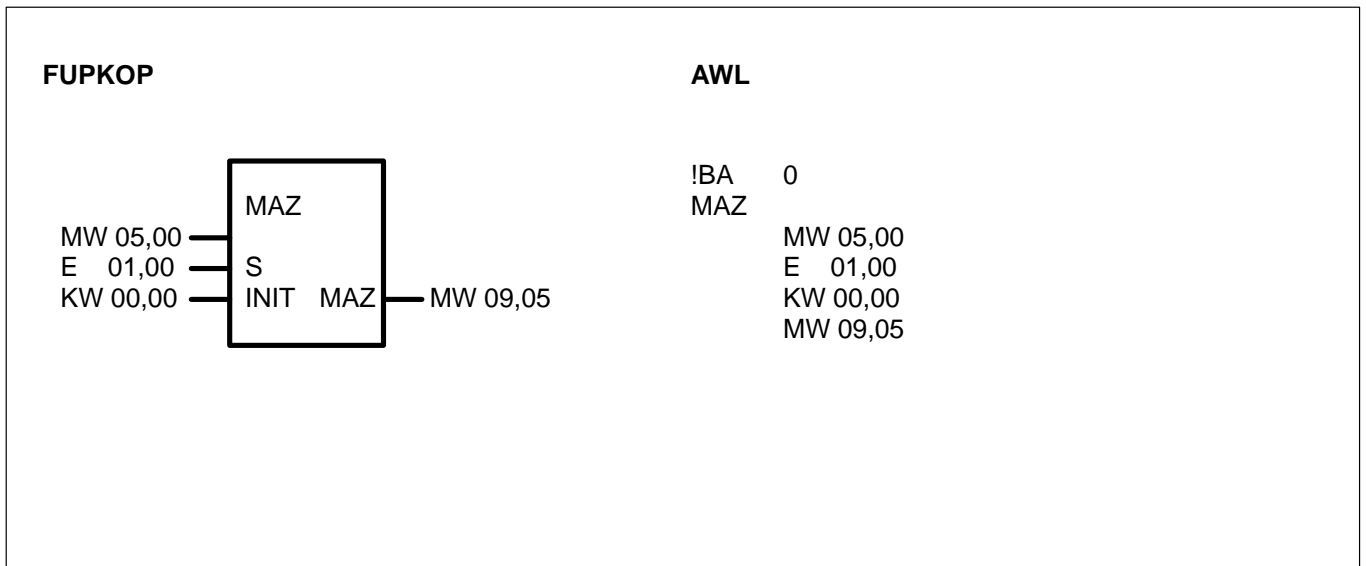
Ist der Eingangswert E1 kleiner als der bisher aufgetretene Maximalwert, so wird dem Ausgang der bisherige Maximalwert zugewiesen.

Mit der 0→1-Flanke am binären Eingang S wird der Ausgang MAZ auf den Wert des Operanden am Eingang INIT (Initialwert) gesetzt.

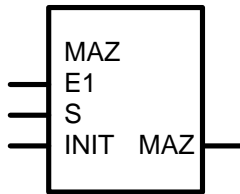
Es gilt:
 $E1 < MAZ \rightarrow MAZ = MAZ$
 $E1 \geq MAZ \rightarrow MAZ = E1$
 $S = 0 \rightarrow 1 \text{-Flanke} \rightarrow MAZ = INIT$

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar/invertierbar.

Beispiel



VE-FUP-Definition

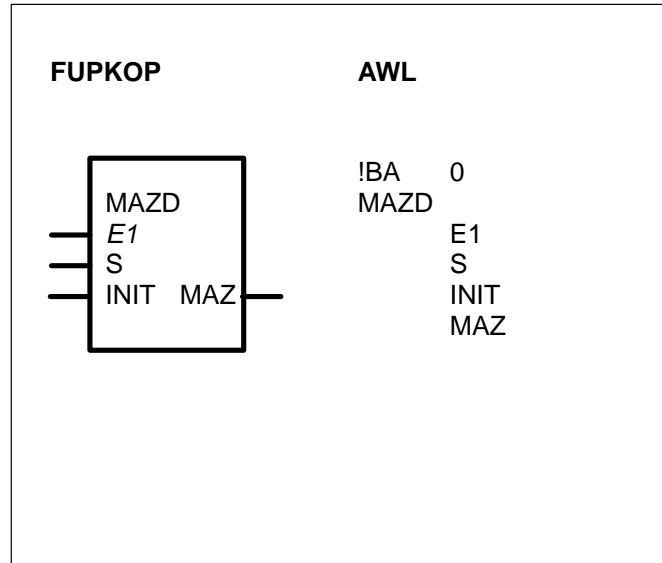


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
S	E	L	N	P	J	0	0
INIT	E	W	N	P	J	0	0
MAZ	A	W	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MAZ			
00002		PP 0	E1	Eingang WORT (Eingangswert)
00003		PP 0	S	Eingang BINÄR (Setzeingang)
00004		PP 0	INIT	Eingang WORT (Initialwert)
00005		PP 0	MAZ	Ausgang WORT(Maximalwert)

Der Funktionsbaustein ermittelt aus dem zeitlichen Verlauf eines Signals dessen bis zum aktuellen Zeitpunkt aufgetretenen Maximalwert.



Parameter

E1	DOPPELWORT	MD, KD	Eingangswert, dessen zeitliches Maximum ermittelt wird
S	BINÄR	E, M, A, S, K	Setzeingang
INIT	DOPPELWORT	MD	Initialwert
MAZ	DOPPELWORT	MD	Maximalwert

VE-DATEN

Laufzeit:	
Grundlaufzeit:	49 ... 58 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	3 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein ermittelt aus dem zeitlichen Verlauf eines Signals dessen bis zum aktuellen Zeitpunkt aufgetretenen Maximalwert.

Der Wert des Operanden am Eingang E1 wird mit dem bisher aufgetretenen Maximalwert verglichen. Ist der Eingangswert E1 größer als das bisher aufgetretene Maximum, so ist der Eingangswert der neue Maximalwert und wird dem Operanden am Ausgang MAZ zugewiesen.

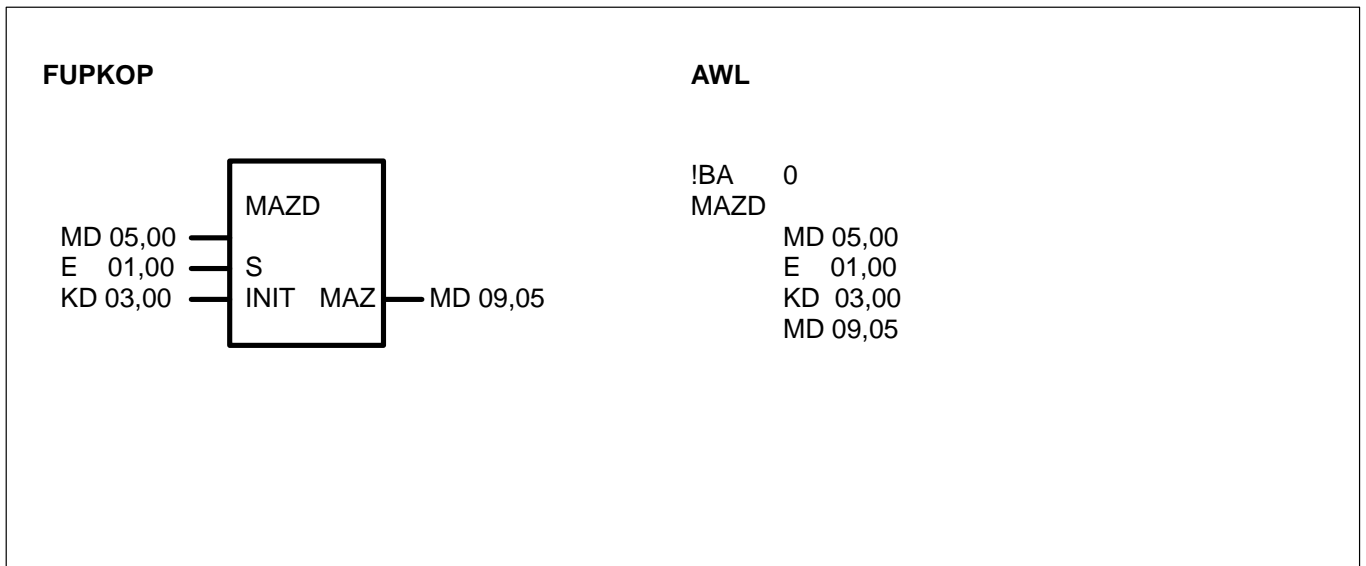
Ist der Eingangswert E1 kleiner als der bisher aufgetretene Maximalwert, so wird dem Ausgang der bisherige Maximalwert zugewiesen.

Mit der 0→1-Flanke am binären Eingang S wird der Ausgang MAZ auf den Wert des Wortoperanden am Eingang INIT (Initialwert) gesetzt.

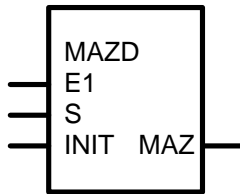
Es gilt:
 $E1 < MAZ \rightarrow MAZ = MAZ$
 $E1 \geq MAZ \rightarrow MAZ = E1$
 $S = 0 \rightarrow 1\text{-Flanke} \rightarrow MAZ = INIT$

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar/invertierbar.

Beispiel



VE-FUP-Definition

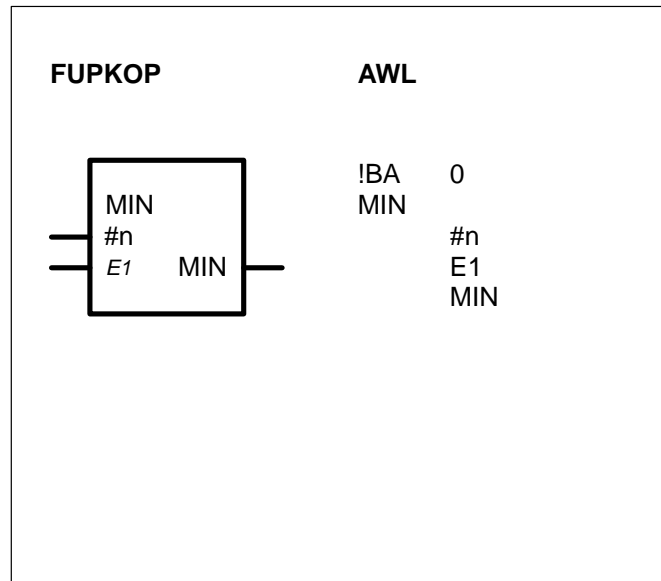


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
S	E	L	N	P	J	0	0
INIT	E	D	N	P	J	0	0
MAZ	A	D	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MAZD			
00002		PP 0	E1	Eingang DOPPELWORT (Eingangswert)
00003		PP 0	S	Eingang BINÄR (Setzeingang)
00004		PP 0	INIT	Eingang DOPPELWORT (Initialwert)
00005		PP 0	MAZ	Ausgang DOPPELWORT(Maximalwert)

Der Funktionsbaustein ermittelt aus n Operanden den Minimalwert und weist diesen dem Ausgang zu.



Parameter

#n	DIREKTE KONSTANTE	#, #H	Anzahl der Eingänge
E1	WORT	EW, MW, AW, KW	1. Eingangswert; Eingang ist doppelbar
MIN	WORT	AW, MW	Ausgang (Minimalwert)

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:	35 µs
Zusätzliche Laufzeit:	7 µs, pro projektiertem Eingang E1

74 µs	
64 µs,	pro projektiertem Eingang E1

Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

ja	
keine	
907 PC 331 / 07 KR 31 V 1.0,	
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51	

Beschreibung

Der Funktionsbaustein ermittelt aus n Operanden den Minimalwert und weist diesen dem Ausgang zu.

MIN WORT

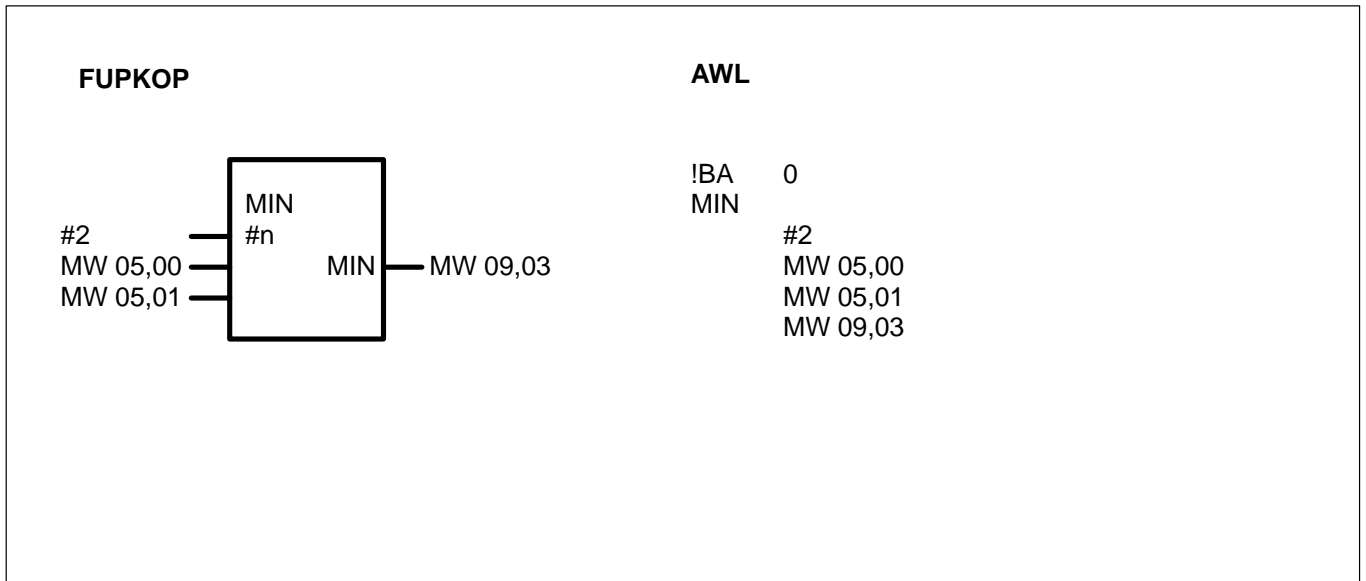
Am Ausgang MIN steht der Minimalwert aus den n Eingangs-Operanden zur Verfügung.

#n DIREKTE KONSTANTE (#, #H)
 Am Eingang #n wird die Anzahl der Operanden angegeben, aus denen der Minimalwert ermittelt werden soll. Die Angabe erfolgt als direkte Konstante.
 Es gilt n > 0.

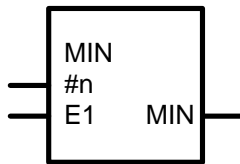
Der Eingang E1 ist doppelbar.
 Der Ausgang ist weder doppelbar noch negierbar.

E1 WORT
 Der Eingang E1 ist so oft zu doppeln, bis die am Eingang #n angegebene Zahl von Eingängen vorhanden ist. Aus den Werten der Operanden an diesen Eingängen wird der Minimalwert ermittelt und dem Ausgang MIN zugewiesen.

Beispiel



VE-FUP-Definition

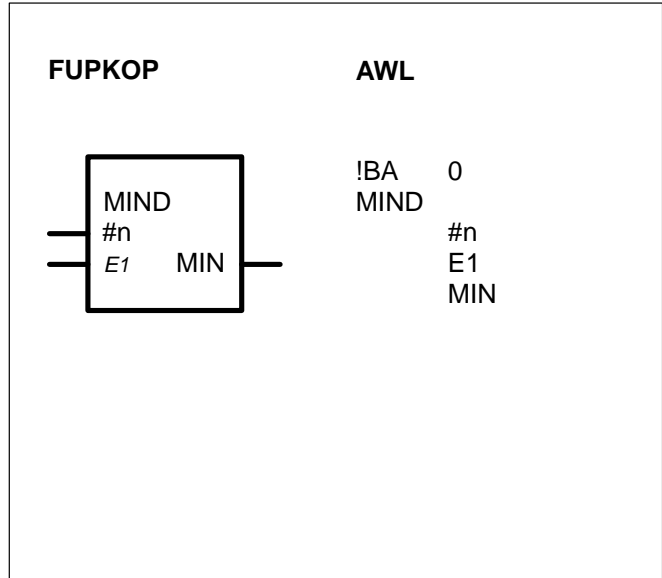


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
E1	E	W	N	P	N	1	0
MIN	A	W	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MIN			
00002	[PP 0	#n	# KONSTANTE (Anzahl Eingänge)
		1		
00003	E1	PP 1	E1	Eingang WORT
]	1		
00004		PP 0	MIN	Ausgang WORT (Minimalwert)

Der Funktionsbaustein ermittelt aus n Operanden den Minimalwert und weist diesen dem Ausgang zu.



Parameter

#n	DIREKTE #, #H KONSTANTE	Anzahl der Eingänge
E1	DOPPELWORT MD, KD	1. Eingangswert; Eingang ist doppelbar
MIN	DOPPELWORT MD	Ausgang (Minimalwert)

VE-Daten

Laufzeit:	
Grundlaufzeit:	24 µs
Zusätzliche Laufzeit:	14 µs pro projektiertem Eingang E1
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein ermittelt aus n Operanden den Minimalwert und weist diesen dem Ausgang zu.

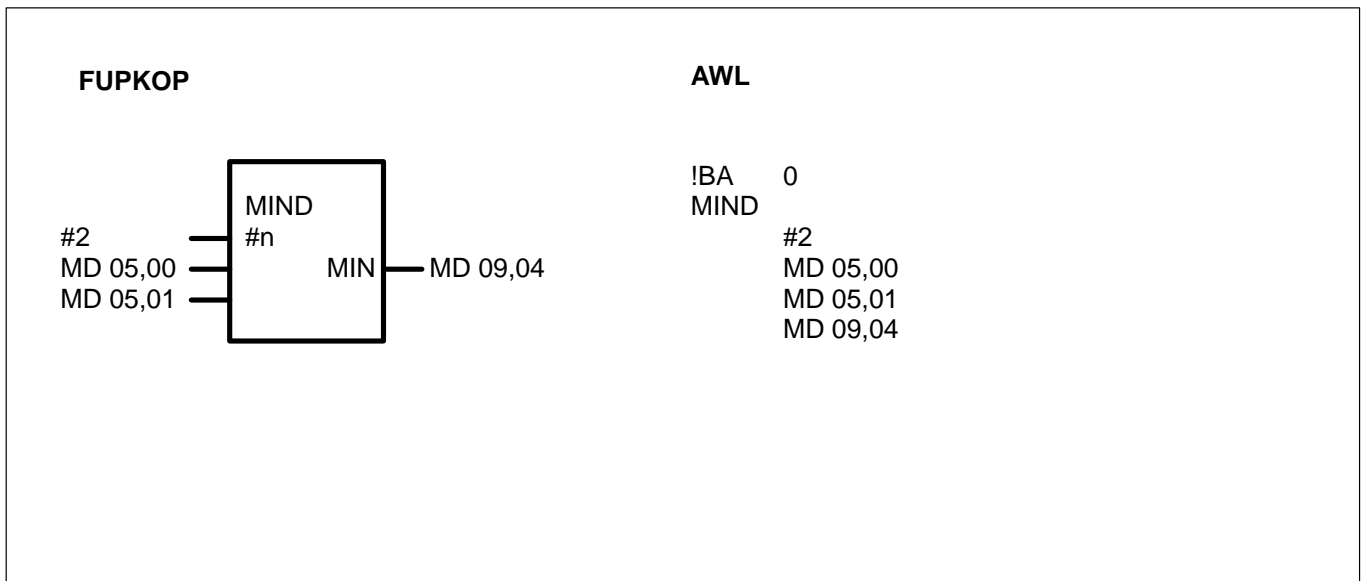
#n DIREKTE KONSTANTE (#, #H)
Am Eingang #n wird die Anzahl der Operanden angegeben, aus denen der Minimalwert ermittelt werden soll. Die Angabe erfolgt als direkte Konstante. Es gilt n > 0.

E1 DOPPELWORT
Der Eingang E1 ist so oft zu doppelnd, bis die am Eingang #n angegebene Zahl von Eingängen vorhanden ist. Aus den Werten der Operanden an diesen Eingängen wird der Minimalwert ermittelt und dem Ausgang MIN zugewiesen.

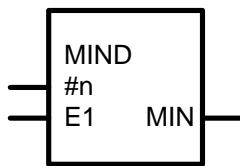
MIN DOPPELWORT
Am Ausgang MIN steht der Minimalwert aus den n Eingang-Operanden zur Verfügung.

Der Eingang E1 ist doppelbar.
Der Ausgang ist weder doppelbar noch negierbar.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
E1	E	D	N	P	N	1	0
MIN	A	D	N	P	J	0	0

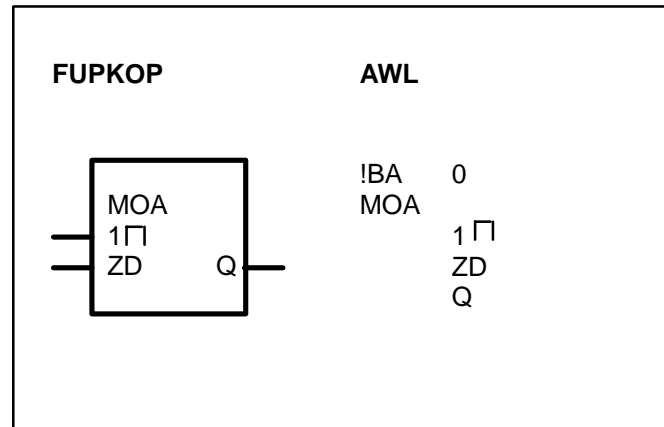
VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MIND			
00002	[PP 0	#n	# KONSTANTE (Anzahl Eingänge)
		1		
00003	E1	PP 1	E1	Eingang DOPPELWORT
]	1		
00004	MIN	PP 0	MIN	Ausgang DOPPELWORT (Minimalwert)

Eine 0–1–Flanke am Eingang 1Γ bewirkt eine 0–1–Flanke am Ausgang Q. Bleibt der Eingang 1Γ auf 1–Pegel, wird nach der Zeitdauer ZD eine 1–0–Flanke am Ausgang Q ausgegeben.

Geht der Eingang 1Γ vor Ablauf der Zeitdauer ZD wieder auf 0–Pegel zurück, so wird auch sofort der Ausgang Q wieder auf 0–Pegel gesetzt.

Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit



Parameter

1Γ	BINÄR	E, M, A, S, K	Eingangssignal
ZD	DOPPELWORT	MD, KD	Impulslänge
Q	BINÄR	M, A	Ausgangssignal

VE-Daten

Laufzeit:

Grundlaufzeit: 36 μ s; Höchstlaufzeit 600 μ s,
bei Flanken am Eingang 1Γ

Zusätzliche Laufzeit: —

Aktualisierung der Ausgänge: ja

Anzahl der Vergangenheitswerte: 1 Wort

Verfügbar ab: ABB Procontic CS31 / 907 PC 331

Serie 30, 40, 50

97 μ s / 551 μ s

ja

1 Wort

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Eine 0–1–Flanke am Eingang 1Γ bewirkt eine 0–1–Flanke am Ausgang Q. Bleibt der Eingang 1Γ auf 1–Pegel, wird nach der Zeitdauer ZD eine 1–0–Flanke am Ausgang Q ausgegeben.

Geht der Eingang 1Γ vor Ablauf der Zeitdauer ZD wieder auf 0–Pegel zurück, so wird auch sofort der Ausgang Q wieder auf 0–Pegel gesetzt.

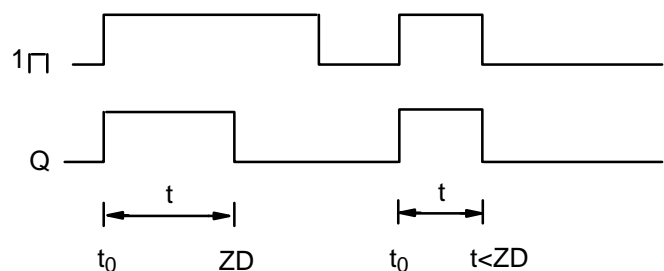
Die Zeitangabe erfolgt in Millisekunden, wobei nur ganzzahlige Vielfache von 5 ms zulässig sind (Bsp.: 5 ms, 10 ms, 500 ms, 100 000 ms ...). Angebbare Zeitbereich: 5 ms ... 24,8 Tage.

Bei der indirekten Konstanten KD wird der Wert der Impulslänge im Langtext angegeben.

Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit

Sinnvoller Bereich für ZD: > 1 Zykluszeit

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar.



Allgemeines Verhalten

- Gestartete Zeitwerke werden vom Betriebssystem der SPS bearbeitet und sind deshalb vollkommen unabhängig von der Bearbeitung des SPS-Programms. Erst nach Ablauf des Zeitwerks erfolgt eine entsprechende Meldung des Betriebssystems an den zugehörigen Zeitbaustein im SPS-Programm.
- Die Bearbeitung eines Zeitwerks im Betriebssystem der SPS wird durch folgende Befehle *nicht* beeinflusst:
 - Programm abbrechen
 - Programm starten
 - Programm stoppen
 - Programm fortsetzen

D. h. ist ein Zeitwerk gestartet, so wird es im Betriebssystem der SPS auch dann weiter bearbeitet, wenn das zugehörige SPS-Programm abgebrochen, wieder gestartet oder gestoppt und wieder fortgesetzt wird.

Initialisierung

Die Zeitwerke werden grundsätzlich bei jedem Kalt- oder Warmstart der SPS initialisiert. Ein laufendes Zeitwerk wird also stets durch einen Kalt- oder Warmstart der SPS abgebrochen.

Kaltstart:

- Befehl KALT <CR>
- erstmaliges Zuschalten der Spannung

Warmstart:

- Befehl WARM <CR>
- Zuschalten der Spannung
- RESET-Schalter

Wichtiger Hinweis

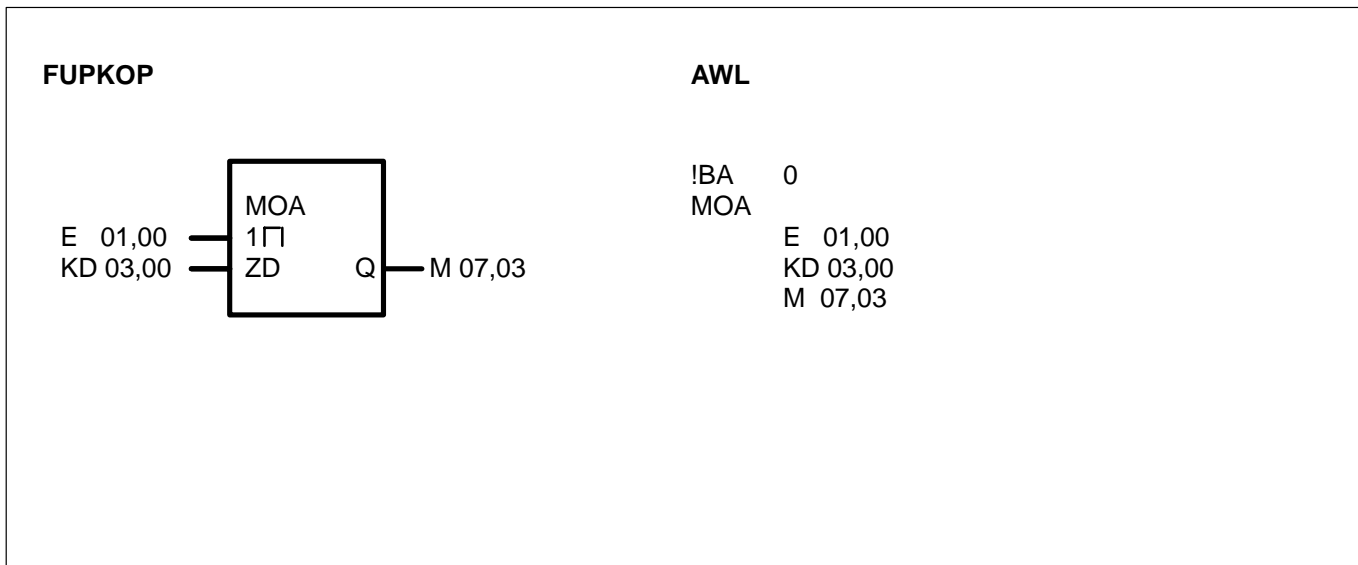
Das Verhalten der Zeitwerke ist geändert worden. Alle laufenden Zeitwerke werden bei folgenden Aktionen automatisch gestoppt:

- SPS-Programm abbrechen
- RUN/STOP-Schalter von RUN -> STOP

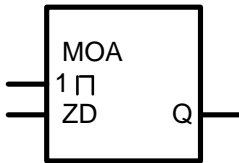
Diese Änderung gilt ab:

07 KR 91 Index g
07 KT 92 Index g
07 KT 93

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
1Π	E	L	N	P	J	0	0
ZD	E	D	N	P	J	0	0
Q	A	L	N	P	J	0	0

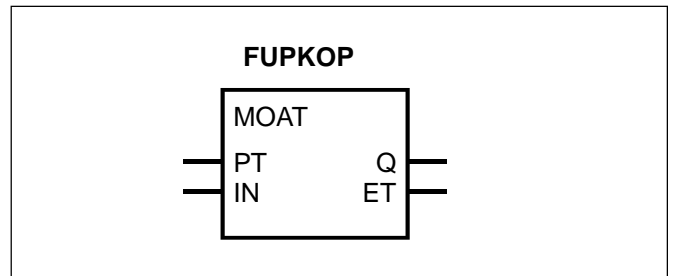
VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MOA			
00002		PP 0	1Π	Eingang BINÄR (Starte Zeit)
00003		PP 0	ZD	Eingang DOPPELWORT (Zeitwert)
00004		PP 0	Q	Ausgang BINÄR (Zeit)

Eine 0/1-Flanke am Eingang IN bewirkt eine 0/1-Flanke am Ausgang Q. Bleibt der Eingang IN auf 1-Pegel, wird nach der Zeitdauer PT eine 1/0-Flanke am Ausgang Q ausgegeben.

Der Ausgang ET zeigt die aktuelle Zeit an.

Geht der Eingang IN vor Ablauf der Zeitdauer PT wieder auf 0-Pegel zurück, so wird auch sofort der Ausgang Q wieder auf 0-Pegel gesetzt.



Parameter

IN	BINÄR	E, M, A, S, K	Eingangssignal
PT	WORT/DOPPELWORT	MW&MW+1, MD, KW&KW+1, KD, EW&EW+1, AW&AW+1	Impulslänge
Q	BINÄR	M, A	Ausgangssignal
ET	WORT/DOPPELWORT	MW&MW+1, MD	Zeitanzeige

Verfügbar ab: Advant Controller 31, Serie 40 und 50

Beschreibung

Eine 0/1-Flanke am Eingang IN bewirkt eine 0/1-Flanke am Ausgang Q. Bleibt der Eingang IN auf 1-Pegel, wird nach der Zeitdauer PT eine 1/0-Flanke am Ausgang Q ausgegeben.

Geht der Eingang IN vor Ablauf der Zeitdauer PT wieder auf 0-Pegel zurück, so wird auch sofort der Ausgang Q wieder auf 0-Pegel gesetzt.

Die Istzeit wird am Ausgang ET angezeigt.

Für Zeiten, die kleiner als 65 s sind, können am Eingang PT bzw. Ausgang ET Variablen vom Typ WORT verwendet werden: Merker, Konstanten, andere WORT-Funktionen (+, -, AWT) und die Potentiometereingänge der Zentraleinheit. In diesem Falle belegen die MW, KW, EW und AW zwei aufeinanderfolgende Adressen. MW+1, KW+1, EW+1 und AW+1 dürfen im Programm nicht mehr verwendet werden.

Die Zeitangabe erfolgt in Millisekunden.

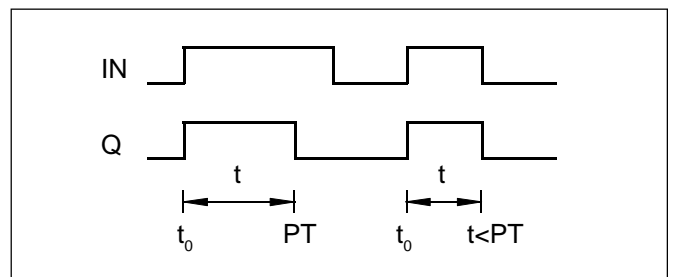
Angebbarer Zeitbereich: 5 ms ... 24,8 Tage.

Bei der indirekten Konstanten KD wird der Wert der Impulslänge im Langtext angegeben.

Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit.

Sinnvoller Bereich für PT: > 1 Zykluszeit.

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar.



Allgemeines Verhalten

- Gestartete Zeitwerke werden vom Betriebssystem der SPS bearbeitet und sind deshalb vollkommen unabhängig von der Bearbeitung des SPS-Programms. Erst nach Ablauf des Zeitwerks erfolgt eine entsprechende Meldung des Betriebssystems an den zugehörigen Zeitbaustein im SPS-Programm.
- Die Bearbeitung des Zeitwerks im Betriebssystem der SPS wird durch folgende Befehle nicht beeinflusst:
 - Programm abbrechen
 - Programm starten
 - Programm stoppen
 - Programm fortsetzen

Dies bedeutet: Ist ein Zeitwerk gestartet, so wird es im Betriebssystem der SPS auch dann weiter bearbeitet, wenn das zugehörige SPS-Programm abgebrochen, wieder gestartet oder gestoppt und wieder fortgesetzt wird.

Initialisierung

Die Zeitwerke werden grundsätzlich bei jedem Kalt- oder Warmstart der SPS initialisiert. Ein laufendes Zeitwerk wird also stets durch einen Kalt- oder Warmstart der SPS abgebrochen.

Kaltstart

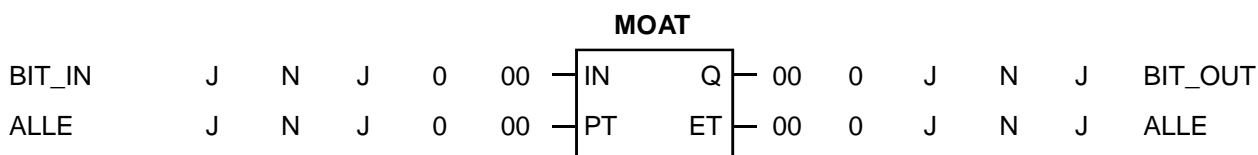
- Befehl KALT <CR>
- Erstmaliges Zuschalten der Spannung

Warmstart

- Befehl WARM <CR>
- Zuschalten der Spannung
- RESET-Schalter

Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr		Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	--	----	-----	-----	-----	-----	--------------



Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	IBA	000	
	00001	MOAT		
	00002	PP	00	IN
	00003	PP	00	PT
	00004	PP	00	ET
	00005	PP	00	Q

Kommunikations–Funktionsblock

Die Zentraleinheit ist ein Master an einem MODBUS–Netzwerk und kann mit anderen Geräten über das MODBUS–Protokoll kommunizieren.

- die Systemkonstante KW 00,06 = 100
- eine Verbindung zwischen den Pins 7 und 6 am Stecker der seriellen Schnittstelle

Die Funktion MODBUS–MASTER der Zentraleinheit wird eingestellt durch:

In einem Anwenderprogramm können mehrere MODBUS–Funktionsblocks verwendet werden.

Parameter

FREI	BINÄR	E,A,M,K,S	Freigabesignal für eine Kommunikation (0→1–Flanke)
SLAV	WORT	EW,AW,MW,KW	Slave–Nummer
FCT	WORT	EW,AW,MW,KW	Funktions–Code
TIME	WORT	EW,AW,MW,KW	Time–out für MODBUS Kommunikation
ADDR	WORT	EW,AW,MW,KW	Adresse im Slave
NB	WORT	EW,AW,MW,KW	Anzahl an Daten (zum Senden oder Lesen)
DATA	WORT	EW,AW,MW,KW	Daten zum Senden an den Slave oder zum Schreiben mit den Daten, die von einem Slave–MODBUS empfangen wurden
	BINÄR	E,A,M	
RDY	BINÄR	A,M	Ready, Kommunikation läuft
ERR	BINÄR	A,M	Kommunikations–Fehler
ERN	WORT	AW,MW	Kommunikations–Fehler, Detail–Information

VE–Daten

Laufzeit:		
Grundlaufzeit:		nicht verfügbar
Zusätzliche Laufzeit:		nicht verfügbar
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		1 Wort
Verfügbar ab:		ABB Procontic CS31 / 07 KR 31 V 2.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Das MODBUS–Protokoll ist ein Master/Slave–Protokoll. Der Master sendet einen Datenübertragungsblock (Frame) und wartet auf eine Antwort (das Timeout ist festgelegt). Binäre oder numerische Daten können in einem Slave geschrieben oder gelesen werden.

Der Datenbereich im Master wird durch die Adresse der ersten Variablen gewählt. Die Größe des Bereichs ist zum Senden oder Empfangen notwendig. Das Lesen oder Schreiben findet automatisch aus diesen Datenbereichen statt.

FREI BINÄR

Eine steigende Flanke am FREI–Eingang führt zur Ausgabe einer Anforderung an einen Slave–MODBUS, vorausgesetzt, daß der Funktionsblock dazu in der Lage ist (RDY = 1).

Wenn am FREI–Eingang eine steigende Flanke auftritt, obwohl RDY = 0 ist (Baustein ist nicht bereit für eine neue MODBUS–Kommunikation), so wird diese Flanke ignoriert.

SLAV WORT

Adresse des Slaves, der die Anforderung empfängt.
Wert: $0 \leq ADDR \leq 255$

Falls die Adresse 0 ist (ADDR = 0), lesen alle Slaves am MODBUS–Netzwerk den Frame.

FCT WORT

Die Funktion hängt ab von den Parameter–Typen und ob geschrieben oder gelesen wird.

- Wert:
- 1: Lesen von n Bits
 - 2: Lesen von n Bytes
 - 3: Lesen von n Worten
 - 4: Schreiben von n Worten
 - 5: Schreiben von einem Bit
 - 6: Schreiben von einem Wort
 - 7: Schnelles Lesen
 - 15: Schreiben von n Bits
 - 16: Schreiben von n Worten

Die übrigen Funktionscodes werden von der 07 KR 31 nicht unterstützt. Tritt ein falscher Funktionscode auf, wird im Wort ERN der Fehler 1 gesetzt.

TIME WORT

Time-out für die Kommunikation (maximale Zeit für eine Antwort des Slave-MODBUS).

Der Wert wird in Millisekunden angegeben.

Zykluszeit (KD 00,00) < TIME < 32767

Wenn ein Timeout auftritt, liefert der Ausgang ERN den Wert 9.

ADDR WORT

Adresse der Daten im Slave-Speicher, die zu lesen oder zu schreiben sind.

NB WORT

Anzahl der Daten, die im Slave zu lesen oder zu schreiben sind.

Die Anzahl definiert ebenfalls die Größe des Datenbereiches im Master, von dem gesendet oder empfangen wird.

DATA WORT, BINÄR

DATA definiert die erste Variable des Datenbereiches im Master. Die Größe des Bereichs hängt vom Parameter NB ab.

Unterschiedliche Fälle sind möglich in Abhängigkeit vom Funktionscode und dem Operanden:

Lesen:

ADDR	DATA	Ergebnis
Wort	Wort	identisch
Bit	Bit	identisch
Bit	Wort	dasselbe wie Funktionsblock PACK
Wort	Bit	Das erste Bit im Wort wird im ersten Bit der Gruppen-Nummer plaziert. Beispiel: DATA = M 00,07 Das erste Bit des gelesenen Wortes wird in M 00,00 geschrieben.

Schreiben:

ADDR	DATA	Ergebnis
Wort	Wort	identisch
Bit	Bit	identisch
Bit	Wort	dasselbe wie Funktionsblock PACK
Wort	Bit	dasselbe wie Funktionsblock UNPACK

RDY BINÄR

Der Ausgang RDY (Ready) zeigt an, ob eine MODBUS-Kommunikation läuft oder nicht. Solange eine Kommunikation läuft, ist der Ausgang auf 0. Der Funktionsbaustein kann für eine neue Kommunikation nur benutzt werden, wenn RDY = 1 ist.

ERR BINÄR

Der Ausgang ERR zeigt einen Fehler an, der während der Kommunikation aufgetreten ist. Der Wort-Ausgang ERN liefert Einzelheiten zu dem Fehler.

ERR = 1 -> Fehler,
ERR = 0 -> Kein Fehler oder Kommunikation läuft.

Der Fehler ist klar nach einer Zykluszeit.

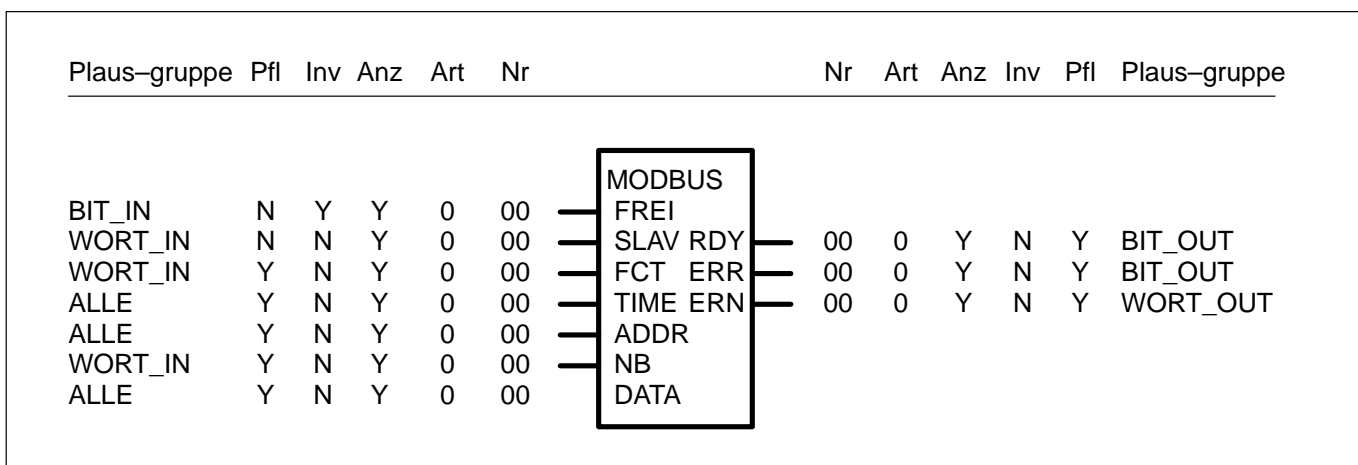
ERN WORT

Einzelheiten zum Fehler:

- 0: Kein Fehler
- 1: Unbekannter Funktionscode
- 2: Adreßfehler
- 3: Datenfehler
- 9: Time-out
- 10: Prüfsummen-Fehler (Checksum)

Der Fehler ist klar nach einer Zykluszeit.

Grafik:



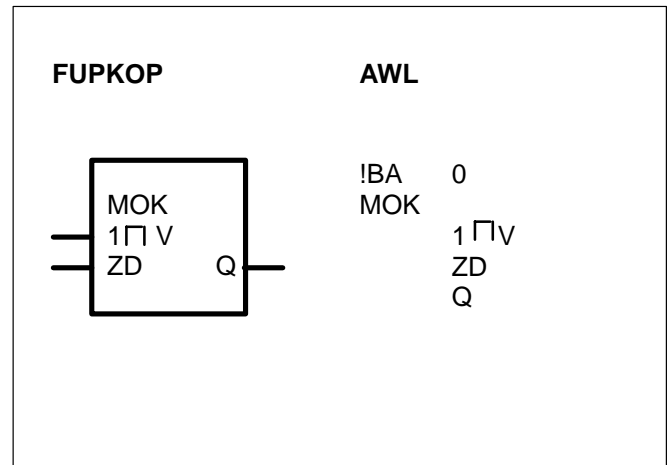
Quelle:

SN	LN	Befehl / Variable	Symbol	Langtext
00000	00000	;MODBUS-MASTER		
	00001	!BA 000		
	00002	CALLUP		
	00003	PP 00	FREI	
	00004	#H 0300	OFFSET	
	00005	#H 0000	SEGMENT	
	00006	#H 0000	NB HIST	
	00007	# +00009	NB DATA	
	00008	PP 00	SLAV	
	00009	PP 00	FCT	
	00010	PP 00	TIME	
	00011	PP 00	ADDR	
	00012	PP 00	NB	
	00013	PP 00	DATA	
	00014	PP 00	RDY	
	00015	PP 00	ERR	
	00016	PP 00	ERN	
	00017	;MODBUS-MASTER END		

Eine 0–1–Flanke am Eingang 1Π V bewirkt eine 0–1–Flanke am Ausgang Q. Nach Ablauf der Zeitdauer ZD wird der Ausgang Q wieder auf 0–Pegel zurückgesetzt.

Eine zweite 0–1–Flanke des Eingangs 1Π V vor Ablauf der Zeit ZD wird ignoriert.

Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit



Parameter

1Π V	BINÄR	E, M, A, S, K	Eingangssignal
ZD	DOPPELWORT	MD, KD	Impulslänge
Q	BINÄR	M, A	Ausgangssignal

VE–Daten

Serie 90

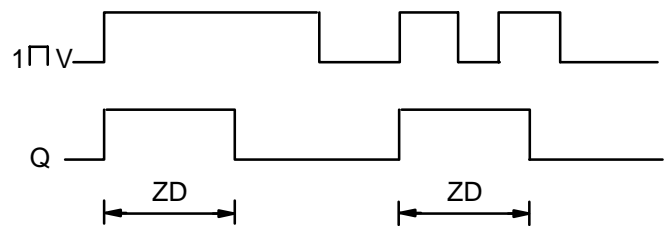
Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	20 µs; Höchstlaufzeit 380 µs, bei Flanken am Eingang 1Π V	97 µs / 554 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	1 Wort	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Eine 0–1–Flanke am Eingang 1Π V bewirkt eine 0–1–Flanke am Ausgang Q. Nach Ablauf der Zeitdauer ZD wird der Ausgang Q wieder auf 0–Pegel zurückgesetzt.

Eine zweite 0–1–Flanke des Eingangs 1Π V vor Ablauf der Zeit ZD wird ignoriert.



Die Zeitangabe erfolgt in Millisekunden, wobei nur ganzzahlige Vielfache von 5 ms zulässig sind (Bsp.: 5 ms, 10 ms, 500 ms, 100 000 ms ...). Angebbbarer Zeitbereich: 5 ms ... 24,8 Tage.

Bei der indirekten Konstanten KD wird der Wert der Impulslänge im Langtext angegeben.

Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit

Sinnvoller Bereich für ZD: > 1 Zykluszeit

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar.

Allgemeines Verhalten

- Gestartete Zeitwerke werden vom Betriebssystem der SPS bearbeitet und sind deshalb vollkommen unabhängig von der Bearbeitung des SPS-Programms. Erst nach Ablauf des Zeitwerks erfolgt eine entsprechende Meldung des Betriebssystems an den zugehörigen Zeitbaustein im SPS-Programm.
- Die Bearbeitung eines Zeitwerks im Betriebssystem der SPS wird durch folgende Befehle *nicht* beeinflusst:
 - Programm abbrechen
 - Programm starten
 - Programm stoppen
 - Programm fortsetzen

D.h. ist ein Zeitwerk gestartet, so wird es im Betriebssystem der SPS auch dann weiter bearbeitet, wenn das zugehörige SPS-Programm abgebrochen, wieder gestartet oder gestoppt und wieder fortgesetzt wird.

Initialisierung

Die Zeitwerke werden grundsätzlich bei jedem Kalt- oder Warmstart der SPS initialisiert. Ein laufendes Zeitwerk wird also stets durch einen Kalt- oder Warmstart der SPS abgebrochen.

Kaltstart:

- Befehl KALT <CR>
- erstmaliges Zuschalten der Spannung

Warmstart:

- Befehl WARM <CR>
- Zuschalten der Spannung
- RESET-Schalter

Wichtiger Hinweis

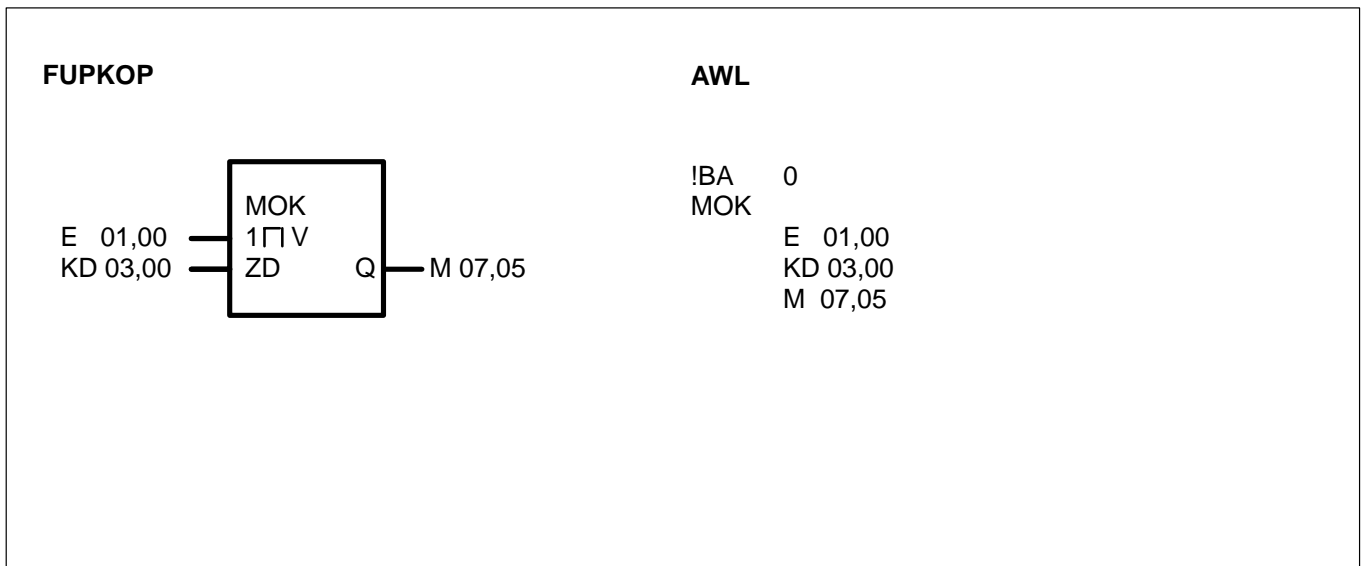
Das Verhalten der Zeitwerke ist geändert worden. Alle laufenden Zeitwerke werden bei folgenden Aktionen automatisch gestoppt:

- SPS-Programm abbrechen
- RUN/STOP-Schalter von RUN -> STOP

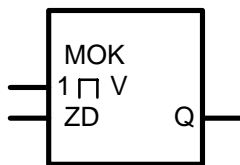
Diese Änderung gilt ab:

07 KR 91 Index g
07 KT 92 Index g
07 KT 93

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
1Π V	E	L	N	P	J	0	0
ZD	E	D	N	P	J	0	0
Q	A	L	N	P	J	0	0

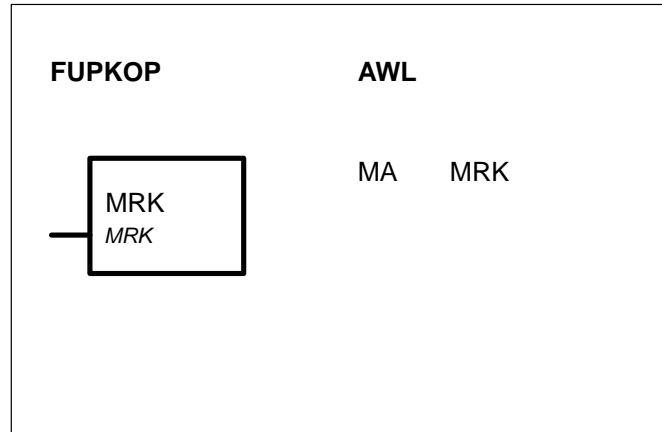
VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MOK			
00002		PP 0	1Π V	Eingang BINÄR (Starte Zeit)
00003		PP 0	ZD	Eingang DOPPELWORT (Zeitwert)
00004		PP 0	Q	Ausgang BINÄR (Zeit)

Das Sprungziel eines Sprungbausteins oder Laufzahlblocks wird im SPS-Programm durch eine Marke definiert.

Am Eingang MRK wird der Name und die Nummer der Marke in folgender Form angegeben: MRK Nr

dabei ist: MRK Schlüsselwort
 $0 \leq \text{Nr} \leq 999$ Nummer der Marke



Parameter

MRK SPEZIAL MRK Zielmarke

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

0 μ s

0 μ s

Zusätzliche Laufzeit:

—

—

Aktualisierung der Ausgänge:

entfällt

entfällt

Anzahl der Vergangenheitswerte:

keine

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

907 PC 331 / 07 KR 31 V 1.0,
 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

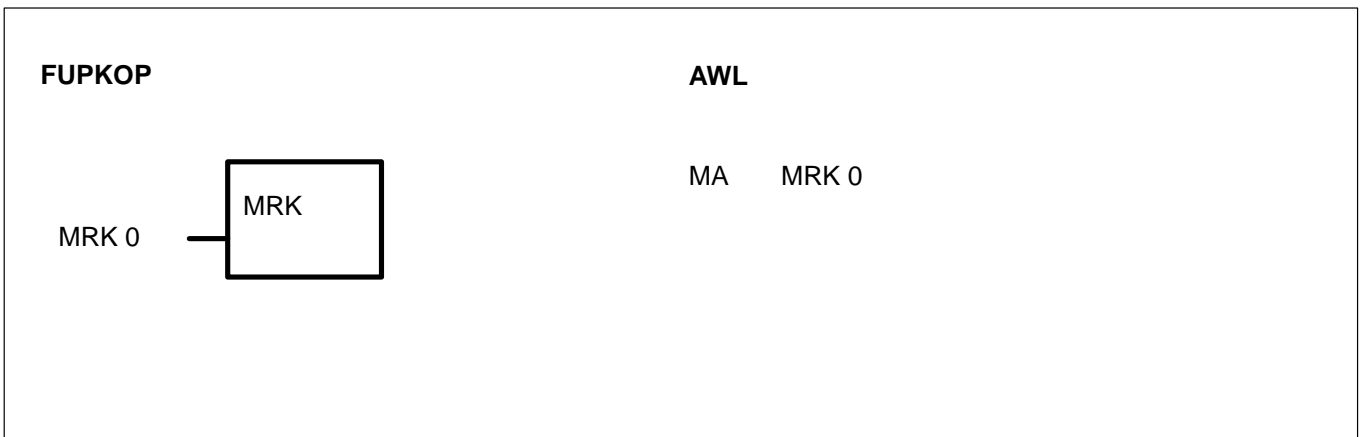
Beschreibung

Das Sprungziel eines Sprungbausteins oder Laufzahlblocks wird im SPS-Programm durch eine Marke definiert.

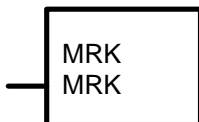
Am Eingang MRK wird der Name und die Nummer der Marke in folgender Form angegeben: MRK Nr

dabei ist: MRK Schlüsselwort
 $0 \leq \text{Nr} \leq 999$ Nummer der Marke

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
MRK	S	S	N	P	N	0	0

VE-AWL-Definition

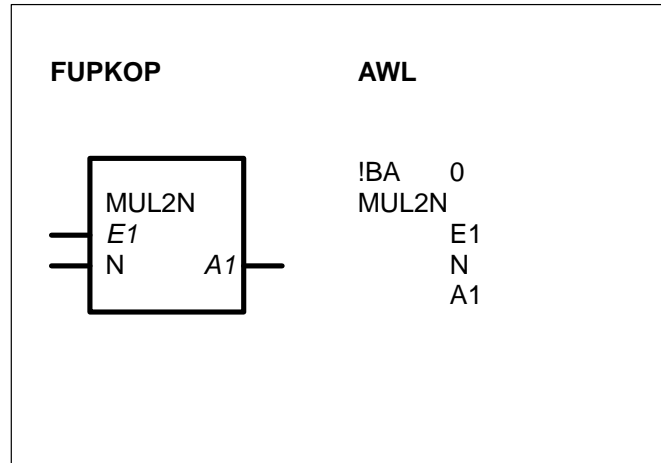
00000 MA PP 0 MRK Zielmarke

Der Wert des Operanden am Eingang E1 wird N-mal bitweise geschoben.

Ist der Wert am Eingang N positiv, wird nach links geschoben. Dies entspricht pro Schieben um eine Bitposition einer Multiplikation des aktuellen Wertes mit 2.

Ist der Wert am Eingang N negativ, wird nach rechts geschoben. Dies entspricht pro Schieben um eine Bitposition einer Division des aktuellen Wertes durch 2.

Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	WORT	EW, MW, AW, KW	Eingangsoperand
N	WORT	EW, MW, AW, KW	Anzahl
A1	WORT	MW, AW	Ergebnis

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	30–35 µs	72 µs
Zusätzliche Laufzeit:	4 µs pro Schiebeoperation	15 µs pro Schiebeoperation
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Wert des Operanden am Eingang E1 wird N-mal bitweise geschoben.

Ist der Wert am Eingang N positiv, wird nach links geschoben. Dies entspricht pro Schieben um eine Bitposition einer Multiplikation des aktuellen Wertes mit 2.

Ist der Wert am Eingang N negativ, wird nach rechts geschoben. Dies entspricht pro Schieben um eine Bitposition einer Division des aktuellen Wertes durch 2.

Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar.

Sinnvoller Bereich für N: $-14 \leq N \leq +14$

Bei N = 0 wird der Wert am Eingang E1 direkt an den Ausgang A1 gegeben.

Vorzeichen des Wertes am Eingang E1:

Das Vorzeichen des Wertes E1 wird durch die Schiebeoperation nicht beeinflusst. D. h. das Vorzeichen des Ausgangswertes ist immer identisch mit dem des Eingangswertes.

Links Schieben (Multiplikation):

Beim Links–Schieben des Wertes am Eingang wird jeweils das freiwerdende Bit 0 mit 0 aufgefüllt. Das Vorzeichenbit (Bit 15) wird dabei nicht verändert, da vorher auf den Grenzwert des Zahlenbereichs begrenzt wird.

Begrenzung des Wertes am Ausgang A1 beim Links–Schieben:

- Für positive Werte am Eingang E1 gilt:

Ist Bit 14 mit einer "1" besetzt und sind aufgrund des Wertes am Eingang N noch Schiebeoperationen auszuführen, so werden diese nicht mehr ausgeführt. Stattdessen wird der Ausgang auf den Grenzwert des positiven Zahlenbereichs gesetzt. D.h. spätestens nach 14–maligem Schieben ist auf jeden Fall der Grenzwert erreicht.

Grenzwert: Ausgang A1 = +32767 (7FFF_H).

- Für negative Werte am Eingang E1 gilt:

Ist Bit 14 mit einer "0" besetzt und sind aufgrund des Wertes am Eingang N noch Schiebeoperationen auszuführen, so werden diese nicht mehr ausgeführt. Stattdessen wird der Ausgang auf den Grenzwert des negativen Zahlenbereichs gesetzt. D.h. spätestens nach 14–maligem Schieben ist auf jeden Fall der Grenzwert erreicht.

Grenzwert: Ausgang A1 = –32767 (8001_H).

Rechts Schieben (Division):

Beim Rechts–Schieben rückt jedes Bit eine Stelle nach rechts. Das Vorzeichenbit (Bit 15) behält dabei seinen Wert immer bei. Das freiwerdende Bit (Bit14) wird jeweils mit dem Wert des Vorzeichenbits aufgefüllt.

Begrenzung des Wertes am Ausgang beim Rechts–Schieben:

- Für positive Werte am Eingang E1 gilt:

Ist nur noch Bit 0 mit einer "1" besetzt und sind aufgrund des Wertes am Eingang N noch Schiebeoperationen auszuführen, so wird der Ausgang auf den Wert 0 gesetzt. D.h. spätestens nach 14–maligem Schieben ist auf jeden Fall der Wert 0 erreicht.

Ausgang A1 = 0.

- Für negative Werte am Eingang E1 gilt:

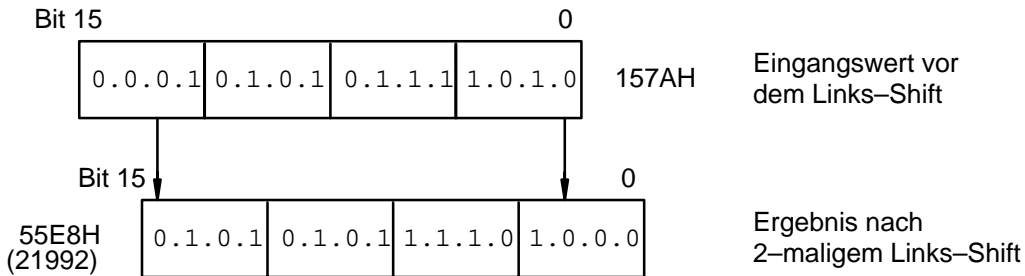
Sind aufgrund des Schiebevorgangs Bit 0 ... Bit 15 mit einer "1" besetzt, so ist der Grenzwert (–1) erreicht. Weitere Schiebevorgänge bleiben ohne Auswirkung. D.h. spätestens nach 15–maligem Schieben ist auf jeden Fall der Wert –1 erreicht.

Ausgang A1 = –1 (FFFF_H).

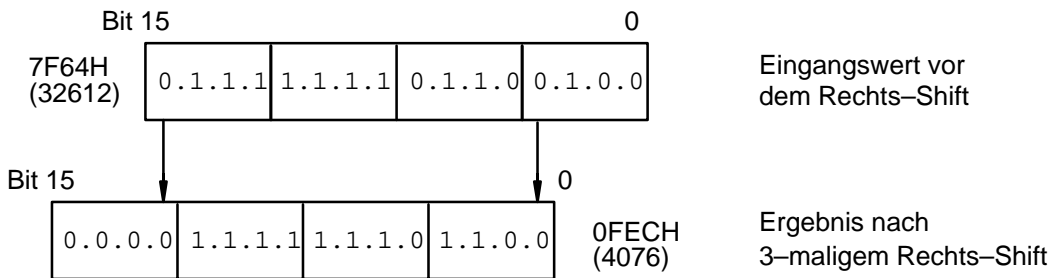
Die Eingänge und der Ausgang sind weder doppelbar noch negierbar.

Beispiele:

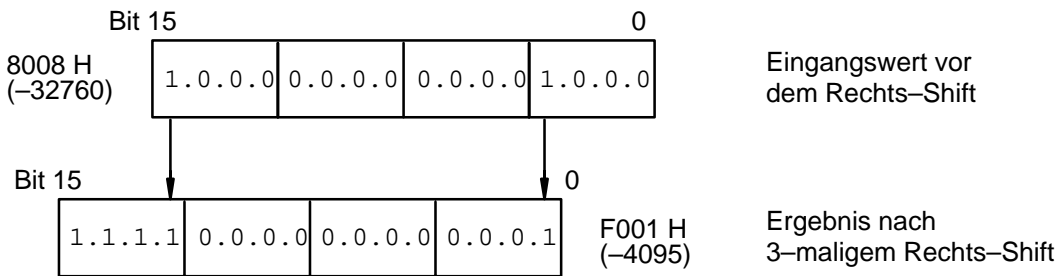
- Eingangswert E1 = 5498 (157AH)
Exponent N = 2 → 2 * Links-Shift



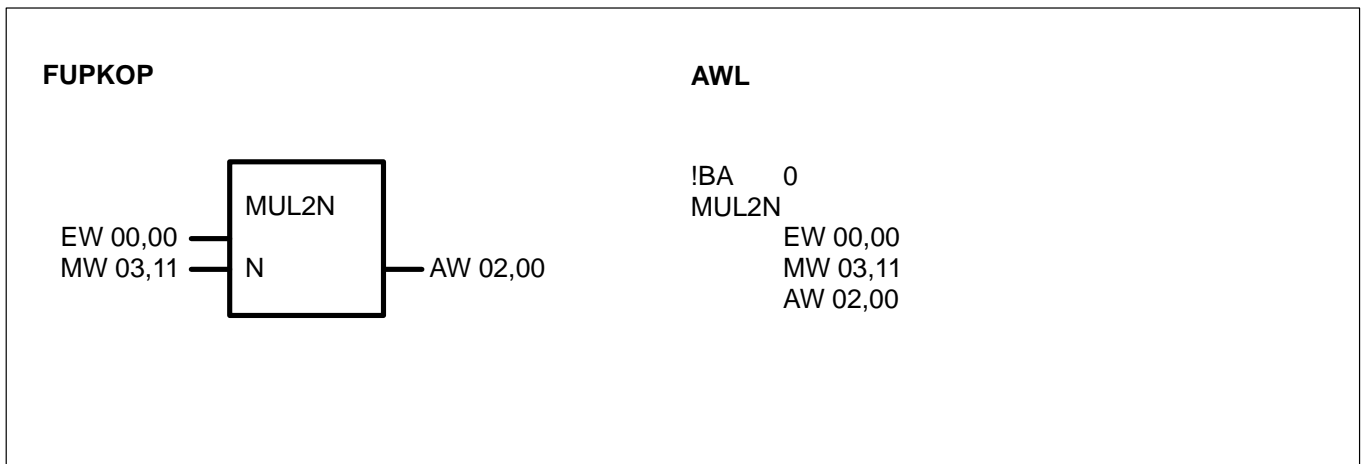
- Eingangswert E1 = 32612 (7F64H)
Exponent N = -3 → 3 * Rechts-Shift



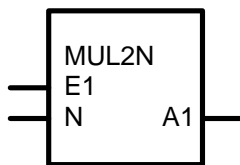
- Eingangswert E1 = -32612 (8008 H)
Exponent N = -3 → 3 * Rechts-Shift



Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
N	E	W	N	P	J	0	0
A1	A	W	N	P	N	0	0

VE-AWL-Definition

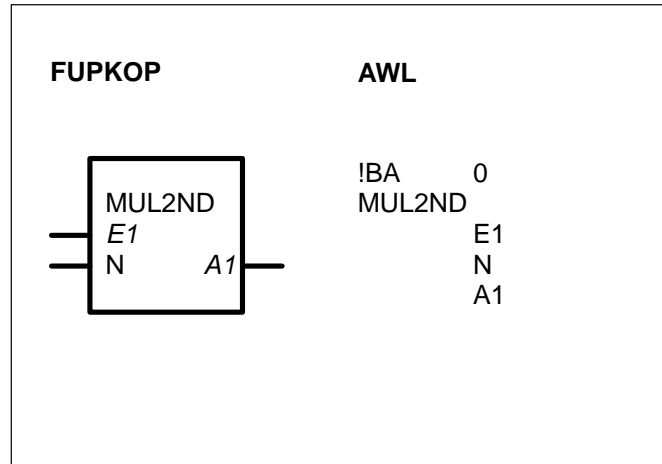
00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MUL2N			
00002		PP 0	E1	Eingang WORT
00003		PP 0	N	Eingang WORT (Anzahl Shift's)
00004		PP 0	A1	Ausgang WORT

Der Wert des Operanden am Eingang E1 wird N–mal bitweise geschoben.

Ist der Wert am Eingang N positiv, wird nach links geschoben. Dies entspricht pro Schieben um eine Bitposition einer Multiplikation des aktuellen Wertes mit 2.

Ist der Wert am Eingang N negativ, wird nach rechts geschoben. Dies entspricht pro Schieben einer Division des aktuellen Wertes durch 2.

Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	DOPPELWORT MD, KD	Eingangsoperand
N	WORT EW, AW, MW, KW	Anzahl
A1	DOPPELWORT MD	Ergebnis

VE–Daten

Laufzeit:	
Grundlaufzeit:	34 µs
Zusätzliche Laufzeit:	6 µs pro Schiebeoperation
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang E1 wird N–mal bitweise geschoben.

Ist der Wert am Eingang N positiv, wird nach links geschoben. Dies entspricht pro Schieben um eine Bitposition einer Multiplikation des aktuellen Wertes mit 2.

Ist der Wert am Eingang N negativ, wird nach rechts geschoben. Dies entspricht pro Schieben einer Division des aktuellen Wertes durch 2.

Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.

Sinnvoller Bereich für N: $-30 \leq N \leq +30$

Bei N = 0 wird der Wert am Eingang E1 direkt an den Ausgang A1 gegeben.

Vorzeichen des Wertes am Eingang E1:

Das Vorzeichen des Wertes E1 wird durch die Schiebeoperation nicht beeinflusst. D.h. das Vorzeichen des Ausgangswertes ist immer identisch mit dem des Eingangswertes.

Links Schieben (Multiplikation):

Beim Links–Schieben des Wertes am Eingang wird jeweils das freiwerdende Bit 0 mit 0 aufgefüllt. Das Vorzeichenbit (Bit 31) wird dabei nicht verändert, da vorher auf den Grenzwert des Zahlenbereichs begrenzt wird.

Begrenzung des Wertes am Ausgang A1 beim Links–Schieben:

- Für positive Werte am Eingang E1 gilt:

Ist Bit 30 mit einer "1" besetzt und sind aufgrund des Wertes am Eingang N noch Schiebeoperationen auszuführen, so werden diese nicht mehr ausgeführt. Stattdessen wird der Ausgang auf den Grenzwert des positiven Zahlenbereichs gesetzt. D.h. spätestens nach 30–maligem Schieben ist auf jeden Fall der Grenzwert erreicht.
Ausgang A1 = +2 147 483 647 (7FFF FFFF_H).

- Für negative Werte am Eingang E1 gilt:

Ist Bit 30 mit einer "0" besetzt und sind aufgrund des Wertes am Eingang N noch Schiebeoperationen auszuführen, so werden diese nicht mehr ausgeführt. Stattdessen wird der Ausgang auf den Grenzwert des positiven Zahlenbereichs gesetzt. D. h. spätestens nach 30-maligem Schieben ist auf jeden Fall der Grenzwert erreicht.

Ausgang A1 = $-2\ 147\ 483\ 647$ ($8000\ 0001_H$).

Rechts Schieben (Division):

Beim Rechts-Schieben rückt jedes Bit eine Stelle nach rechts. Das Vorzeichenbit (Bit 31) behält dabei seinen Wert immer bei. Das freiwerdende Bit (Bit 30) wird jeweils mit dem Wert des Vorzeichenbits aufgefüllt.

Begrenzung des Wertes am Ausgang beim Rechts-Schieben:

- Für positive Werte am Eingang gilt:

Ist nur noch Bit 0 mit einer "1" besetzt und sind aufgrund des Wertes am Eingang N noch Schiebeoperationen auszuführen, so wird der Ausgang auf den Wert 0 gesetzt. D. h. spätestens nach 30-maligem Schieben ist auf jeden Fall der Wert 0 erreicht.

Ausgang A1 = 0.

- Für negative Werte am Eingang E1 gilt:

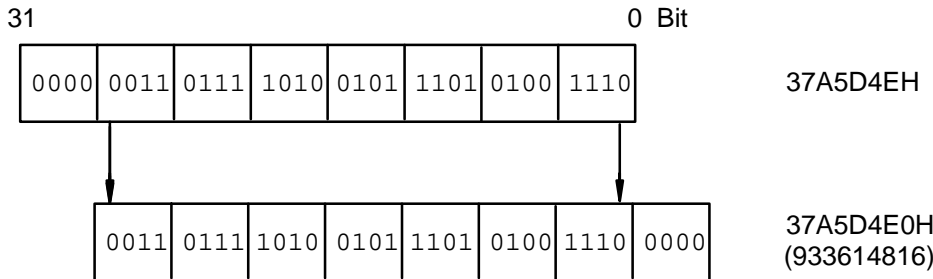
Sind aufgrund des Schiebevorgangs Bit 0 ... Bit 31 mit einer "1" besetzt, so ist der Grenzwert (-1) erreicht. Weitere Schiebevorgänge bleiben ohne Auswirkung. D. h. spätestens nach 31-maligem Schieben ist auf jeden Fall der Wert -1 erreicht.

Ausgang A1 = -1 ($FFFF\ FFFF_H$).

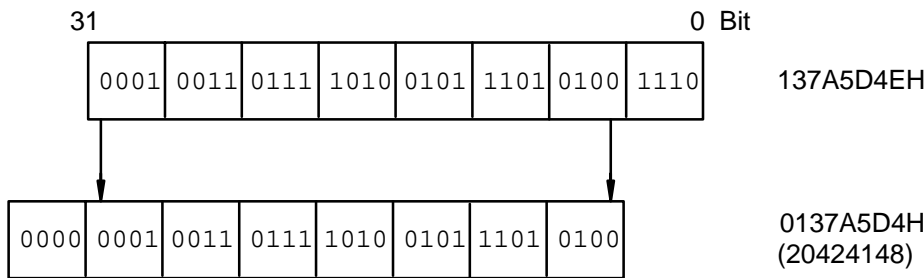
Die Eingänge und der Ausgang sind weder doppelbar noch negierbar.

Beispiele:

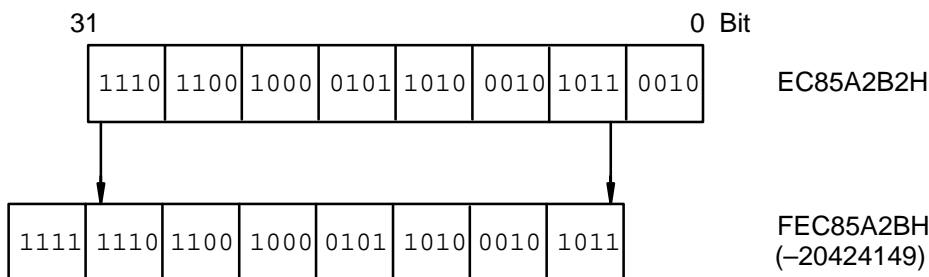
- Eingangswert E1 = 58350926 (37A5D4E_H)
Exponent N = 4 → 4 * Links-Shift



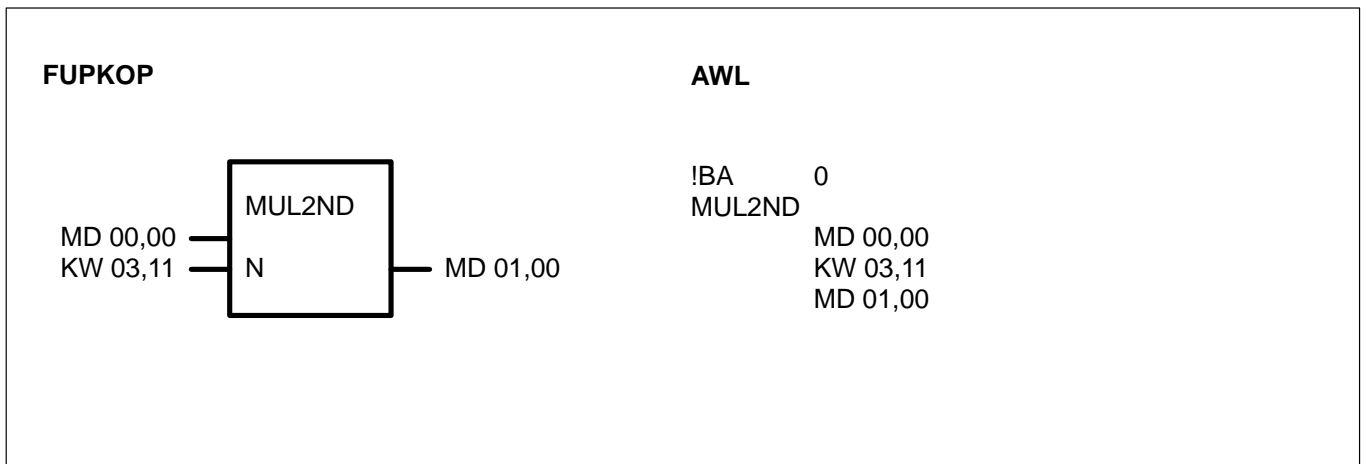
- Eingangswert E1 = 326786382 (137A5D4E_H)
Exponent N = -4 → 4 * Rechts-Shift



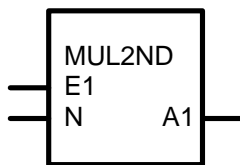
- Eingangswert E1 = -326786382 (EC85A2B2_H)
Exponent N = -4 → 4 * Rechts-Shift



Beispiel



VE–FUP–Definition

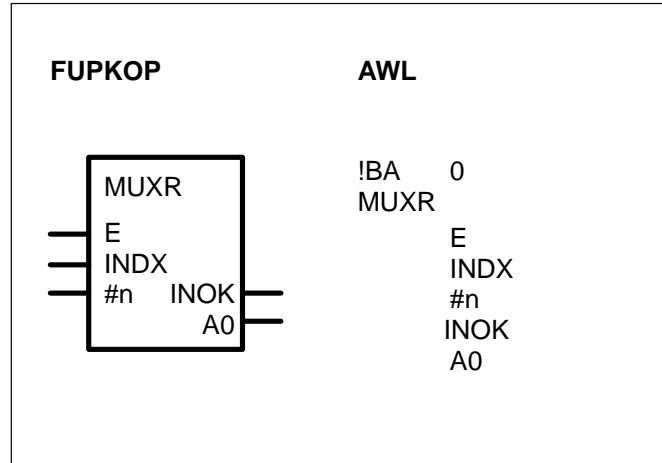


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
N	E	W	N	P	J	0	0
A1	A	D	N	P	N	0	0

VE–AWL–Definition

00000	!BA	0	Nr	Baustein–Nr. (vorbelegt 0)
00001	MUL2ND			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	N	Eingang WORT (Anzahl Shift's)
00004		PP 0	A1	Ausgang DOPPELWORT

Der Funktionsbaustein verbindet in Abhängigkeit vom Eingang INDX den Eingang E mit einem der Ausgänge A0 ... A_{n-1}.



Parameter

E	WORT	EW, AW, MW, KW	Eingang
INDX	WORT	EW, AW, MW, KW	Indexeingang
#n	DIREKTE KONSTANTE	#, #H	Anzahl n der Wortausgänge A0 ... An-1
INOK	BINÄR	A, M	Bereichsüberwachung für Eingang INDX
A0	WORT	AW, MW	Wortausgänge A0 ... An-1; der Ausgang A0 ist doppelbar

VE-Daten

Laufzeit:	
Grundlaufzeit:	44,5 µs
Zusätzliche Laufzeit:	7,5 µs pro zusätzlichem Wortausgang A1 ... An-1
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein verbindet in Abhängigkeit vom Eingang INDX den Eingang E mit einem der Ausgänge A0 ... A_{n-1}.

Die jeweils nicht verbundenen Wortausgänge werden zu 0 gesetzt. Der Wert am Eingang INDX wird auf Zulässigkeit überwacht.

Zusammenhang zwischen E, INDX und A0...An-1:

Mit dem Eingang INDX wird festgelegt, mit welchem der Ausgänge A0...A_{n-1} der Eingang E verbunden wird.

- Es gilt:
- INDX = 1 : E → A0
 - INDX = 2 : E → A1
 - INDX = 3 : E → A2
 - : : :
 - INDX = n : E → An-1

mit $1 \leq \text{INDX} \leq n \leq 32767$ (theoretisch)

E WORT
Eingang, der zu einem der Ausgänge A0 ... An-1 durchgeschaltet wird.

INDX WORT
Indexeingang zur Auswahl eines der Ausgänge A0 ... An-1.
Wertebereich: $1 \leq \text{INDX} \leq n$

Anmerkung:
INDX = 0 kann zur Initialisierung der Ausgänge A0 ... A_{n-1} benutzt werden (A0 ... An-1 = 0).

#n DIREKTE KONSTANTE

Anzahl n der Wortausgänge A0 ... A_{n-1}.

INOK BINÄR

Bereichsüberwachung des Eingangs INDX

Der Ausgang zeigt an, ob sich der Eingang INDX im gültigen Bereich befindet.

Gültiger Bereich: $1 \leq \text{INDX} \leq n$

INOK = 1 → Indexeingang INDX im gültigen Bereich

INOK = 0 → Indexeingang INDX im ungültigen Bereich

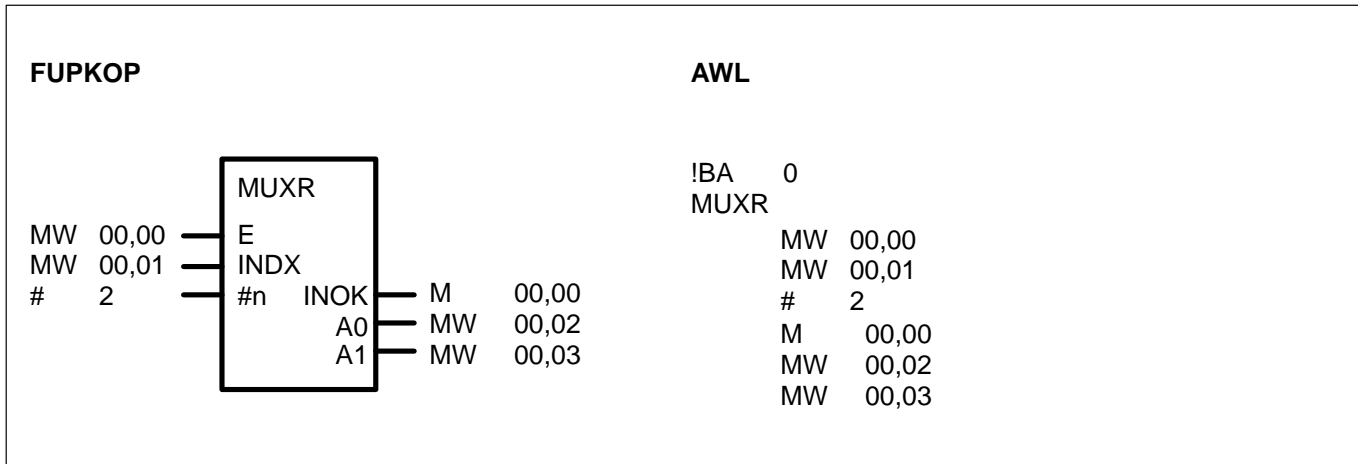
→ $A_0 \dots A_{n-1} = 0$

Befindet sich der Worteingang INDX nicht im gültigen Bereich, so werden alle Wortausgänge A_0 bis A_{n-1} zu Null gesetzt. Der $\text{INDX} = 0$ kann somit z.B. zur Initialisierung der Ausgänge benutzt werden ($A_0 \dots A_{n-1} = 0$).

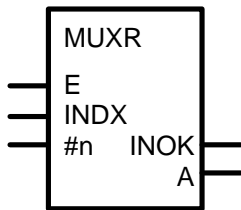
A0 WORT

Der Ausgang A_0 ist doppelbar ($A_0 \dots A_{n-1}$). Der Eingang E wird einem der n Ausgänge $A_0 \dots A_{n-1}$ zugewiesen.

Beispiel



VE-FUP-Definition

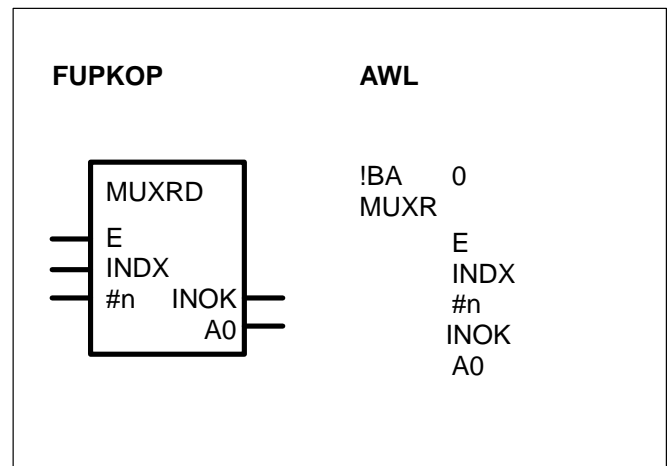


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E	E	W	N	P	J	0	0
INDX	E	W	N	P	J	0	0
#n	K	W	N	P	J	0	0
INOK	A	L	N	P	J	0	0
A	A	W	N	P	J	1	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	MUXR			
00002		PP 0	E	Eingang (WORT)
00003		PP 0	INDX	Index (WORT)
00004		PP 0	#n	Anzahl der Ausgänge (#)
00005		PP 0	INOK	Ausgang Index gültig (BINÄR)
00006	[1		
		PP 1	A	Ausgang (WORT)
]	1		

Der Funktionsbaustein verbindet in Abhängigkeit vom Eingang INDX den Eingang E mit einem der Ausgänge A0 ... A_{n-1}.



Parameter

E	DOPPELWORT	MD, KD	Eingang
INDX	WORT	EW, AW, MW, KW	Indexeingang
#n	DIREKTE KONSTANTE	#, #H	Anzahl n der Doppelwortausgänge A0 ... A _{n-1}
INOK	BINÄR	A, M	Bereichsüberwachung für Eingang INDX
A0	WORT	MD	Doppelwortausgänge A0 ... A _{n-1} ; der Ausgang A0 ist doppelbar

VE-Daten

Laufzeit:	
Grundlaufzeit:	43 µs
Zusätzliche Laufzeit:	10 µs pro zusätzlichem Doppelwort-Ausgang A1 ... A _{n-1}
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein verbindet in Abhängigkeit vom Eingang INDX den Eingang E mit einem der Ausgänge A0 ... A_{n-1}.

Die jeweils nicht verbundenen Doppel-Wortausgänge werden zu 0 gesetzt. Der Wert am Eingang INDX wird auf Zulässigkeit überwacht.

Zusammenhang zwischen E, INDX und A0...A_{n-1}:

Mit dem Eingang INDX wird festgelegt, mit welchem der Ausgänge A0...A_{n-1} der Eingang E verbunden wird.

- Es gilt:
- INDX = 1 : E → A0
 - INDX = 2 : E → A1
 - INDX = 3 : E → A2
 - : : :
 - INDX = n : E → A_{n-1}

mit: $1 \leq \text{INDX} \leq n \leq 32767$ (theoretisch)

E DOPPELWORT

Eingang, der zu einem der Ausgänge A0 ... A_{n-1} durchgeschaltet wird.

INDX WORT

Indexeingang zur Auswahl eines der Ausgänge A0 ... A_{n-1}.
Wertebereich: $1 \leq \text{INDX} \leq n$

Anmerkung:
INDX = 0 kann zur Initialisierung der Ausgänge A0 ... A_{n-1} benutzt werden (A0 ... A_{n-1} = 0).

#n DIREKTE KONSTANTE

Anzahl n der Doppel-Wortausgänge A0 ... A_{n-1}.

INOK BINÄR

Bereichsüberwachung des Eingangs INDX

Der Ausgang zeigt an, ob sich der Eingang INDX im gültigen Bereich befindet.

Gültiger Bereich: $1 \leq \text{INDX} \leq n$

INOK = 1 → Indexeingang INDX im gültigen Bereich

INOK = 0 → Indexeingang INDX im ungültigen Bereich

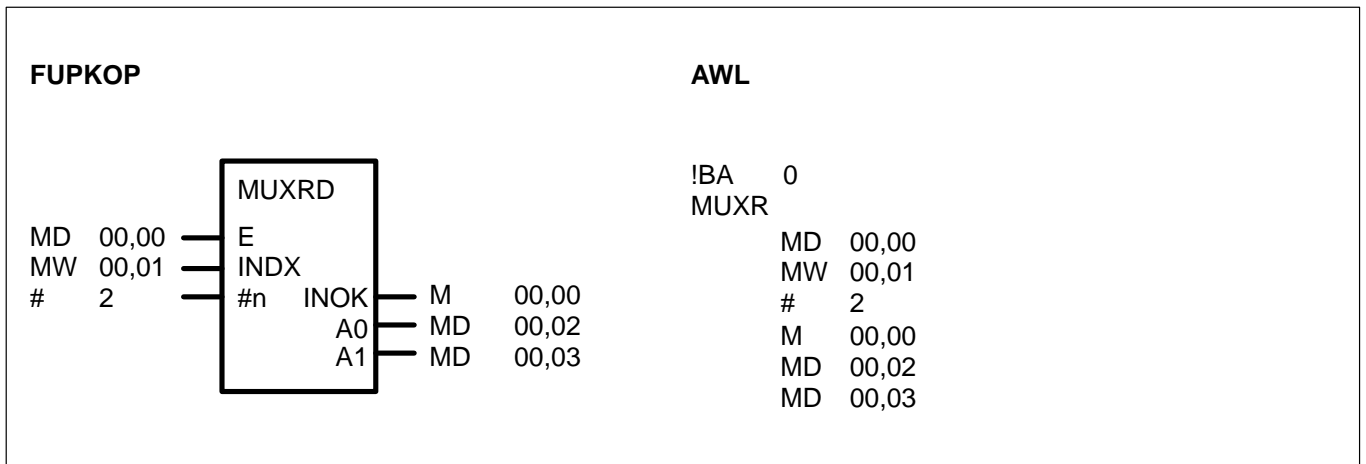
→ $A_0 \dots A_{n-1} = 0$

Befindet sich der Worteingang INDX nicht im gültigen Bereich, so werden alle Doppel-Wortausgänge A_0 bis A_{n-1} zu Null gesetzt. $\text{INDX} = 0$ kann somit z. B. zur Initialisierung der Ausgänge benutzt werden ($A_0 \dots A_{n-1} = 0$).

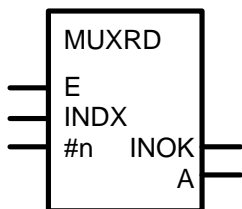
A0 DOPPELWORT

Der Ausgang A_0 ist doppelbar ($A_0 \dots A_{n-1}$). Der Eingang E wird einem der n Ausgänge $A_0 \dots A_{n-1}$ zugewiesen.

Beispiel



VE-FUP-Definition

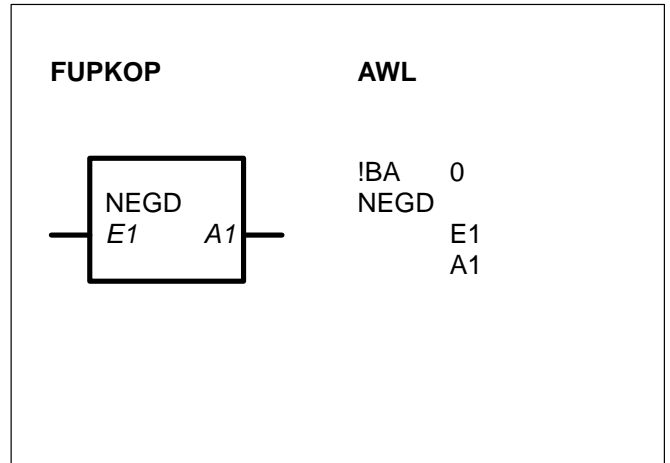


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E	E	D	N	P	J	0	0
INDX	E	W	N	P	J	0	0
#n	K	W	N	P	J	0	0
INOK	A	L	N	P	J	0	0
A	A	D	N	P	J	1	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr (vorbelegt 0)
00001	MUXRD			
00002		PP 0	E	Eingang (DOPPELWORT)
00003		PP 0	INDX	Index (WORT)
00004		PP 0	#n	Anzahl der Ausgänge (#)
00005		PP 0	INOK	Ausgang Index gültig (BINÄR)
	[1		
00006		PP 1	A	Ausgang (DOPPELWORT)
]	1		

Der Wert des Operanden am Eingang E1 wird negiert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	DOPPELWORT MD, KD	Eingangswert
A1	DOPPELWORT MD	negierter Wert

VE-Daten

Laufzeit:	
Grundlaufzeit:	34 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Wert des Operanden am Eingang E1 wird negiert und das Ergebnis dem Operanden am Ausgang A1 zugewiesen.

Liegt am Eingang E1 der unzulässige Zahlenwert 8000 0000_H (−2 147 483 648) an, so wird dem Ausgang A1 der Wert 7FFF FFFF_H (+2 147 483 647) zugewiesen. Der unzulässige Wert wird also vor seiner Negierung durch den zulässigen Wert 8000 0001_H (−2 147 483 647) ersetzt.

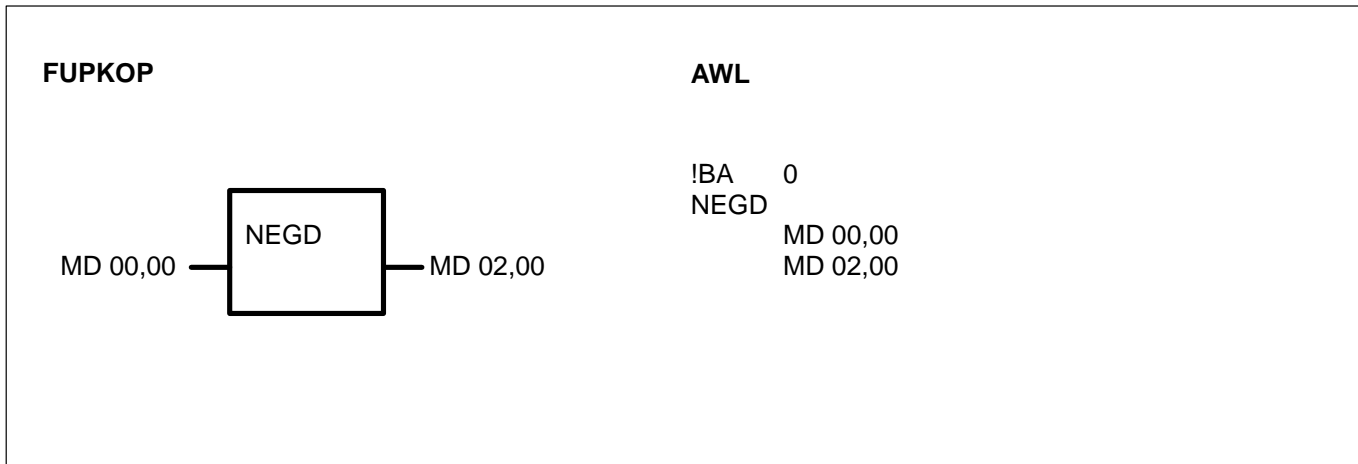
Der Eingang und der Ausgang sind weder doppelbar noch negierbar.

Zahlenbereich

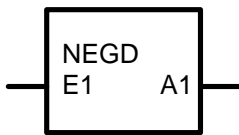
Integer Doppelwort (32 Bit).

- untere Grenze: 8000 0001_H −2 147 483 647
- obere Grenze: 7FFF FFFF_H +2 147 483 647
- unzulässiger Wert: 8000 0000_H —

Beispiel



VE-FUP-Definition

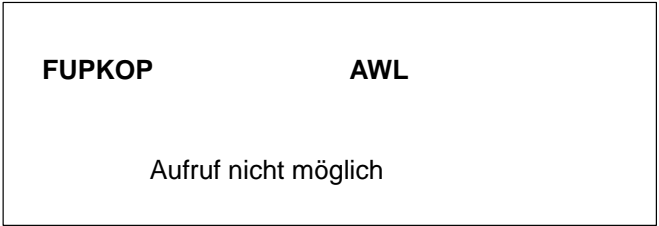


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	D	N	P	N	0	0
A1	A	D	N	P	N	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	NEGD			
00002		PP 0	E1	Eingang DOPPELWORT
00003		PP 0	A1	Ausgang DOPPELWORT

NOTBIT ist eine Grundfunktion in der Hersteller- Bibliothek. Der Name des Elementes in der Bibliothek ist NOTBIT. Das Element kann nicht im VE- und VE-AWL-Editor aufgerufen werden.



Parameter

—

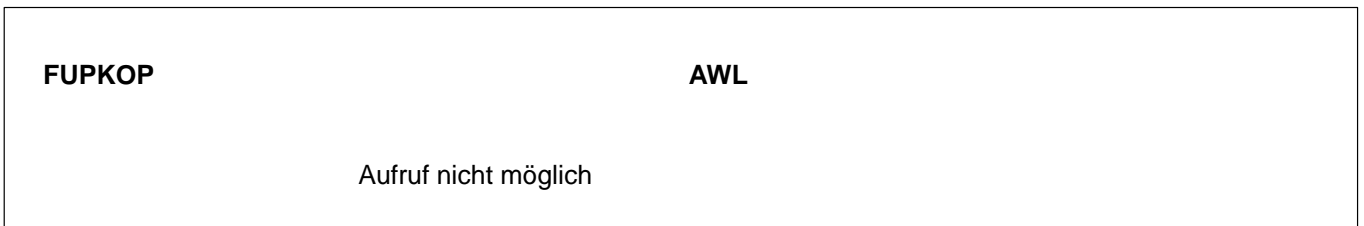
VE-Daten

Laufzeit:
 Grundlaufzeit: 4 µs
 Zusätzliche Laufzeit: —
 Aktualisierung der Ausgänge: —
 Anzahl der Vergangenheitswerte: —
 Verfügbar ab: ABB Procontic CS31 / 907 PC 331

Beschreibung

NOTBIT ist eine Grundfunktion in der Hersteller- Bibliothek. Der Name des Elementes in der Bibliothek ist NOTBIT. Das Element kann nicht im VE- und VE-AWL-Editor aufgerufen werden. Das Programmiergerät benutzt dieses Element zur Realisierung von Verbindungslinien.

Beispiel



VE-FUP-Definition

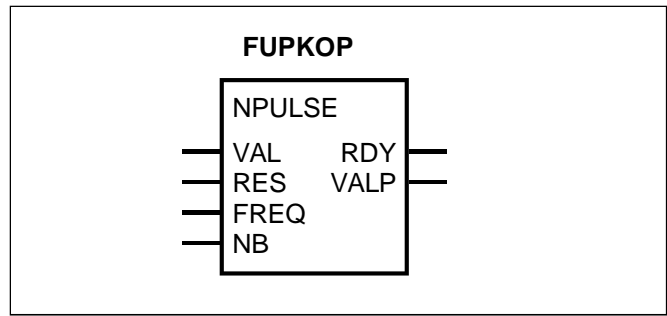
nicht definiert

VE-AWL-Definition

00000 ! PP 0 inp
 00002 =N PP 0 outp

Der Funktionsbaustein erzeugt Impulsfolgen mit bestimmten Frequenzen und bestimmter Anzahl von Impulsen. Er eignet sich damit zur Ansteuerung von Schrittmotoren.

Die Impulsfolgen stehen am Ausgang A 62,00 zur Verfügung.



Parameter

VAL	BINÄR	E, A, M, S	Freigabe
RES	BINÄR	E, A, M, S	Reset
FREQ	WORT	EW, AW, MW, KW	Frequenz
NB	WORT	EW, AW, MW, KW	Anzahl Impulse
RDY	BINÄR	A, M	Ready
VALP	WORT	AW, MW	Anzahl der abgelaufenen Impulse

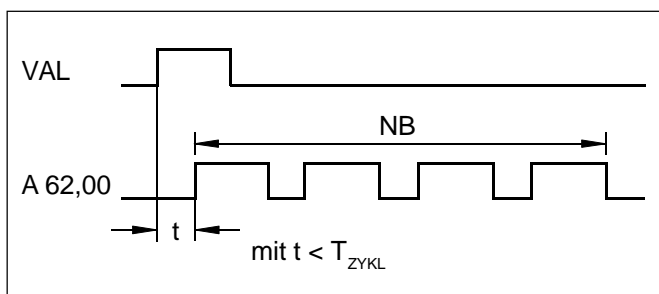
Verfügbar ab: Advant Controller 31, Serie 40 und 50

Beschreibung

Der Funktionsbaustein erzeugt Impulsfolgen mit bestimmten Frequenzen und bestimmter Anzahl von Impulsen. Er eignet sich damit zur Ansteuerung von Schrittmotoren. Die Impulsfolgen stehen am Ausgang A 62,00 zur Verfügung.

Am Ausgang VALP kann die gerade erzeugte Anzahl Impulse abgelesen werden. Die Impulsausgabe wird durch eine positive Flanke am Eingang VAL gestartet. Die erste Periode startet mit dem binaren Wert 1.

Max. Versatz ist eine Zykluszeit.



VAL BINÄR

Freigabe für die Ausgabe der Impulsfolge am festen Ausgang A 62,00.

RES BINÄR

Ein 1-Signal stoppt die Impulserzeugung. Der Ausgang A 62,00 ist 0.

FREQ WORT

Frequenz der Impulsfolge:
 Frequenzbereich: 10 Hz bis 2,6 kHz
 Formel zur Frequenzberechnung:
 Frequenz [Hz = 1000000/((256 - FREQ) • 384)

Beispiele:

FREQ	Frequenz
0	10,1 Hz
1	10,212 Hz
2	10,253 Hz
10	10,586 Hz
100	16,693 Hz
254	1,302 kHz
255	2,604 kHz
<0	10,1 Hz
>255	2,604 kHz

NB WORT

Anzahl der Impulse, die erzeugt werden sollen.
 Wertebereich: 0 bis 32767
 Für NB <0 werden solange Impulse erzeugt, bis der RES-Eingang gesetzt ist.

RDY BINÄR

Ready Bit
 RDY = 0 Impulse werden erzeugt
 RDY = 1 Impulsfolge ist beendet

VALP WORT

Anzahl der gerade erzeugten Impulse. Im Falle eines Fehlers: +/-1.

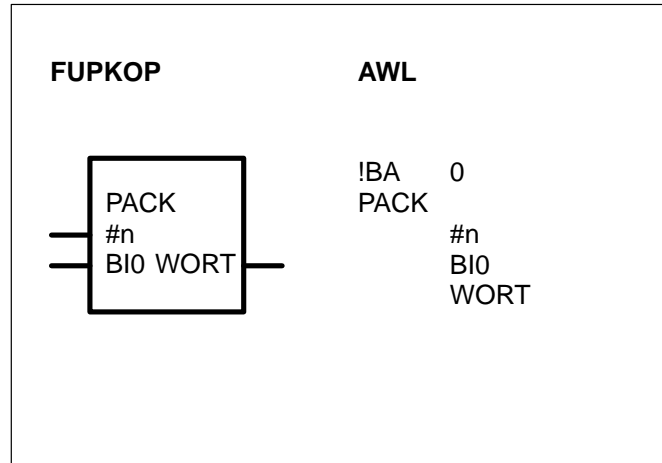
Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr		Nr	Art	Anz	Inv	Pf1	Plaus-gruppe	
NPULSE													
BIT_IN	J	N	J	0	00	VAL	RDY	00	0	J	N	J	BIT_OUT
BIT_IN	J	N	J	0	00	RES	VALP	00	0	J	N	J	WORT_OUT
WORT_IN	J	N	J	0	00	FREQ							
WORT_IN	J	N	J	0	00	NB							

Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	!BA	000	
	00001	NPULSE		
	00002	PP	00	VAL
	00003	PP	00	RES
	00004	PP	00	FREQ
	00005	PP	00	NB
	00006	PP	00	RDY
	00007	PP	00	VALP

Der Funktionsbaustein packt n Binär-Variablen in eine Wort-Variable.



Parameter

#n	DIREKTE KONSTANTE	#, #H	Anzahl der zu packenden Binär-Variablen
BI0 WORT	BINÄR WORT	E, M, A, K, S AW, MW	1. zu packende Binär-Variable; Eingang ist doppelbar Wort-Variable

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:	25 µs
Zusätzliche Laufzeit:	8 µs, pro projektiertes Binär-Variablen

44 µs
31 µs, pro projektiertes Binär-Variablen

Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

ja
keine
907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein packt n Binär-Variablen in eine Wort-Variable.

#n DIREKTE KONSTANTE (#, #H)
Am Eingang #n wird die Anzahl der zu packenden Binär-Variablen angegeben. Die Angabe erfolgt als direkte Konstante.
Es gilt: $1 \leq n \leq 16$
n = 0 ist verboten!

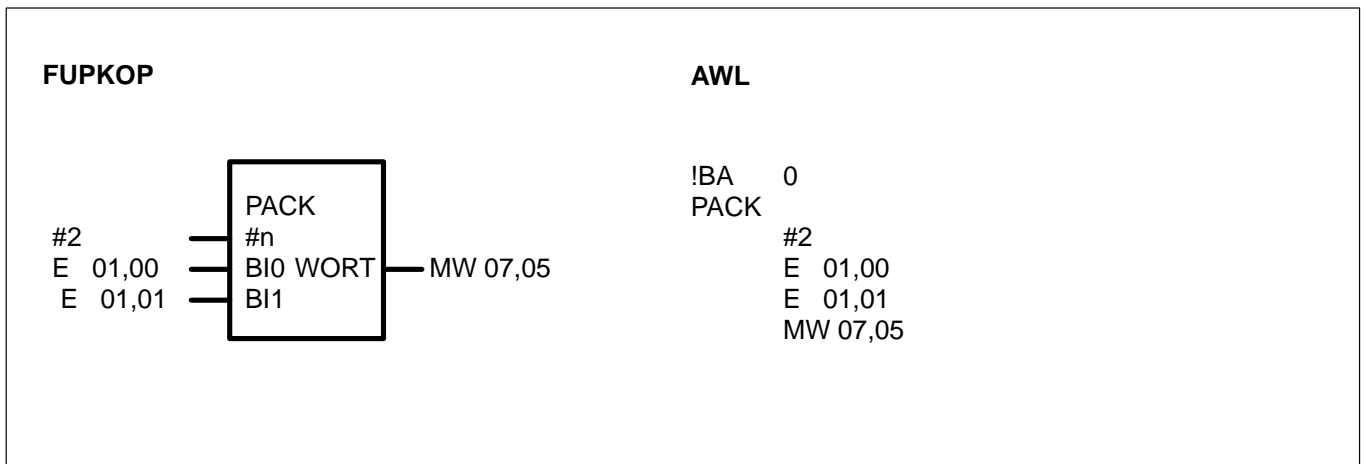
BI0 BINÄR
Der Eingang BI0 ist doppelbar (BI0 ... BI15). An den Eingängen BI0 ... BI15 werden die zu packenden Binär-Variablen vorgegeben.

WORT WORT
Der Wert jeder Binär-Variablen an den Eingängen BI0 ... BI15 wird in das entsprechende Bit (Bit0 ... Bit15) der Variablen am Ausgang WORT geladen.

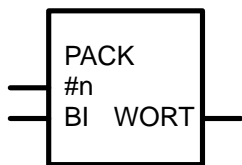
Zuordnung
BI0 → Bit0 der Ausgangsvariablen
BI1 → Bit1 der Ausgangsvariablen
.
.
BI15 → Bit15 der Ausgangsvariablen

Hinweis: Werden vom Anwender weniger als 16 binäre Eingangsvariablen projektiert, so werden die nicht benötigten Bits der Ausgangsvariablen mit dem Wert 0 besetzt.

Beispiel



VE-FUP-Definition

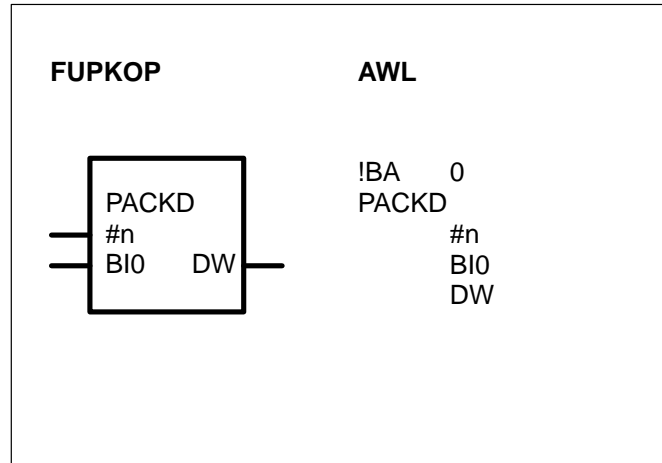


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
BI	E	L	N	P	J	1	0
WORT	A	W	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	PACK			
00002	[PP 0	#n	# KONSTANTE (Anzahl Bit)
		1		
00003]	PP 1	BI	Eingang BINÄR
		1		
00004		PP 0	WORT	Ausgang WORT

Der Funktionsbaustein packt n Binär-Variablen in eine Doppelwort-Variable.



Parameter

#n	DIREKTE KONSTANTE	#, #H	Anzahl der zu packenden Binär-Variablen
BI0	BINÄR DOPPELWORT	E, M, A, K, S MD	1. zu packende Binär-Variable; Eingang ist doppelbar Doppelwort-Variable

VE-Daten

Laufzeit:		
Grundlaufzeit:		31 µs
Zusätzliche Laufzeit:		9 µs, pro projektiertes Binärvariable
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein packt n Binär-Variablen in eine Doppelwort-Variable.

#n DIREKTE KONSTANTE (#,#H)
 Am Eingang #n wird die Anzahl der zu packenden Binär-Variablen angegeben. Die Angabe erfolgt als direkte Konstante.
 Es gilt: $1 \leq n \leq 32$
 n=0 ist verboten!

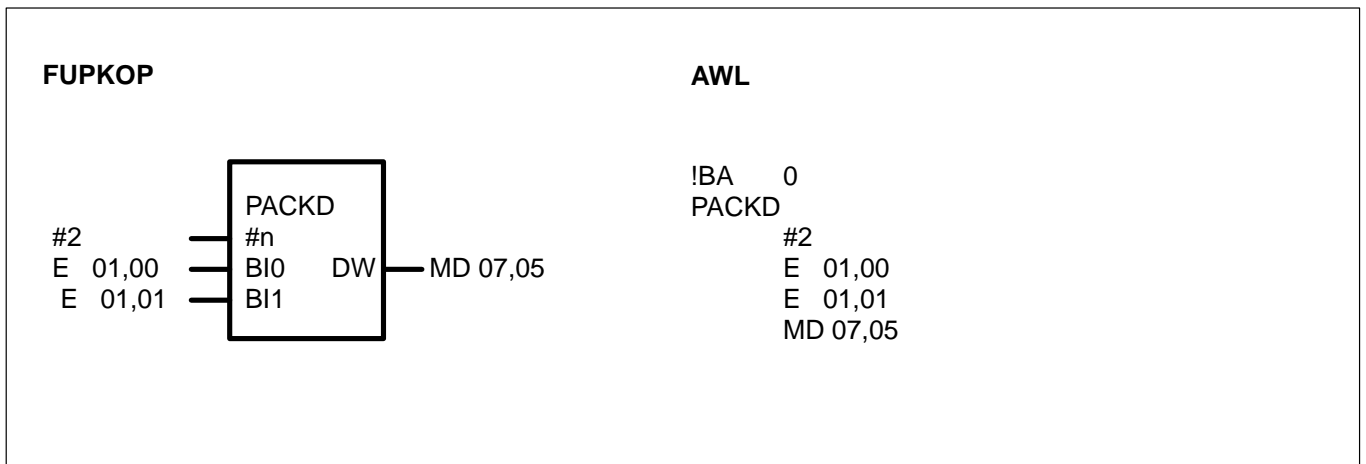
BI0 BINÄR
 Der Eingang BI0 ist doppelbar (BI0 ... BI31). An den Eingängen BI0...BI_n-1 werden die zu packenden Binär-Variablen vorgegeben.

DW DOPPELWORT
 Der Wert jeder Binär-Variablen an den Eingängen BI0 ... BI_n wird in das entsprechende Bit (Bit0 ... Bit31) der Variablen am Ausgang DW geladen.

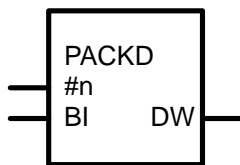
Zuordnung
 BI0 → Bit0 der Ausgangsvariablen
 BI1 → Bit1 der Ausgangsvariablen
 . . .
 BI31 → Bit31 der Ausgangsvariablen

Hinweis: Werden vom Anwender weniger als 32 binäre Eingangsvariablen projektiert, so werden die nicht benötigten Bits der Ausgangsvariablen mit dem Wert 0 besetzt.

Beispiel



VE-FUP-Definition



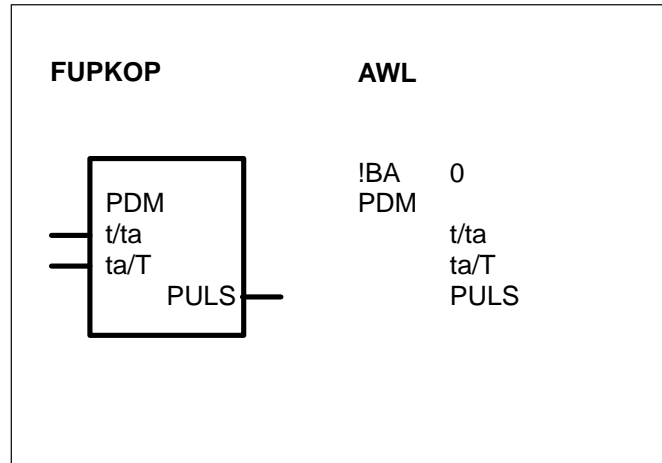
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
BI	E	L	N	P	J	1	0
DW	A	D	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	PACKD			
00002	[PP 0	#n	# KONSTANTE (Anzahl Bit)
		1		
00003]	PP 1	BI	Eingang BINÄR
		1		
00004		PP 0	DW	Ausgang DOPPELWORT

Der Funktionsbaustein erzeugt an seinem Ausgang PULS ein pulsdauermoduliertes binäres Signal.

Am Eingang t/ta wird das Tastverhältnis und am Eingang ta/T wird die Periodendauer für das Ausgangssignal vorgegeben.



Parameter

t/ta	WORT	EW, MW, AW, KW	Tastverhältnis
ta/T	WORT	EW, MW, AW, KW	Periodendauer bezogen auf die Zykluszeit
PULS	BINÄR	A, M	Pulsdauermoduliertes Signal

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	35 µs	1300 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	3 Worte	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 2.2, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein erzeugt an seinem Ausgang PULS ein pulsdauermoduliertes binäres Signal.

Am Eingang t/ta wird das Tastverhältnis und am Eingang ta/T wird die Periodendauer für das Ausgangssignal vorgegeben.

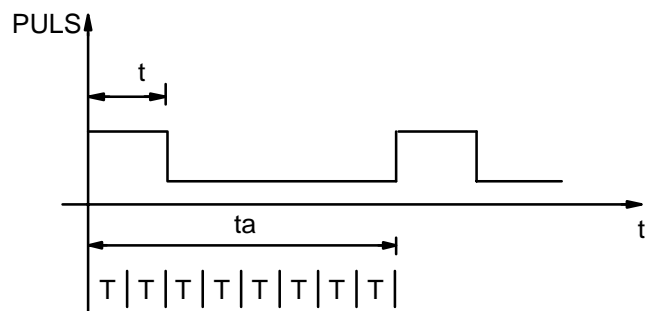
Die Eingänge und der Ausgang können weder gedoppelt noch negiert noch invertiert werden.

t/ta WORT Tastverhältnis

Am Eingang t/ta wird das gewünschte Tastverhältnis für das Ausgangssignal PULS vorgegeben. Dabei ist ta die Periodendauer des Signals am Ausgang PULS, und t ist die Zeit innerhalb der Periodendauer ta , während der das Ausgangssignal 1-Pegel annimmt. Der Vorgabewert für das gewünschte Tastverhältnis am Eingang t/ta muß in skaliertes Form angegeben werden. Dazu ist das gewünschte Tastverhältnis mit dem Wert 32767 zu multiplizieren und auf eine ganze Zahl zu runden. Der so erhaltene Zahlenwert wird dann am Eingang t/ta angegeben.

Randbedingung für $t: t > T$

D.h. die gewünschte Einschaltdauer des Ausgangssignals muß größer sein als die Zykluszeit des SPS-Programms.



Für die Vorgabe des Tastverhältnisses am Eingang t/ta gilt folgende Zusammenhang:

skalierter Wert am Eingang t/ta	ergibt Tastverhältnis am Ausgang PULS
negativer Wert	0 (0 %)
0 (0 * 32767)	0 (0 %)
.	.
.	.
16384 (0,5 * 32767)	0,5 (50 %)
.	.
.	.
32767 (1 * 32767)	1 (100 %)

ta/T WORT

Am Eingang ta/T wird die gewünschte Periodendauer ta für das Signal am Ausgang PULS angegeben. Die Periodendauer ta ist dabei auf die Zykluszeit T zu normieren.

Randbedingung für ta :

- ta muß ein ganzzahliges Vielfaches von T sein ($ta = n * T$)
- $ta \gg T > 0$; je größer ta im Verhältnis zu T ist, desto genauer wird das gewünschte Tastverhältnis eingehalten
Bsp.: $ta \geq 10 * T \rightarrow$ Ungenauigkeit des Tastverhältnisses am Ausgang PULS $\leq 10\%$.

Wird für ta/T der Wert $ta/T < 0$ vorgegeben, so ersetzt der Funktionsbaustein diesen unsinnigen Wert automatisch durch den Wert 32767.

PULS BINÄR

Am Ausgang PULS steht das pulsdauermodulierte Signal zur Verfügung.

Kombination des Bausteins PDM mit einem Regler
Wird der Funktionsbaustein PDM an den Ausgang eines Reglers angeschlossen um einen "schaltenden" Regler zu realisieren, so gelten folgende Randbedingungen:

- Periodendauer ta vom PDM = Abtastzeit des Reglers.
- Periode des Pulssignals muß synchron zur Periode der Abtastzeit des Reglers sein.

Diese Randbedingungen werden dadurch erfüllt, daß der Regler im gleichen SPS-Programm wie der PDM, aber innerhalb eines Laufzahlblocks, projiziert ist. Durch den Laufzahlblock wird die Abtastzeit des Reglers um ein ganzzahliges Vielfaches der Zykluszeit verlängert. Der Regler wird also innerhalb des Laufzahlblocks weniger oft bearbeitet als der PDM außerhalb des Laufzahlblocks.

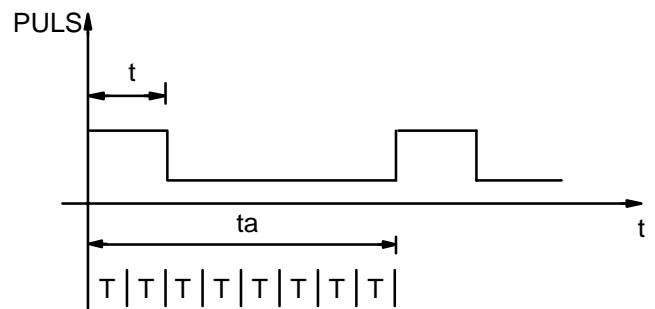
Beispiel:

Gewünscht:

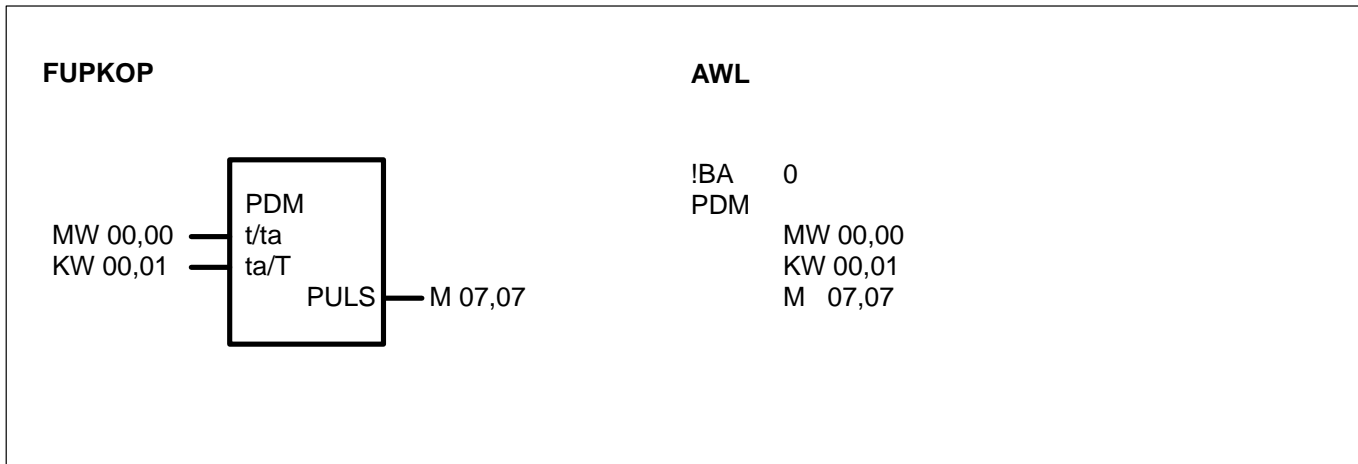
- Tastverhältnis: $t/ta = 0,25$ (25 %)
- Periodendauer: $ta = 800$ ms (nur ganzzahliges Vielfaches der SPS-Zykluszeit möglich)
- Zykluszeit: $T = 100$ ms

Vorzugebende Bausteinparameter:

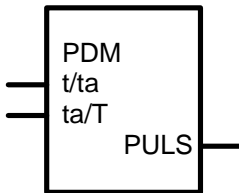
- Wert am Eingang t/ta : 8192 (0,25 * 32767)
- Wert am Eingang ta/T : 8 (800 ms/100 ms)



Beispiel



VE-FUP-Definition

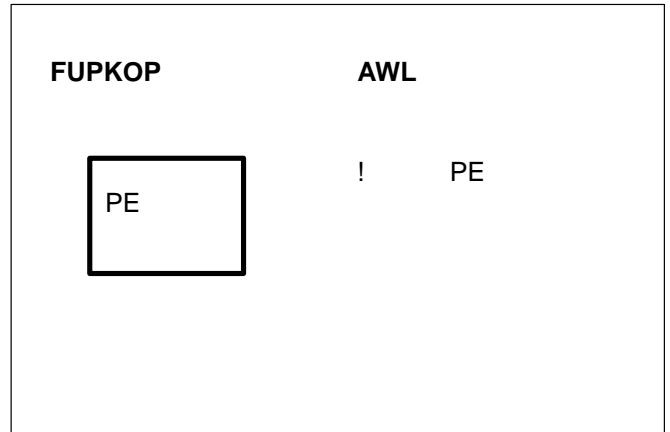


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
t/ta	E	W	N	P	J	0	0
ta/T	E	W	N	P	J	0	0
PULS	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	PDM			
00002		PP 0	t/ta	Eingang WORT (Tastverhältnis)
00003		PP 0	ta/T	Eingang WORT (Periodendauer)
00004		PP 0	PULS	Ausgang BINÄR

Das Programmende ist die Kennzeichnung für das Ende des SPS-Programms. Befehle, die nach dieser Enderkennung stehen werden von der SPS nicht bearbeitet.



Parameter

—

VE-Daten

Serie 90, 07 KP 62

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

2 µs

2 µs

Zusätzliche Laufzeit:

—

—

Aktualisierung der Ausgänge:

ja

ja

Anzahl der Vergangenheitswerte:

keine

keine

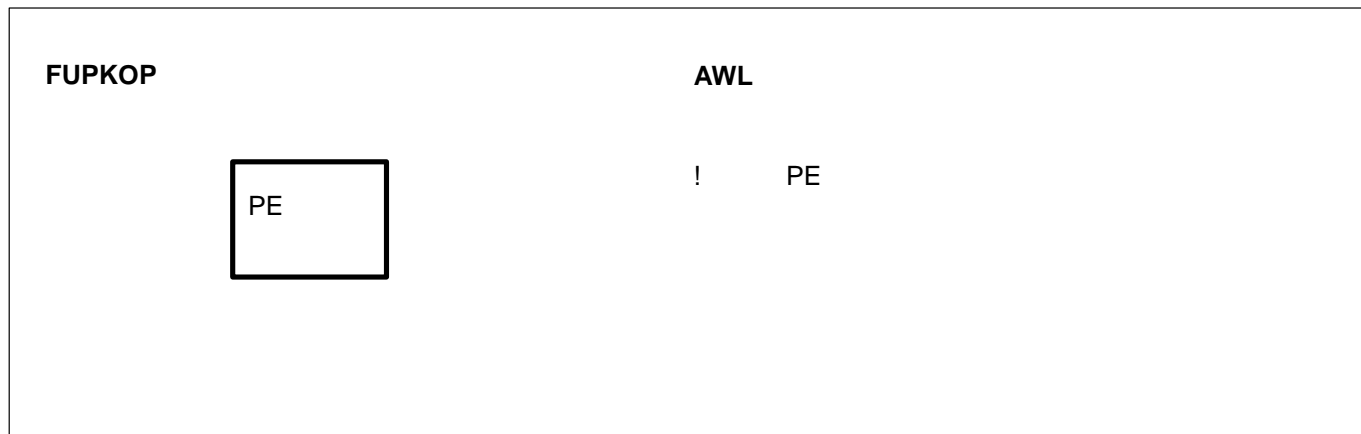
Verfügbar ab:

ABB Procontic CS31 / 907 PC 331
ABB Procontic T200 / 07 KP 62

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

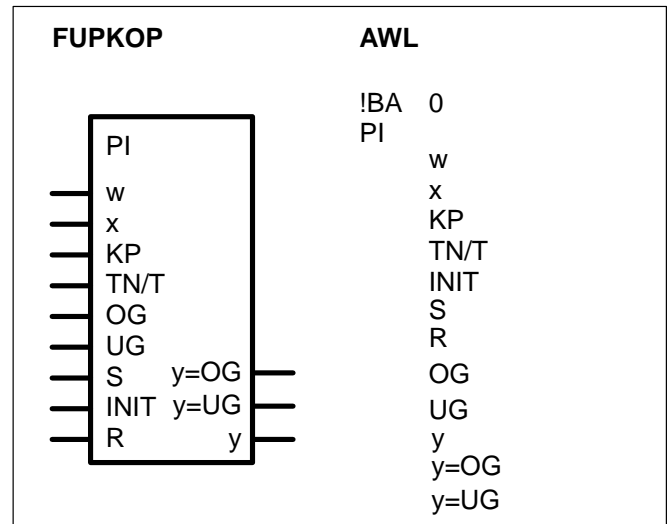
Beschreibung

Das Programmende ist die Kennzeichnung für das Ende des SPS-Programms. Befehle, die nach dieser Enderkennung stehen werden von der SPS nicht bearbeitet.

Beispiel**VE-FUP-Definition****VE-AWL-Definition**

00000 !PE

Der PI-Regler ändert den Wert an seinem Ausgang y (Stellgröße) solange, bis der Wert am Eingang x (Regelgröße) gleich dem Wert am Eingang w (Führungsgröße) ist.



Parameter

w	WORT	EW, AW, MW, KW	Führungsgröße (Sollwert)
x	WORT	EW, AW, MW, KW	Regelgröße (Istwert)
KP	WORT	EW, AW, MW, KW	Proportionalbeiwert; Angabe erfolgt in Prozent
TN/T	WORT	EW, AW, MW, KW	Nachstellzeit normiert auf SPS-Zykluszeit
OG	WORT	EW, AW, MW, KW	Obere Grenze für die Stellgröße y
UG	WORT	EW, AW, MW, KW	Untere Grenze für die Stellgröße y
S	BINÄR	E, A, M, S, K	Freigabe zum Setzen der Stellgröße y auf Initialwert INIT
INIT	WORT	EW, AW, MW, KW	Initialwert für die Stellgröße y
R	BINÄR	E, A, M, S, K	Rücksetzen der Stellgröße y auf den Wert 0
y=OG	BINÄR	A, M	Oberer Grenzwert ist erreicht
y=UG	BINÄR	A, M	Unterer Grenzwert ist erreicht
y	WORT	AW, MW	Ausgang für die Stellgröße y

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	280 µs	3600 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	4 Worte	6 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 2.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der PI-Regler ändert den Wert an seinem Ausgang y (Stellgröße) solange, bis der Wert am Eingang x (Regelgröße) gleich dem Wert am Eingang w (Führungsgröße) ist.

Regelalgorithmus: einfache Rechteckregel

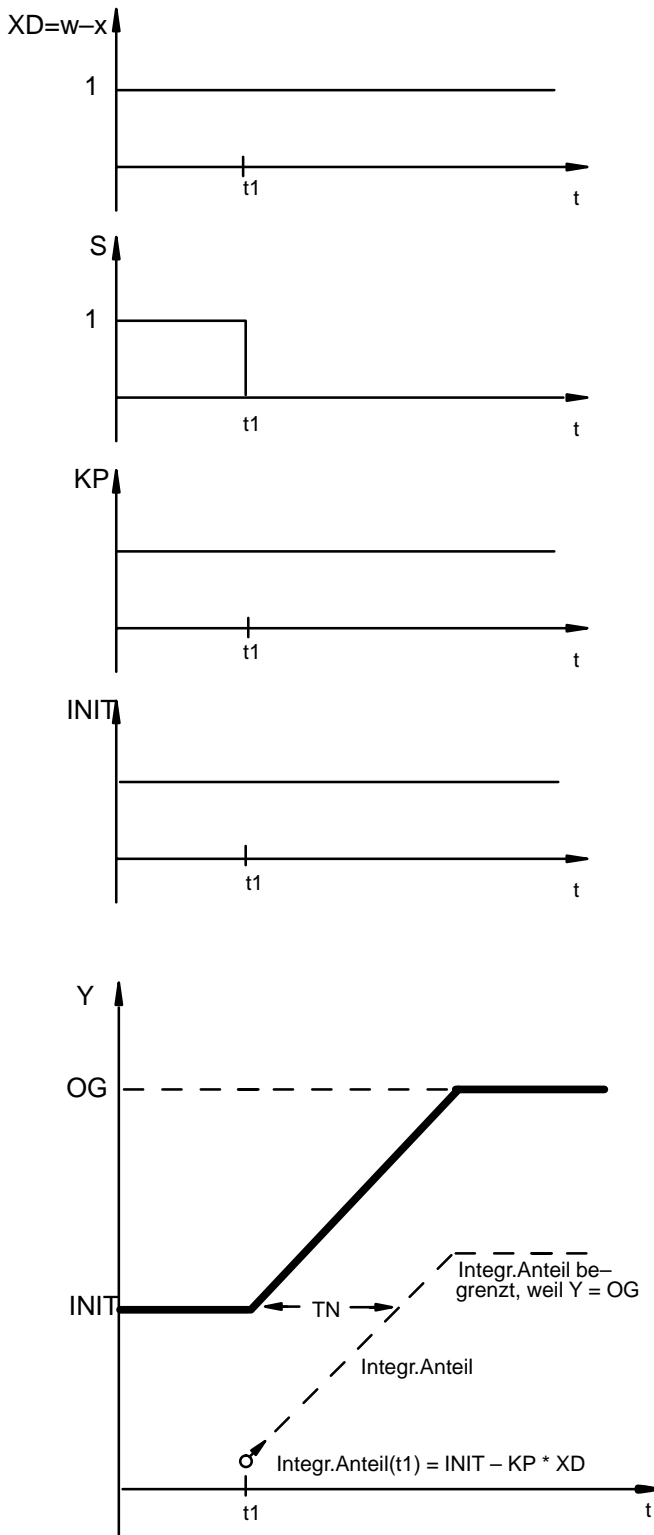
$$Y = \frac{KP}{100} * \frac{w-x}{TN/T} + YI(z-1) + \frac{KP}{100} * (w-x)$$

Dabei ist: YI(z-1) der Integral-Anteil aus dem vorhergehenden Programmzyklus

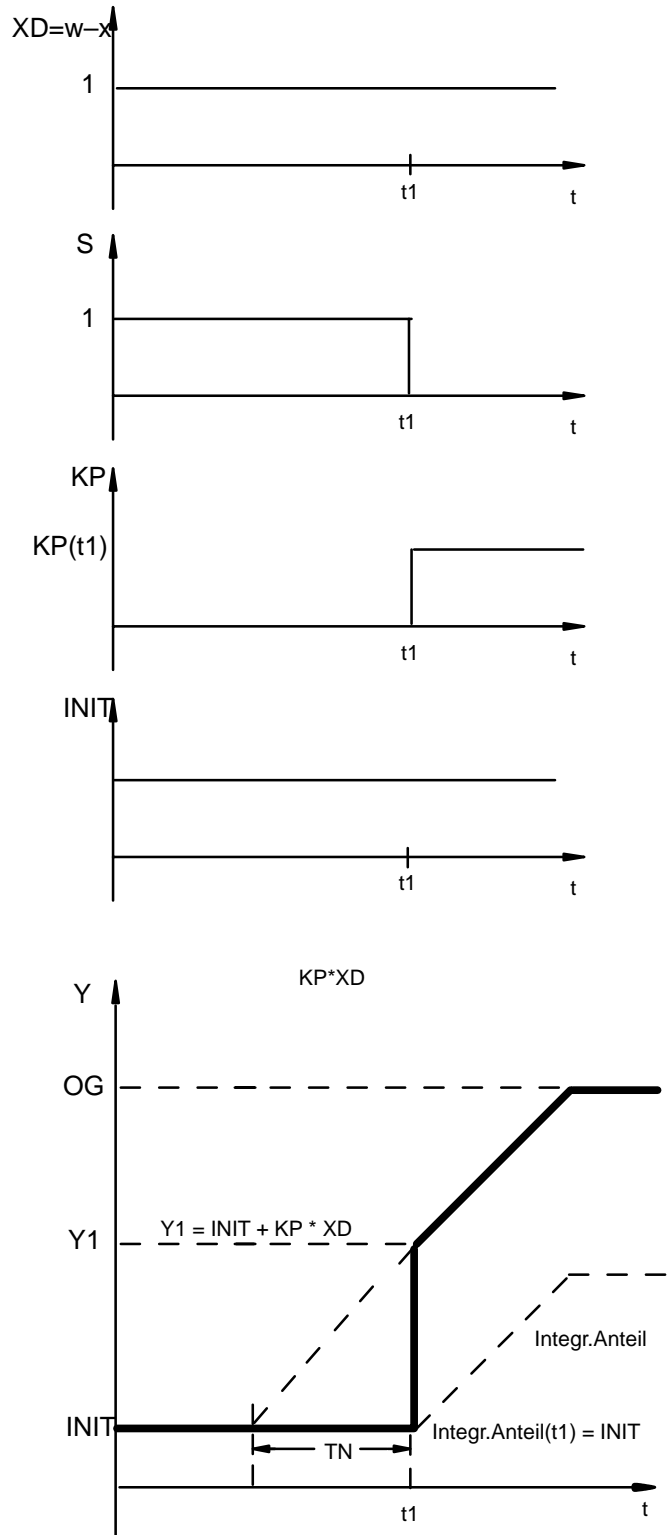
Übertragungsfunktion:

$$F(s) = KP * \left(1 + \frac{1}{s * TN}\right)$$

Die Ein- und Ausgänge sind weder doppelbar noch negierbar/invertierbar.



PI-Regler: Stoßfreier Übergang vom vorgegebenen Initialwert in den Regelbetrieb



PI-Regler: Stoßbehafteter Übergang vom vorgegebenen Initialwert in den Regelbetrieb

w WORT

Die Führungsgröße (Sollwert) wird am Eingang w vorgegeben.

x WORT

Die Regelgröße (Istwert) wird am Eingang x vorgegeben.

KP WORT

Der Proportionalbeiwert wird am Eingang KP angegeben. Die Angabe dieses Werts erfolgt in Prozent und kann positiv oder negativ sein.

Beispiel:

1	=	1	Prozent
55	=	55	Prozent
100	=	100	Prozent
1000	=	1000	Prozent
-500	=	-500	Prozent

- 1 Prozent heißt, daß der Baustein die Regeldifferenz mit dem Faktor 0,01 multipliziert (s. a. Regelalgorithmus).
- 100 Prozent heißt, daß der Baustein die Regeldifferenz mit dem Faktor 1 multipliziert (s. a. Regelalgorithmus).
- 1000 Prozent heißt, daß der Baustein die Regeldifferenz mit dem Faktor 10 multipliziert (s. a. Regelalgorithmus).

Proportionalbeiwerte größer als 1000 Prozent sind regelungstechnisch in der Regel nicht sinnvoll.

TN/T WORT

Die Nachstellzeit TN wird auf die SPS–Zykluszeit T normiert und am Eingang TN/T angegeben.

Wertebereich: $0 \leq TN/T \leq 328$

- Werden Werte vorgegeben, die außerhalb des zulässigen Wertebereichs liegen, so arbeitet die SPS grundsätzlich mit dem Wert 328.
- Eine große Nachstellzeit TN kann dadurch erreicht werden, daß auch die Zykluszeit T groß gewählt wird. Befindet sich der Baustein innerhalb eines Laufzahlblockes, so ist für ihn die Zykluszeit des Laufzahlblockes und nicht die Zykluszeit (KD 0,0) des SPS–Programmes maßgebend.

Begrenzung der Stellgröße y

OG WORT

UG WORT

Der Ausgang y (Stellgröße) des Reglers kann durch

- Vorgabe eines Grenzwertes am Eingang OG (Obere Grenze) auf einen Maximalwert begrenzt werden
- Vorgabe eines Grenzwertes am Eingang UG (Untere Grenze) auf einen Minimalwert begrenzt werden

Die obere und untere Grenze gilt *ebenfalls* für den regler-internen I–Anteil. D. h. der I–Anteil kann nur Werte zwischen oberer und unterer Grenze annehmen. Erreicht die Stellgröße y einen der beiden Grenzwerte, so wird der I–Anteil des Reglers *nicht* mehr verändert. Dadurch wird verhindert, daß der I–Anteil bei Begrenzung des Regler–Ausgangs y wegläuft, regelungstechnisch sinnlose Werte annimmt und u. U. erst nach sehr langer Zeit wieder in den Arbeitsbereich zurückkehrt. Dieses Verhalten eines Reglers wird auch als "spezielle Anti–Reset–Windup–Maßnahme (ARW)" bezeichnet.

Setzen und Rücksetzen des Reglers

S BINÄR

INIT WORT

R BINÄR

Setzen des Reglers auf einen Initialwert

- Mit einem 1–Signal am Eingang S (Set) wird der Ausgangs Y des Reglers auf den am Eingang INIT angegebenen Initialwert gesetzt.
- Ein 1–Signal am Eingang R (Reset) ist gleichbedeutend wie die Vorgabe des Initialwerts 0 (s. o.).

Stoßfreies Setzen/Rücksetzen

- Mit einem 1–Signal am binären Eingang S (Set) wird der Ausgang y des Reglers auf den am Eingang INIT angegebenen Initialwert gesetzt.
- Ein 1–Signal am Eingang R (Reset) ist gleichbedeutend wie die Vorgabe des Initialwerts 0.

Dabei erfolgt intern im Regler ein Abgleich auf den Initialwert. Der Abgleich ist eine Verschiebung des Reglerausgangs vom momentanen Wert auf den gewünschten Initialwert. Der Regler arbeitet jetzt von diesem Initialwert aus genau so weiter, wie er es vor der Verschiebung im alten Arbeitspunkt getan hätte, d. h. stoßfrei. Der I–Anteil des Reglers wird dabei so festgelegt, daß die Summe aus dem P–Anteil und I–Anteil gerade den Initialwert ergibt.

Vorteil des stoßfreien Setzens:

- Der Regelvorgang ab dem neuen Initialwert erfolgt stoßfrei.

Nachteil des stoßfreien Aufsetzens:

- Es gilt die Gleichung: $I_Anteil = INIT - P_Anteil$

Der I–Anteil wird dabei u. U. auf große Werte gesetzt, und es kann dann sehr lange dauern bis dieser regelungstechnisch "falsche" I–Anteil wieder abgebaut ist.

Stoßbehaftetes Setzen/Rücksetzen

- Mit einem 1–Signal am Eingang S (Set) wird der Ausgang y des Reglers auf den am Eingang INIT angegebenen Initialwert gesetzt.
- Ein 1–Signal am Eingang R (Reset) ist gleichbedeutend wie die Vorgabe des Initialwerts 0.

Beim stoßbehafteten Setzen bzw. Rücksetzen des Reglers wird der I–Anteil gleich dem Initialwert gesetzt. Der P–Anteil muß dazu während des Setzvorgangs unterdrückt werden.

Es gilt dabei: $I_Anteil = INIT$

Das stoßbehaftete Setzen auf einen Initialwert wird durch folgende Maßnahme während des Setzvorgangs erreicht:

- Vorgabe des Wertes 0 am Eingang KP.

Durch diese Maßnahme wird der P–Anteil des Reglers unwirksam. Der Regler–Ausgang y nimmt im Setz–Zyklus den Initialwert an.

Nach dem Setz–Zyklus wird der P–Anteil wieder freigegeben. Der Regler–Ausgang y macht vom Initialwert aus einen Sprung entsprechend dem P–Anteil des Reglers.

Vorteil des stoßbehafteten Setzens:

- Der I–Anteil wird nicht auf regelungstechnisch "falsche" Werte gesetzt.

Nachteil des stoßbehafteten Setzens:

- Keine Stoßfreiheit

y WORT

Am Ausgang y wird die Stellgröße y des Reglers ausgegeben.

$y=OG$ BINÄR

Am Ausgang $y=OG$ wird signalisiert, ob der Wert am Ausgang y den vorgegebenen oberen Grenzwert erreicht hat.

$y=OG = 0$ Grenzwert ist *nicht* erreicht.

$y=OG = 1$ Grenzwert ist erreicht.

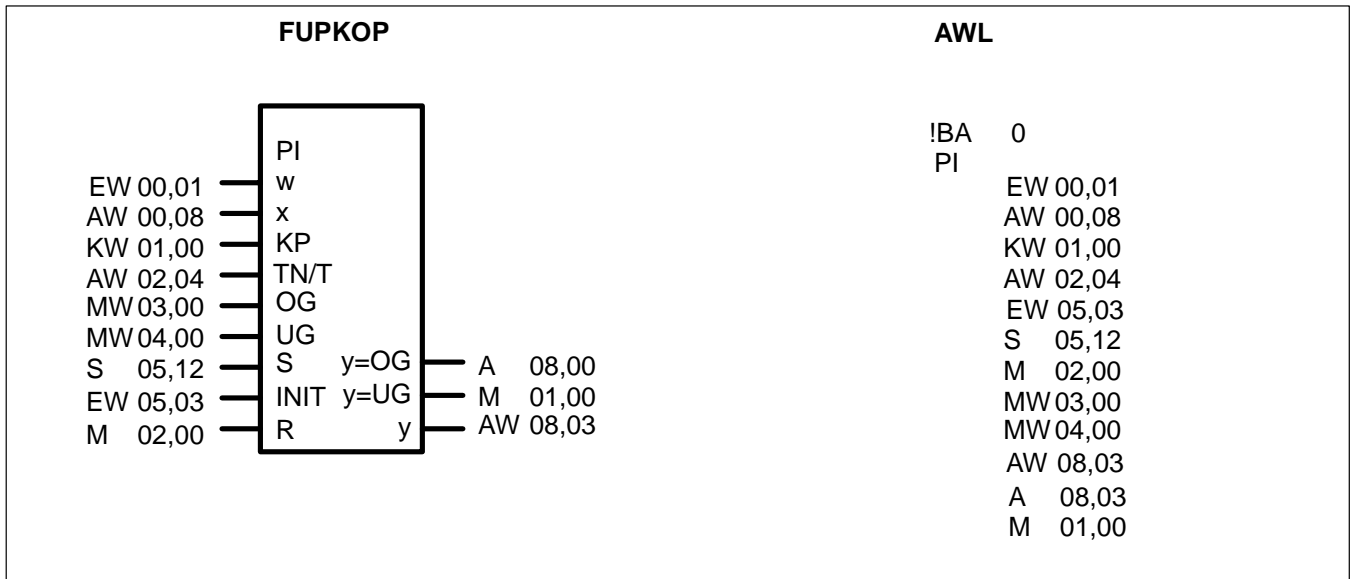
$y=UG$ BINÄR

Am Ausgang $y=UG$ wird signalisiert, ob der Wert am Ausgang y den vorgegebenen unteren Grenzwert erreicht hat.

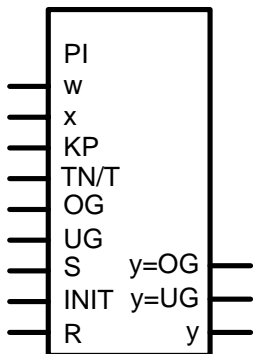
$y=UG = 0$ Grenzwert ist *nicht* erreicht.

$y=UG = 1$ Grenzwert ist erreicht.

Beispiel



VE-FUP-Definition

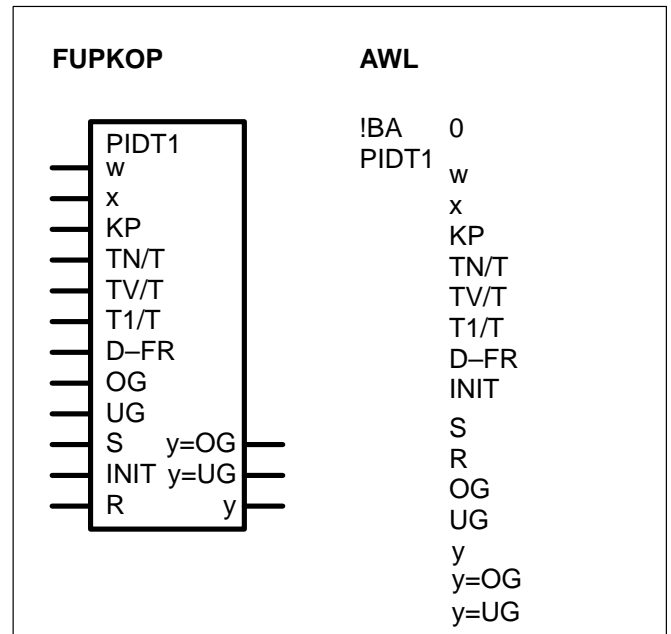


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
w	E	W	N	P	J	0	0
x	E	W	N	P	J	0	0
KP	E	W	N	P	J	0	0
TN/T	E	W	N	P	J	0	0
OG	E	W	N	P	J	0	0
UG	E	W	N	P	J	0	0
S	E	L	N	P	J	0	0
INIT	E	W	N	P	J	0	0
R	E	L	N	P	J	0	0
y=OG	A	L	N	P	J	0	0
y=UG	A	L	N	P	J	0	0
y	A	W	N	P	J	0	0

VE-AWL-Definition

	IBA	0	Nr	Baustein-Nr (vorbelegt 0)
00000	PI			
00001				
00002		PP 0	w	Eingang WORT (Führungsgröße)
00003		PP 0	x	Eingang WORT (Regelgröße)
00004		PP 0	KP	Eingang WORT (Proport. beiwert)
00005		PP 0	TN/T	Eingang WORT (Nachstellzeit)
00006		PP 0	INIT	Eingang WORT (Initialwert)
00007		PP 0	S	Eingang BIT (Setze auf INIT)
00008		PP 0	R	Eingang BIT (Rücksetze auf 0)
00009		PP 0	OG	Eingang WORT (Obere Grenze)
00010		PP 0	UG	Eingang WORT (untere Grenze)
00011		PP 0	y	Ausgang BINÄR (Stellgröße)
00012		PP 0	y=OG	Ausgang BINÄR (OG erreicht)
00013		PP 0	y=UG	Ausgang BINÄR (UG erreicht)

Der PI-Regler ändert seinen Ausgang y (Stellgröße) so lange, bis der Eingang x (Regelgröße) gleich dem Eingang w (Führungsgröße) ist.



Parameter

w	WORT	EW, AW, MW, KW	Führungsgröße (Sollwert)
x	WORT	EW, AW, MW, KW	Regelgröße (Istwert)
KP	WORT	EW, AW, MW, KW	Proportionalbeiwert, Angabe erfolgt in Prozent
TN/T	WORT	EW, AW, MW, KW	Nachstellzeit normiert auf SPS-Zykluszeit
TV/T	WORT	EW, AW, MW, KW	Vorhaltezeit normiert auf SPS-Zykluszeit
T1/T	WORT	EW, AW, MW, KW	Rücklaufzeit normiert auf SPS-Zykluszeit
D-FR	BINÄR	E, A, M, K	Freigabe DT1-Anteil
OG	WORT	EW, AW, MW, KW	Obere Grenze für die Stellgröße y
UG	WORT	EW, AW, MW, KW	Untere Grenze für die Stellgröße y
S	BINÄR	E, A, M, S, K	Freigabe zum Setzen auf Initialwert INIT
INIT	WORT	EW, AW, MW, KW	Initialwert für die Stellgröße y
R	BINÄR	E, A, M, S, K	Rücksetzen der Stellgröße y auf den Wert 0
y=OG	BINÄR	A, M	Oberer Grenzwert ist erreicht
y=UG	BINÄR	A, M	Unterer Grenzwert ist erreicht
y	WORT	AW, MW	Ausgang für Stellgröße y

VE-Daten

Laufzeit (Serie 90):	
Grundlaufzeit:	318 µs ohne DT1-Anteil, 564 µs mit DT1-Anteil
Zusätzliche Laufzeit:	nein
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	9 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331 Advant Controller 31, Serie 40...50

Beschreibung

Der PI-Regler ändert seinen Ausgang y (Stellgröße) solange, bis der Eingang x (Regelgröße) gleich dem Eingang w (Führungsgröße) ist.

Übertragungsfunktion:

$$F(s) = KP * \left(1 + \frac{1}{s * TN} + \frac{s * TV}{1 + (s * T1)} \right)$$

Regelalgorithmus: einfache Rechteckregel

$$y = \frac{KP * XD}{100} + \frac{KP}{100} * \frac{XD}{TN/TZ} + YI(z-1) + \frac{T1/TZ}{1 + (T1/TZ)} * (YDT1(z-1) + \frac{1}{T1/TZ} * \frac{TV}{TZ} * \frac{KP}{100} * (XD - XD(z-1)))$$

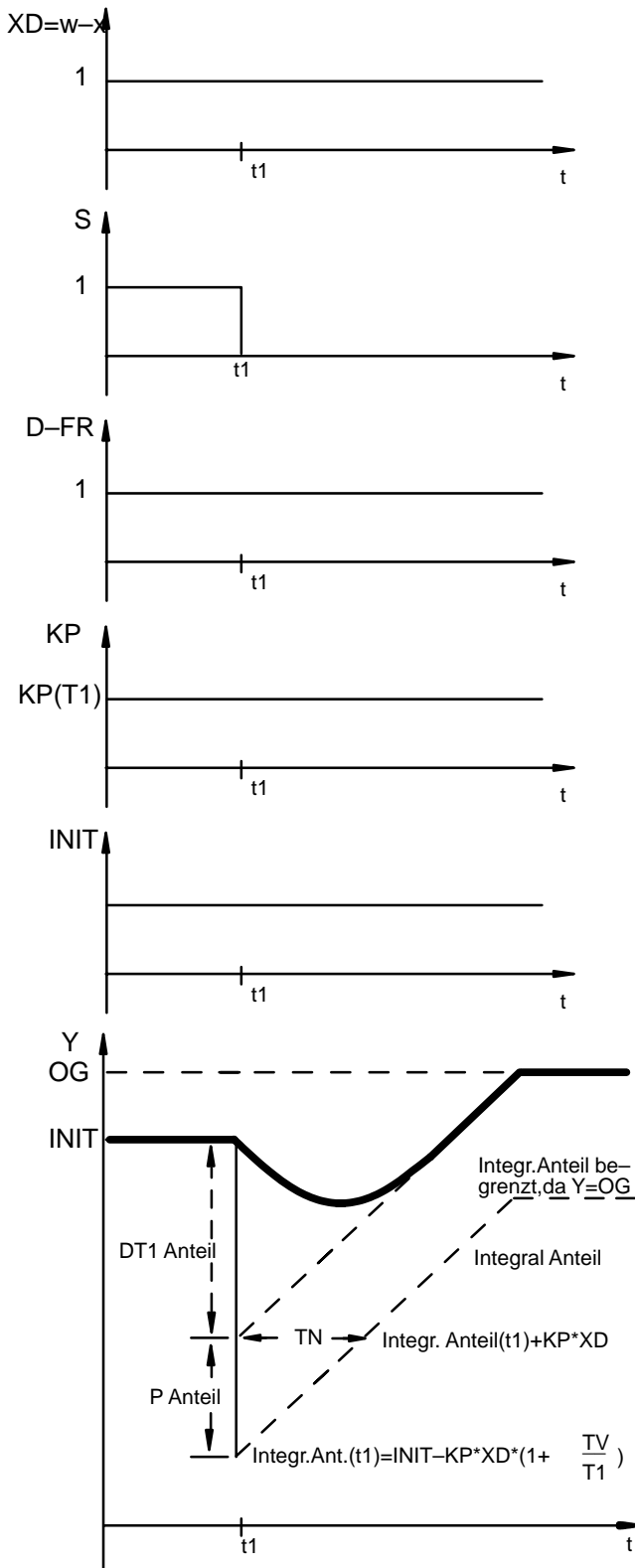
dabei ist:

$YI(z-1)$: Der Integralanteil aus dem vorhergehenden Programmzyklus

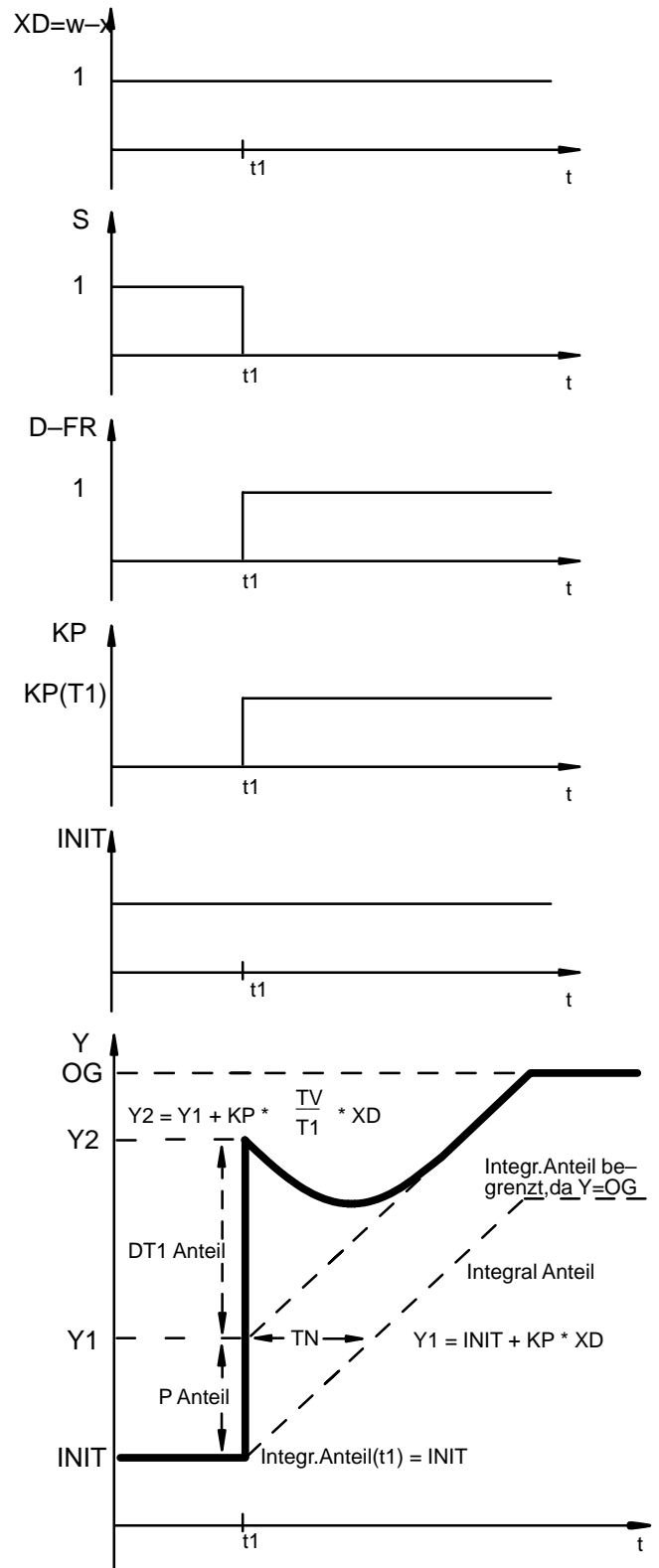
$YDT1(z-1)$: Der Differentialanteil aus dem vorhergehenden Programmzyklus

$XD(z-1)$: Regeldifferenz aus dem vorhergehenden Programmzyklus

Die Ein- und Ausgänge sind weder doppelbar noch negierbar/invertierbar.



PIDT1-Regler: Stoßfreier Übergang vom vorgegebenen Initialwert in den Regelbetrieb



PIDT1-Regler: Stoßbehafteter Übergang vom vorgegebenen Initialwert in den Regelbetrieb

w WORT

Die Führungsgröße (Sollwert) wird am Eingang w vorgegeben.

x WORT

Die Regelgröße (Istwert) wird am Eingang x vorgegeben.

KP WORT

Der Proportionalbeiwert wird am Eingang KP angegeben. Die Angabe dieses Werts erfolgt in Prozent und kann positiv oder negativ sein.

Beispiel:

1	=	1	Prozent
55	=	55	Prozent
100	=	100	Prozent
1000	=	1000	Prozent
-500	=	-500	Prozent

- 1 Prozent heißt, daß der Baustein die Regeldifferenz mit dem Faktor 0,01 multipliziert (s. a. Regelalgorithmus).
- 100 Prozent heißt, daß der Baustein die Regeldifferenz mit dem Faktor 1 multipliziert (s. a. Regelalgorithmus).
- 1000 Prozent heißt, daß der Baustein die Regeldifferenz mit dem Faktor 10 multipliziert (s. a. Regelalgorithmus).

Proportionalbeiwerte größer als 1000 Prozent sind regelungstechnisch in der Regel nicht sinnvoll.

TN/T WORT

Die Nachstellzeit TN wird auf die SPS-Zykluszeit T normiert und am Eingang TN/T angegeben ($0 < TN/T \leq 32767$).

Wertebereich: $0 \leq TN/T \leq 328$

- Werden Werte vorgegeben, die außerhalb des zulässigen Wertebereichs liegen, so arbeitet die SPS grundsätzlich mit dem Wert 328.
- Eine große Nachstellzeit TN kann dadurch erreicht werden, daß auch die Zykluszeit T groß gewählt wird. Befindet sich der Baustein innerhalb eines Laufzahlblockes, so ist für ihn die Zykluszeit des Laufzahlblockes und nicht die Zykluszeit (KD 0,0) des SPS-Programmes maßgebend.

TV/T WORT

Die Vorhaltezeit TV wird auf die SPS-Zykluszeit T normiert und am Eingang TV/T angegeben ($0 < TV/T \leq 32767$).

T1/T WORT

Die Rücklaufzeit T1 wird auf die SPS-Zykluszeit T normiert und am Eingang T1/T angegeben ($0 < T1/T \leq 32767$).

Die Rücklaufzeit ist die Zeit, in der der DT1-Anteil auf ca. 37 % seines Anfangswertes zurückgegangen ist.

Unzulässige Zeitangaben

Jeder Zeitwert wird auf den max. positiven Wert 32767 gesetzt, falls aus Versehen der Zeitwert am Eingang mit kleiner oder gleich "0" angegeben wird.

D-FR WORT

Mit dem Eingang D-FR kann der DT1-Anteil des Reglers zu- bzw. abgeschaltet werden.

D-FR	Bedeutung
0	DT1-Anteil ist abgeschaltet → reiner PI-Regler
1	DT1-Anteil ist zugeschaltet → PIDT1-Regler

In folgenden Fällen ist es aus regelungstechnischer Sicht oft störend und nicht sinnvoll, daß der DT1-Anteil wirksam ist:

- bei Einschaltvorgängen
- bei großen Regeldifferenzen
- beim Setzen des Reglers auf einen vorgegebenen Initialwert
- beim Rücksetzen des Reglers auf den Wert 0

Man kann außerhalb des Reglers einen Vergleich von Führungsgröße und Regelgröße vornehmen. In Abhängigkeit von diesem Vergleich kann über den Eingang D-FR der DT1-Anteil gezielt eingeschaltet oder ausgeschaltet werden.

Das Einschalten kann dabei z. B. darauf beschränkt werden, daß sich die Regeldifferenz innerhalb einer gewünschten Bandbreite befindet. D. h. der DT1-Anteil ist nur dann im Eingriff, wenn die Regelgröße sich innerhalb einer bestimmten Bandbreite um den Sollwert bewegt. Verläßt die Regelgröße dieses Toleranzband, so wird der DT1-Anteil ausgeschaltet.

Begrenzung der Stellgröße y**OG** WORT**UG** WORT

Der Ausgang y (Stellgröße) des Reglers kann durch

- Vorgabe eines Grenzwertes am Eingang OG (Obere Grenze) auf einen Maximalwert begrenzt werden;
- Vorgabe eines Grenzwertes am Eingang UG (Untere Grenze) auf einen Minimalwert begrenzt werden.

Die obere und untere Grenze gilt *ebenfalls* für den regler-internen I-Anteil. D. h. der I-Anteil kann nur Werte zwischen oberer und unterer Grenze annehmen. Erreicht die Stellgröße y einen der beiden Grenzwerte, so wird der I-Anteil des Reglers *nicht* mehr verändert.

Dadurch wird verhindert, daß der I–Anteil bei Begrenzung des Regler–Ausgangs y wegläuft, regelungstechnisch sinnlose Werte annimmt und u.U. erst nach sehr langer Zeit wieder in den Arbeitsbereich zurückkehrt. Dieses Verhalten eines Reglers wird auch als "spezielle Anti–Reset–Windup–Maßnahme (ARW)" bezeichnet.

Setzen und Rücksetzen des Reglers

S BINÄR
INIT WORT
R BINÄR

Setzen des Reglers auf einen Initialwert

- Mit einem 1–Signal am Eingang S (Set) wird der Ausgang y des Reglers auf den am Eingang INIT angegebenen Initialwert gesetzt.
- Ein 1–Signal am Eingang R (Reset) ist gleichbedeutend wie die Vorgabe des Initialwerts 0 (s. o.).

Stoßfreies Setzen/Rücksetzen

- Mit einem 1–Signal am binären Eingang S (Set) wird der Ausgang y des Reglers auf den am Eingang INIT angegebenen Initialwert gesetzt.
- Ein 1–Signal am Eingang R (Reset) ist gleichbedeutend wie die Vorgabe des Initialwerts 0.

Dabei erfolgt intern im Regler ein Abgleich auf den Initialwert. Der Abgleich ist eine Verschiebung des Reglerausgangs vom momentanen Wert auf den gewünschten Initialwert. Der Regler arbeitet jetzt von diesem Initialwert aus genau so weiter, wie er es vor der Verschiebung im alten Arbeitspunkt getan hätte, d.h. stoßfrei. Der I–Anteil des Reglers wird dabei so festgelegt, daß die Summe aus dem P–Anteil, I–Anteil und DT1–Anteil gerade den Initialwert ergibt.

Vorteil des stoßfreien Aufsetzens:

- Der Regelvorgang ab dem neuen Initialwert erfolgt stoßfrei.

Nachteil des stoßfreien Aufsetzens:

- Es gilt die Gleichung:
 $I\text{-Anteil} = \text{INIT} - P\text{-Anteil} - DT1\text{-Anteil}$

Der I–Anteil wird dabei u.U. auf große Werte gesetzt, und es kann dann sehr lange dauern bis dieser regelungstechnisch "falsche" I–Anteil wieder abgebaut ist.

Stoßbehaftetes Setzen/Rücksetzen

- Mit einem 1–Signal am Eingang S (Set) wird der Ausgang y des Reglers auf den am Eingang INIT angegebenen Initialwert gesetzt.

- Ein 1–Signal am Eingang R (Reset) ist gleichbedeutend wie die Vorgabe des Initialwerts 0.
- Beim stoßbehafteten Setzen bzw. Rücksetzen des Reglers wird der I–Anteil gleich dem Initialwert gesetzt. Der P– und der DT1–Anteil müssen dazu während des Setzvorgangs unterdrückt werden.

Es gilt dabei: $I\text{-Anteil} = \text{INIT}$

Das stoßbehaftete Setzen auf einen Initialwert wird durch folgende Maßnahmen während des Setzvorgangs erreicht:

- Ausschalten des DT1–Anteils über den Steuereingang D–FR und
- Vorgabe des Wertes 0 am Eingang KP

Durch diese Maßnahmen werden der P–Anteil und DT1–Anteil beim Setzen des Reglers unwirksam.

Der Regler–Ausgang nimmt im Setz–Zyklus den Initialwert an.

Nach dem Setz–Zyklus wird der P–Anteil und DT1–Anteil wieder freigegeben. Der Regler–Ausgang y macht vom Initialwert aus einen Sprung entsprechend dem P– und DT1–Anteil des Reglers.

Vorteil des stoßbehafteten Setzens:

- Der I–Anteil wird nicht auf regelungstechnisch "falsche" Werte gesetzt.

Nachteil des stoßbehafteten Setzens:

- Keine Stoßfreiheit

y WORT

Am Ausgang y wird die Stellgröße y des Reglers ausgegeben.

y=OG BINÄR

Am Ausgang $y=OG$ wird signalisiert, ob der Wert am Ausgang y den vorgegebenen oberen Grenzwert erreicht hat.

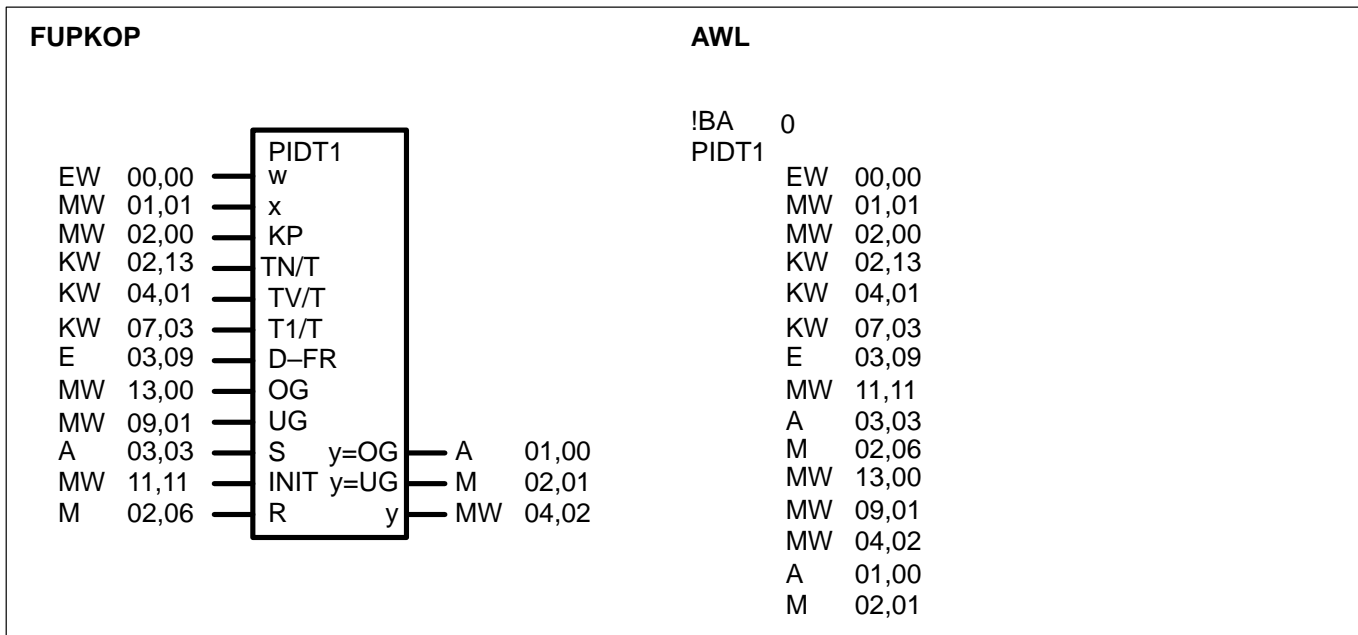
$y=OG = 0$ Grenzwert ist *nicht* erreicht.
 $y=OG = 1$ Grenzwert ist erreicht.

y=UG BINÄR

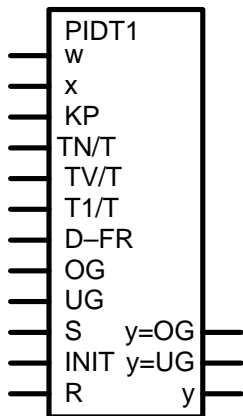
Am Ausgang $y=UG$ wird signalisiert, ob der Wert am Ausgang y den vorgegebenen unteren Grenzwert erreicht hat.

$y=UG = 0$ Grenzwert ist *nicht* erreicht.
 $y=UG = 1$ Grenzwert ist erreicht.

Beispiel



VE-FUP-Definition

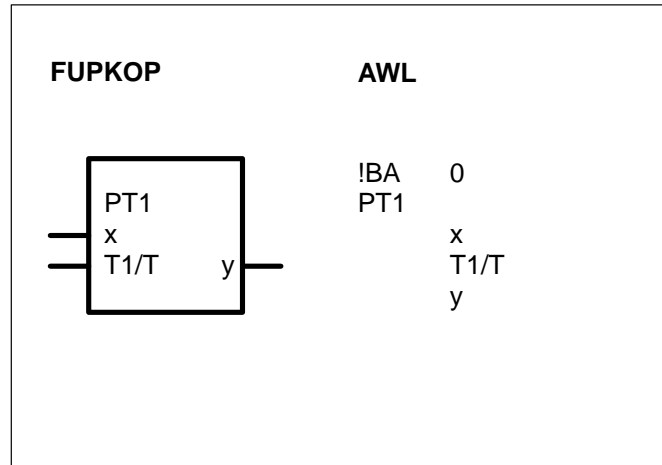


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
w	E	W	N	P	J	0	0
x	E	W	N	P	J	0	0
KP	E	W	N	P	J	0	0
TN/T	E	W	N	P	J	0	0
TV/T	E	W	N	P	J	0	0
T1/T	E	W	N	P	J	0	0
D-FR	E	L	N	P	J	0	0
OG	E	W	N	P	J	0	0
UG	E	W	N	P	J	0	0
S	E	L	N	P	J	0	0
INIT	E	W	N	P	J	0	0
R	E	L	N	P	J	0	0
y=OG	A	L	N	P	J	0	0
y=UG	A	L	N	P	J	0	0
y	A	W	N	P	J	0	0

VE–AWL–Definition

00000	!BA	0	Nr	Baustein–Nr. (vorbelegt 0)	
00001	PIDT1				
00002		PP 0	w	Eingang	WORT (Führungsgröße)
00003		PP 0	x	Eingang	WORT (Regelgröße)
00004		PP 0	KP	Eingang	WORT (Proport.beiwert)
00005		PP 0	TN/T	Eingang	WORT (Nachstellzeit)
00006		PP 0	TV/T	Eingang	WORT (Vorhaltezeit)
00007		PP 0	T1/T	Eingang	WORT (Rücklaufzeit)
00008		PP 0	D–FR	Eingang	BINÄR (Freigabe DT1–Ant)
00009		PP 0	INIT	Eingang	WORT (Initialwert)
00010		PP 0	S	Eingang	BINÄR (Setze auf INIT)
00011		PP 0	R	Eingang	BINÄR (Rücksetze auf 0)
00012		PP 0	OG	Eingang	WORT (Obere Grenze)
00013		PP 0	UG	Eingang	WORT (Untere grenze)
00014		PP 0	y	Ausgang	BINÄR (Stellgröße)
00015		PP 0	y=OG	Ausgang	BINÄR (OG erreicht)
00016		PP 0	y=UG	Ausgang	BINÄR (UG erreicht)

Der Funktionsbaustein realisiert ein Verzögerungsglied 1. Ordnung.



Parameter

x	WORT	EW, AW, MW, KW	Regelgröße
T1/T	WORT	EW, AW, MW, KW	Zeitkonstante
y	WORT	AW, MW	Stellgröße

VE-Daten

Laufzeit:	
Grundlaufzeit:	106 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	2 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

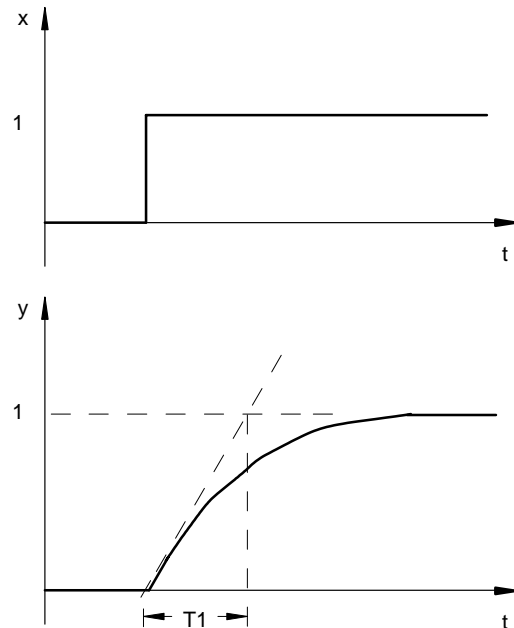
Der Funktionsbaustein realisiert ein Verzögerungsglied 1. Ordnung.

Übertragungsfunktion:

$$F = \frac{1}{1 + T1 * s}$$

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar.

Übergangsfunktion:



x WORT

Am Eingang X wird die Regelgröße vorgegeben.

T1/T WORT

Am Eingang T1/T wird die Verzögerungszeit T1 vorgegeben. Dabei ist die Verzögerungszeit T1 auf die Zykluszeit T zu normieren.

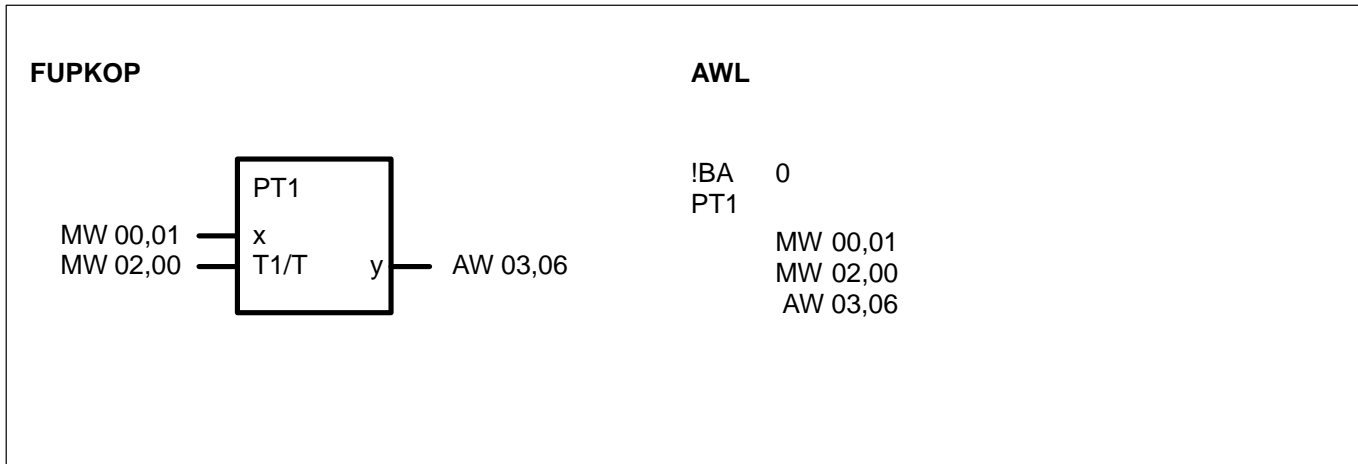
Es muß sein: $T1/T \geq 0$

Wird fälschlicherweise ein negativer Zeitwert vorgegeben, so setzt die SPS automatisch für T1/T den Wert 32767 ein.

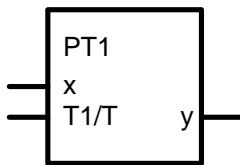
y WORT

Am Ausgang y wird das Ergebnis des Verzögerungsgliedes (Stellgröße) ausgegeben.

Beispiel



VE-FUP-Definition

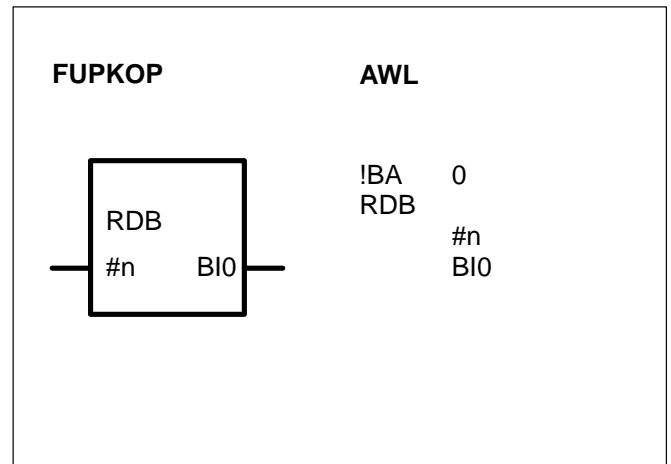


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
x	E	W	N	P	J	0	0
T1/T	E	W	N	P	J	0	0
y	A	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	PT1			
00002		PP 0	x	Eingang WORT (Regelgröße)
00003		PP 0	T1/T	Eingang WORT (Zeitkonstante)
00004		PP 0	y	Ausgang WORT (Stellgröße)

Der Funktionsbaustein liest Binärwerte aus seinem Vergangenheitswertespeicher und weist diese den Ausgängen BI0 ... BIn-1 zu. Die Binärwerte werden zuvor mit dem Baustein WRB in den Vergangenheitswertespeicher geschrieben.



Parameter

#n	DIREKTE KONSTANTE	#, #H	Anzahl der Ausgänge BI0 ... BIn-1
BI0	BINÄR	A, M	Ausgang für die Binärwerte, doppelbar (BI0 ... BIn-1)

VE–Daten

Laufzeit:		
Grundlaufzeit:		48 µs. Die Laufzeit des Bausteins WRB ist hier schon mit enthalten.
Zusätzliche Laufzeit:		14 µs pro Variable
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		#n geradzahlig → #n/2 Worte; #n ungeradzahlig → (#n + 1)/2 Worte
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein liest Binärwerte aus seinem Vergangenheitswertespeicher und weist diese den Ausgängen BI0 ... BIn-1 zu. Die Binärwerte werden zuvor mit dem Baustein WRB in den Vergangenheitswertespeicher geschrieben.

Die Funktionsbausteine RDB und WRB treten immer paarweise auf. Der Baustein WRB schreibt die gewünschten Werte in den Vergangenheitswertespeicher. Der Baustein RDB liest sie wieder aus.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der Ausgänge BI0 ... BIn-1 angegeben. Die Angabe erfolgt als direkte Konstante.

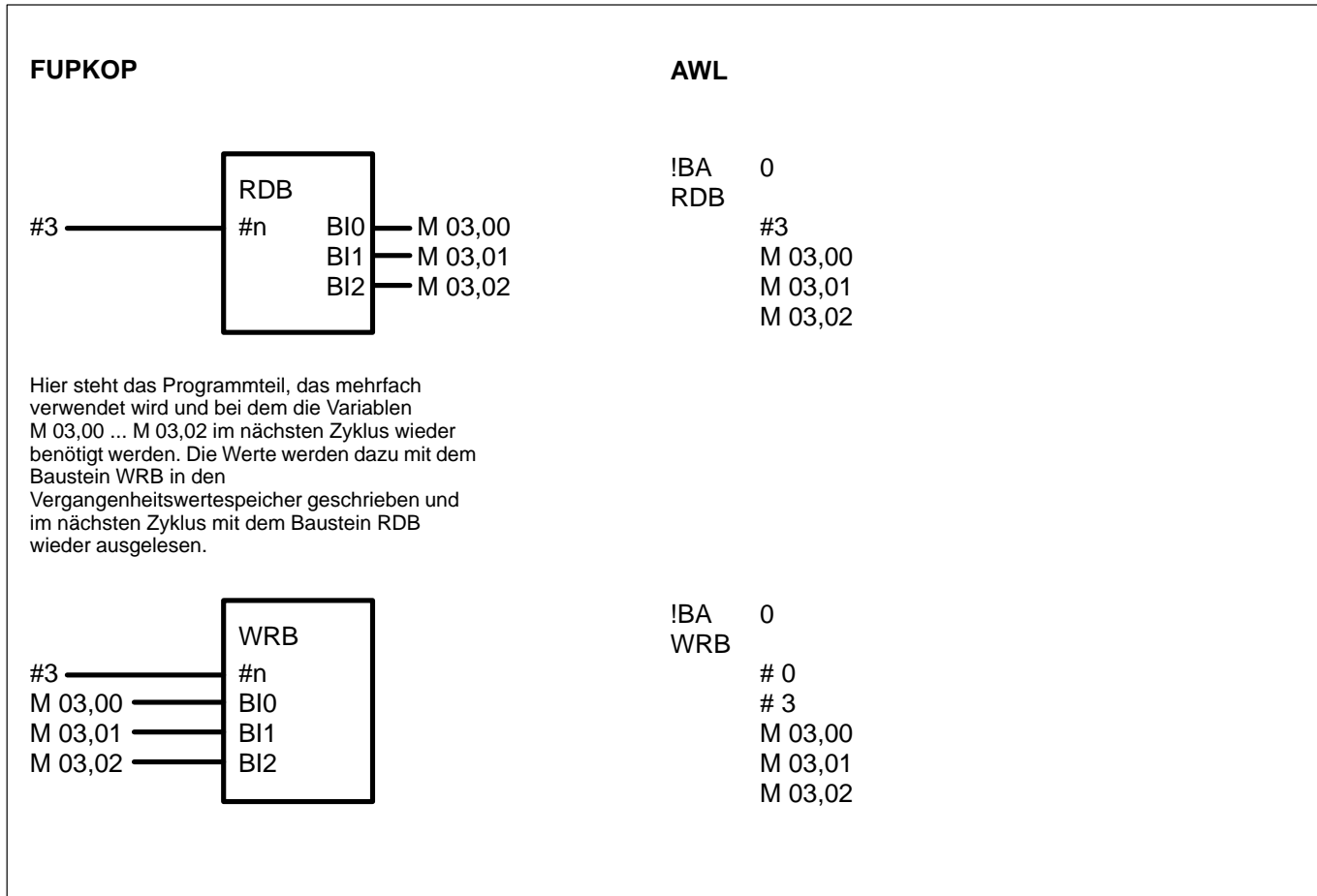
Hinweis:

Der am Eingang #n angegebene Wert muß auch mit der Anzahl der Eingänge des zugehörigen Bausteins WRB übereinstimmen.

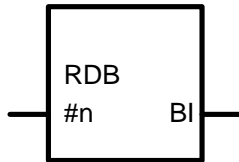
BI0 ... BIn-1 BINÄR

Der Ausgang BI0 ist doppelbar (BI0 ... BIn-1). Der Baustein weist die aus dem Vergangenheitswertespeicher gelesenen Werte den Ausgängen BI0 ... BIn-1 zu.

Beispiel



VE–FUP–Definition

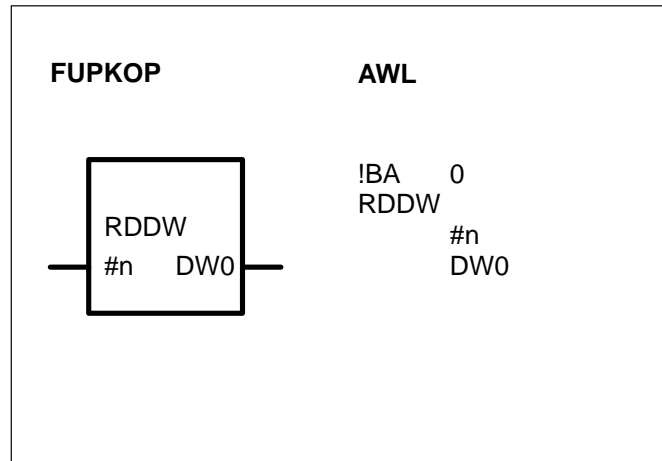


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
BI	A	L	N	P	J	1	0

VE–AWL–Definition

00000	!BA	0	Nr	Baustein–Nr. (vorbelegt 0)
00001	RDB			
00002		PP 0	#n	# DIREKTE KONSTANTE (Anzahl Bit)
	[1		
00003		PP 1	BI	Ausgang BINÄR
]	1		

Der Funktionsbaustein liest Doppelwortwerte aus seinem Vergangenheitswertespeicher und weist diese den Ausgängen DW0 ... DWn-1 zu. Die Doppelwortwerte werden zuvor mit dem Baustein WRDW in den Vergangenheitswertespeicher geschrieben.



Parameter

#n	DIREKTE KONSTANTE	#, #H	Anzahl der Ausgänge DW0 ... DWn-1
DW0	DOPPELWORT	MD	Ausgang für die Doppelwortwerte, doppelbar (DW0 ... DWn-1)

VE-Daten

Laufzeit:		
Grundlaufzeit:	34 µs.	Die Laufzeit des Bausteins WRDW ist hier schon mit enthalten.
Zusätzliche Laufzeit:	28 µs	pro zusätzlichem Ausgang DW1 ... DWn-1
Aktualisierung der Ausgänge:	ja	
Anzahl der Vergangenheitswerte:	2 * #n	Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	

Beschreibung

Der Funktionsbaustein liest Doppelwortwerte aus seinem Vergangenheitswertespeicher und weist diese den Ausgängen DW0 ... DWn-1 zu. Die Doppelwortwerte werden zuvor mit dem Baustein WRDW in den Vergangenheitswertespeicher geschrieben.

Die Funktionsbausteine RDDW und WRDW treten immer paarweise auf. Der Baustein WRDW schreibt die gewünschten Werte in den Vergangenheitswertespeicher. Der Baustein RDDW liest sie wieder aus.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der Ausgänge DW0 ... DWn-1 angegeben. Die Angabe erfolgt als direkte Konstante.

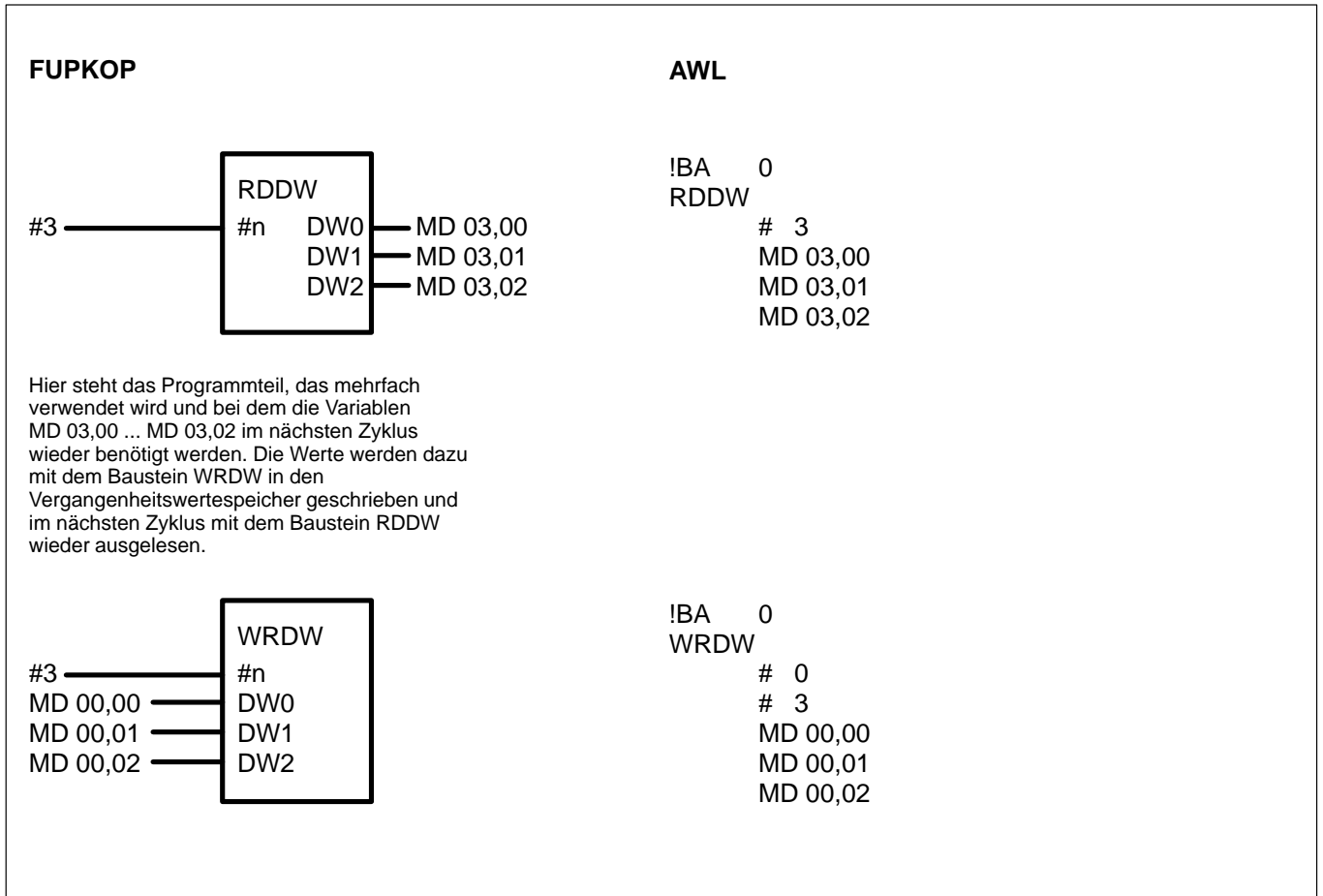
Hinweis:

Der am Eingang #n angegebene Wert muß auch mit der Anzahl der Eingänge des zugehörigen Bausteins WRDW übereinstimmen.

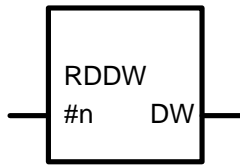
DW0 ... DWn-1 DOPPELWORT

Der Ausgang DW0 ist doppelbar (DW0 ... DWn-1). Der Baustein weist die aus dem Vergangenheitswertespeicher gelesenen Werte den Ausgängen DW0 ... DWn-1 zu.

Beispiel



VE-FUP-Definition

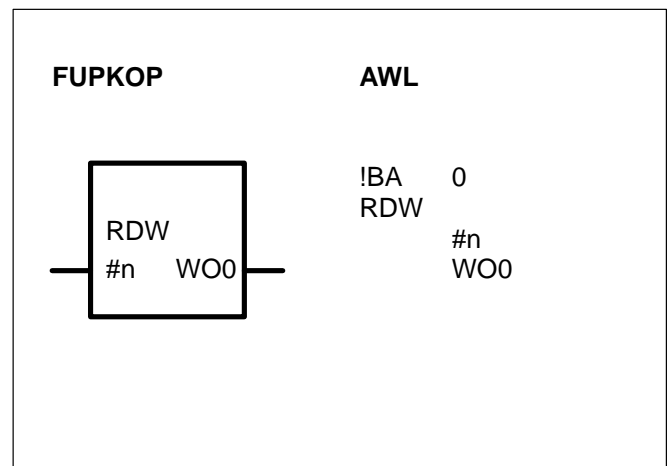


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
DW	A	D	N	P	J	1	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	RDDW			
00002	[PP 0 1	#n	# KONSTANTE (Anzahl Doppelworte)
00003]	PP 1 1	DW	Ausgang DOPPELWORT

Der Funktionsbaustein liest Wort–Werte aus seinem Vergangenheitswertespeicher und weist diese den Ausgängen WO0 ... WOn–1 zu. Die Wort–Werte werden zuvor mit dem Baustein WRW in den Vergangenheitswertespeicher geschrieben.



Parameter

#n	DIREKTE KONSTANTE	#, #H	Anzahl der Ausgänge WO0 ... WOn–1
WO0	WORT	AW, MW	Ausgang für die Wort–Werte, doppelbar (WO0 ... WOn–1)

VE–Daten

Laufzeit:		
Grundlaufzeit:		38 µs. Die Laufzeit des Bausteins WRW ist hier schon mit enthalten.
Zusätzliche Laufzeit:		19 µs pro zusätzlichem Ausgang WO1 ... WOn–1
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		#n Worte
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein liest Wort–Werte aus seinem Vergangenheitswertespeicher und weist diese den Ausgängen WO0 ... WOn–1 zu. Die Wort–Werte werden zuvor mit dem Baustein WRW in den Vergangenheitswertespeicher geschrieben.

Die Funktionsbausteine RDW und WRW treten immer paarweise auf. Der Baustein WRW schreibt die gewünschten Werte in den Vergangenheitswertespeicher. Der Baustein RDW liest sie wieder aus.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der Ausgänge WO0 ... WOn–1 angegeben. Die Angabe erfolgt als direkte Konstante.

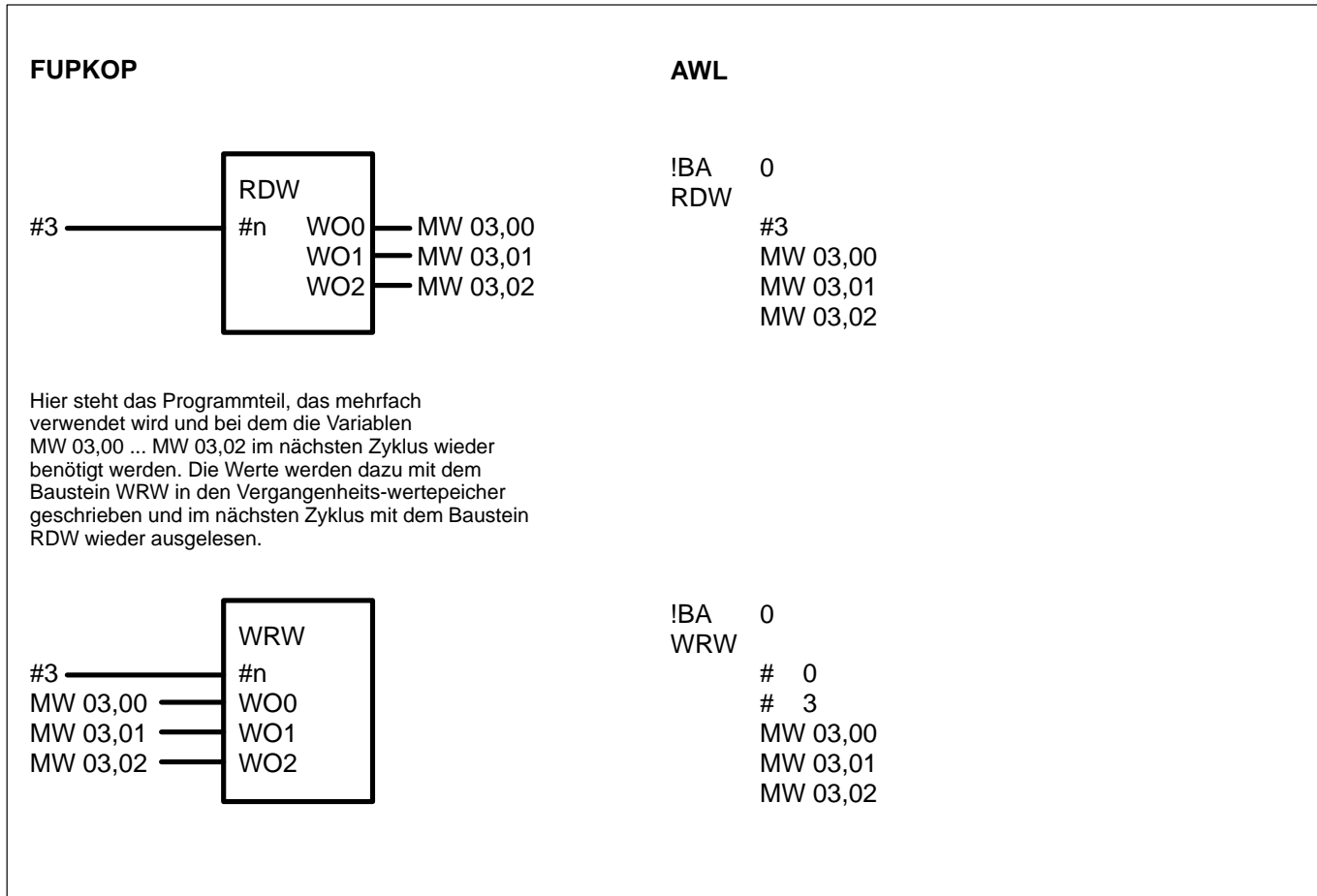
Hinweis:

Der am Eingang #n angegebene Wert muß mit der Anzahl der Eingänge des zugehörigen Bausteins WRW übereinstimmen.

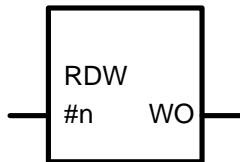
WO0 ... WOn–1 WORT

Der Ausgang WO0 ist doppelbar (WO0 ... WOn–1). Der Baustein weist die aus dem Vergangenheitswertespeicher gelesenen Werte den Ausgängen WO0 ... WOn–1 zu.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
WO	A	W	N	P	J	1	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr (vorbelegt 0)
00001	RDW			
00002		PP 0	#n	DIREKTE KONSTANTE
	[1		
00003		PP 1	WO	Ausgang
]	1		

Dieses Verknüpfungselement realisiert einen Binär-Speicher mit der Eigenschaft "dominierend Setzen".

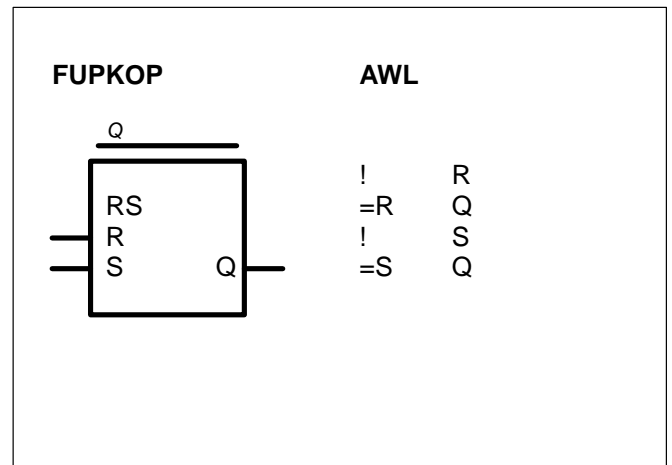
Der Operand Q, der die Speicherfunktion übernimmt, wird über dem Schaltzeichen eingetragen. Der Zustand des Operanden Q wird dem Operanden am Ausgang Q zugewiesen.

Zustand 1 am Eingang R setzt den Operanden Q zurück auf Zustand 0.

Zustand 1 am Eingang S setzt den Operanden Q auf Zustand 1.

Gleichzeitiger 1-Zustand am Eingang S und R setzt den Operanden Q auf Zustand 1 (dominierend Setzen).

Zustand 0 am Eingang S oder R hat keinen Einfluß auf den Operanden Q.



Parameter

R	BINÄR	E, M, A, S, K	Rücksetz-Eingang
S	BINÄR	E, M, A, S, K	Setz-Eingang
Q	BINÄR	M, A	Flip-Flop-Ausgang

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	8,6 µs	30 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Operand Q, der die Speicherfunktion übernimmt, wird über dem Schaltzeichen eingetragen. Der Zustand des Operanden Q wird dem Operanden am Ausgang Q zugewiesen.

Zustand 1 am Eingang R setzt den Operanden Q zurück auf Zustand 0.

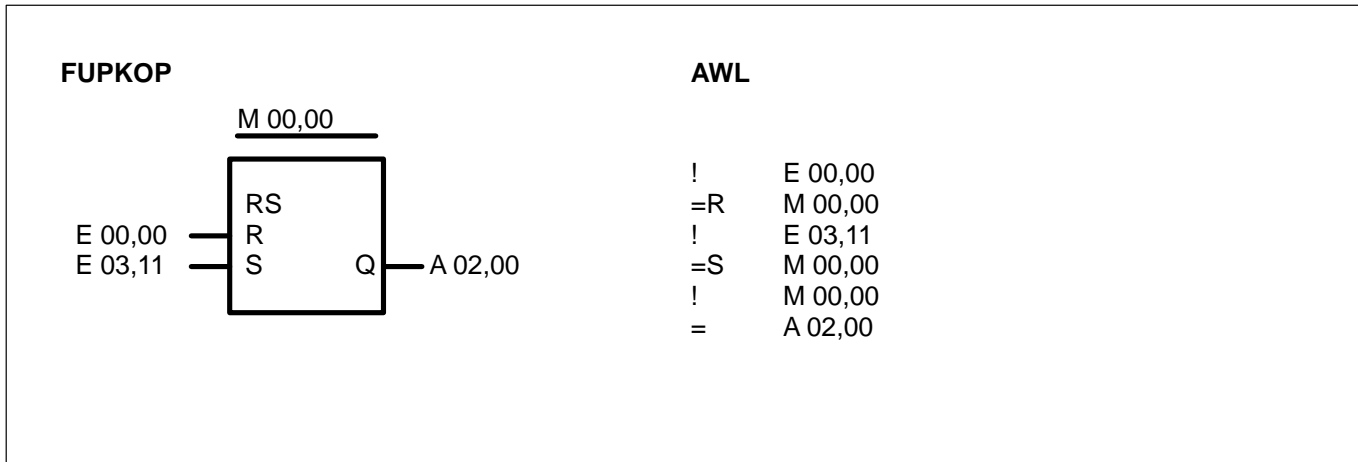
Zustand 1 am Eingang S setzt den Operanden Q auf Zustand 1.

Gleichzeitiger 1-Zustand am Eingang S und R setzt den Operanden Q auf Zustand 1 (dominierend Setzen).

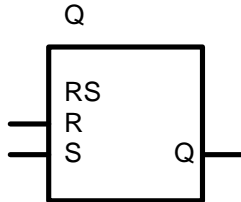
Zustand 0 am Eingang S oder R hat keinen Einfluß auf den Operanden Q.

Die Eingänge und der Ausgang sind nicht doppelbar. Die Eingänge R und S sind invertierbar.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
R	E	L	J	P	J	0	0
S	E	L	J	P	J	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

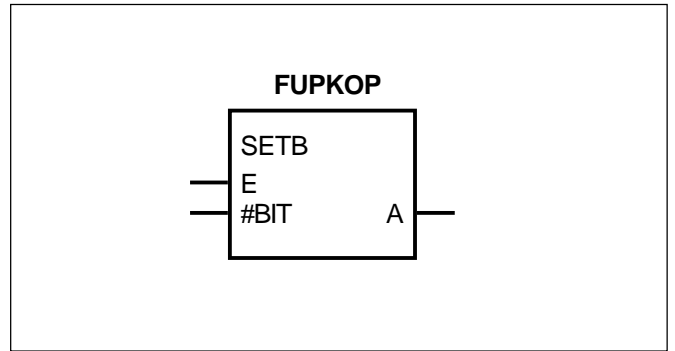
00000	!	PP 0	R	Eingang BINÄR
00002	=R	PP 0	Q	Ausgang BINÄR
00004	!	PP 0	S	Eingang BINÄR
00006	=S	PP 0	Q	Ausgang BINÄR

Das Bit am Eingang #Bit wird im Wort/Doppelwort A in Abhängigkeit vom Eingang E gesetzt (E = 1), bzw. zurückgesetzt (E = 0).

Zulässige Werte am Eingang #BIT: 0...31

Wird am Ausgang A eine Wortvariable eingegeben und #BIT ist größer als 15, so wird das entsprechende Bit der folgenden Wortvariablen gesetzt bzw. zurückgesetzt.

Beispiel: A = MW 00,00 und #BIT = 16 → Bit 0 von MW 00,01 wird in Abhängigkeit von Eingang E gesetzt bzw. zurückgesetzt.



Parameter

E	BINÄR	E, K, M, S	Eingang
#BIT	DIREKTE KONSTANTE	#, #H	Bitposition; zulässige Werte: 0...31
A	WORT DOPPELWORT	AW, EW, MW MD	Ausgang

VE-Daten

Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	07 KR 91 R2 Index c, 07 KT 92 R2 Index c, 07 KT 93 R2 Index f, 07 KT 94

Beschreibung

Das Bit am Eingang #BIT wird im Wort/Doppelwort A in Abhängigkeit vom Eingang E gesetzt (E = 1) bzw. zurückgesetzt (E = 0).

E BINÄR

Bei E = 0 wird das entsprechende Bit im Ausgang A zurückgesetzt, bei E = 1 wird es gesetzt.

#BIT DIREKTE KONSTANTE (#, #H)

Am Eingang #BIT wird die Position des entsprechenden Bits angegeben.

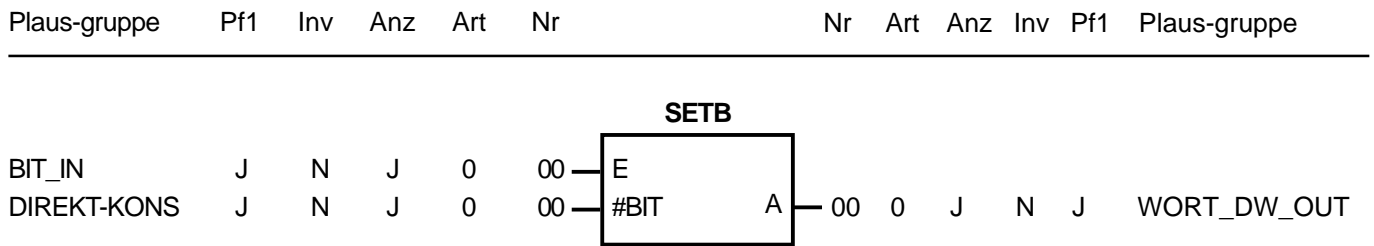
Zulässige Werte: 0...31

A WORT/DOPPELWORT

Am Ausgang A wird das entsprechende Bit gesetzt bzw. zurückgesetzt. Wird am Ausgang A eine Wortvariable angegeben und #BIT ist größer als 15 wird das entsprechende Bit der folgenden Wortvariablen gesetzt bzw. zurückgesetzt.

Beispiel: A = MW 00,02 und #BIT = 18 → Bit 2 von MW 00,03 wird in Abhängigkeit von Eingang E gesetzt bzw. zurückgesetzt.

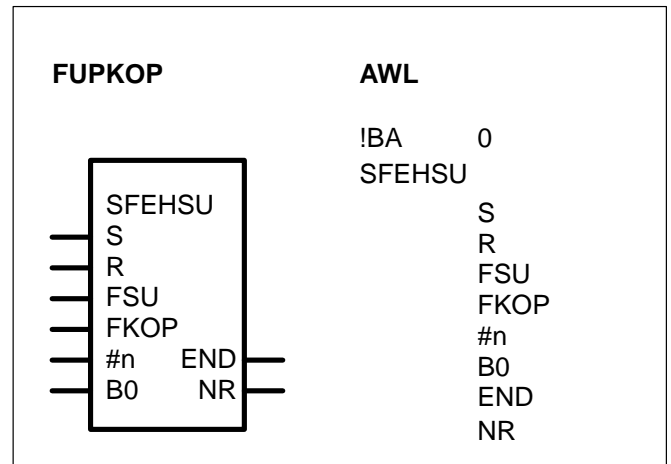
Grafik



Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	SETB		
	00001	PP	00	E
	00002	PP	00	#BIT
	00003	PP	00	A
	00004			

Der Funktionsbaustein durchsucht der Reihe nach eine Liste von Binär-Variablen (E,A,M, S) nach gesetzten Binär-Variablen. Wird eine gesetzte Binär-Variable gefunden, so wird deren Nummer am Ausgang NR ausgegeben. Der Baustein durchsucht dabei nicht direkt die Eingangsliste nach den gesetzten Binär-Variablen, sondern deren Abbild, das er in einer internen Liste speichernd ablegt. Nachdem in der internen Liste eine gesetzte Binär-Variable gefunden und deren Nummer am Ausgang NR ausgegeben wurde, wird diese Binär-Variable in der internen Liste gelöscht.



Parameter

S	BINÄR	E, A, M, S, K	Set
R	BINÄR	E, A, M, S, K	Reset
FSU	BINÄR	E, A, M, S, K	Freigabe suchen
FKOP	BINÄR	E, A, M, S, K	Freigabe Kopieren
#n	DIREKTE KONSTANTE	#, #H	Anzahl Binär-Variablen
B0	BINÄR	E, A, M, S, K	Liste der Binär-Variablen; Eingang ist doppelbar
END	BINÄR	A, M	Listenende erreicht
NR	WORT	AW, MW	Listen-Nummer der gefundenen Variablen

VE-Daten

Laufzeit:

Grundlaufzeit:	83 µs
Zusätzliche Laufzeit:	22 µs pro Eingang B0...B _{n-1}
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	3 Worte + #n/16 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein durchsucht der Reihe nach eine Liste von Binär-Variablen (E,A,M, S) nach gesetzten Binär-Variablen. Wird eine gesetzte Binär-Variable gefunden, so wird deren Nummer am Ausgang NR ausgegeben. Der Baustein durchsucht dabei nicht direkt die Eingangsliste nach den gesetzten Binär-Variablen, sondern deren Abbild, das er in einer internen Liste speichernd ablegt. Nachdem in der internen Liste eine gesetzte Binär-Variable gefunden und deren Nummer am Ausgang NR ausgegeben wurde, wird diese Binär-Variable in der internen Liste gelöscht.

Die Ein- und Ausgänge sind nicht negierbar. Der Eingang B0 ist doppelbar (B0 ... B_{n-1}).

S BINÄR

Ein 1-Signal am Eingang S bewirkt:

- alle gesetzten Binär-Variablen der Eingangs-Liste werden zusätzlich in die interne Liste eingetragen (mit der internen Liste verodert).
- der Baustein wird auf einen Suchlauf vom Anfang der internen Liste vorbereitet, d. h. der Zeiger der internen Liste wird auf deren Anfang gesetzt.
- der Ausgang END (Listenende erreicht) wird zu 1 gesetzt, der Ausgang NR (Nummer der Binär-Variablen) wird zu 0 gesetzt.
- wurde beim vorhergehenden Suchlauf in der internen Liste eine gesetzte Variable gefunden, so wird diese in der internen Liste gelöscht, um eine Doppelmeldung zu vermeiden.

R BINÄR

- Bei einem 1–Signal am Eingang R werden alle gesetzten Binär–Variablen in der internen Liste gelöscht. Die gesetzten Binär–Variablen an den Baustein–Eingängen sind davon *nicht* betroffen.
- Der Zeiger auf die interne Liste wird *nicht* verändert d.h. ein nachfolgender Suchvorgang beginnt ab der Stelle, auf die der Zeiger der internen Liste vor dem Löschen gezeigt hat.
- Der Ausgang END wird zu 1 gesetzt, der Ausgang NR zu 0.

FSU BINÄR

- Ein 1–Signal gibt den Suchlauf nach gesetzten Binär–Variablen in der internen Liste frei.
- Ein 0–Signal sperrt den Suchlauf; dabei werden an den Ausgängen deren Altwerte bei jedem Zyklus neu zugewiesen.
- Wenn eine Binär–Variable mit dem Wert 1 in der internen Liste gefunden wurde, so wird die Nummer dieser gesetzten Binär–Variablen am Ausgang NR ausgegeben. Die Binär–Variable wird dann in der *internen* Liste gelöscht. Der Ausgang END (Listenende erreicht) wird zu 0 gesetzt, falls es sich nicht um die letzte Binär–Variable der Liste handelt.
- Die Numerierung der Binär–Variablen beginnt mit 1.
- Solange der Baustein beim Suchlauf nicht das Ende der internen Liste erreicht hat, behält der Ausgang END den Wert 0 bei.
- Bei jedem erneuten Aufruf setzt der Baustein den Suchlauf in der internen Liste fort und zwar mit der nächsten Binär–Variablen nach der zuletzt gefundenen Binär–Variablen. *Zuvor* wird die zuletzt gefundene Binär–Variable in der internen Liste gelöscht, um eine Doppelmeldung zu vermeiden.
- Wird beim Suchvorgang das Ende der internen Liste erreicht, so gilt:
 - der Ausgang END (Listenende erreicht) wird zu 1 gesetzt.
 - am Ausgang NR wird die Nummer der zuletzt gefundenen Binär–Variablen der internen Liste ausgegeben.
 - bei jedem weiteren Aufruf des Bausteins wird die interne Liste ab der Stelle nach der zuletzt gefundenen Binär–Variablen durchsucht. Die Suche endet beim Auffinden einer gesetzten Binär–Variablen bzw. beim Erreichen vom Listenende.
- Ist die letzte Binär–Variable der Liste gesetzt und beim Suchlauf gefunden, so verharrt der Suchvorgang dort bis mit dem Eingang S ein neuer Suchlauf vom Anfang der internen Liste vorbereitet wird.

- Falls bei einem Suchlauf vom Anfang der Liste keine gesetzte Binär–Variable gefunden wird, wird der Ausgang END (Listenende erreicht) zu 1 gesetzt und der Ausgang NR (Nummer) behält den Wert 0 bei. Bei jedem erneuten Aufruf des Bausteins wird immer die ganze Liste durchsucht bis eine Binär–Variable den Wert 1 annimmt und gefunden wird.

FKOP BINÄR

Mit einem 1–Signal am Eingang FKOP werden alle gesetzten Binär–Variablen der Eingangsliste zusätzlich in die interne Liste eingetragen. Dabei bleiben die schon gesetzten Binär–Variablen in der internen Liste erhalten. Die Aktualisierung erfolgt durch eine "Veroderung" der Eingangsliste in die interne Liste.

Aufsetzen beim Suchvorgang nach dem Aktualisieren

Die Aktualisierung der internen Liste hat keinen Einfluß auf den nächsten Suchlauf. Der Suchlauf beginnt genau an der Stelle in der internen Liste, an der er auch beginnen würde, wenn keine Aktualisierung erfolgt wäre.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Zahl der Binär–Variablen angegeben, die an den Eingängen $B_0 \dots B_{n-1}$ projiziert sind. Die Angabe erfolgt als direkte Konstante.

Achtung: Die Anzahl am Eingang #n *muß* ein ganzzahliges Vielfaches von 16 sein. Um diese Vorschrift einzuhalten, können auch Dummy–Operanden projiziert werden (z. B. an allen nicht benötigten Eingängen wird K0,0 angegeben).

B₀...B_{n-1} BINÄR

Der Eingang B₀ ist doppelbar (B₀...B_{n-1}). An den Eingängen B₀ ... B_{n-1} werden die zu untersuchenden Binär–Variablen vorgegeben. Die Anzahl der Variablen muß immer ein ganzzahliges Vielfaches von 16 sein. Nicht benötigte Eingänge B_i werden dazu am besten mit K0,0 = 0 belegt.

END BINÄR

Am Ausgang END wird signalisiert, ob beim Suchlauf das Ende der Liste erreicht wurde.

END = 0 → Listenende nicht erreicht

END = 1 → Listenende ist erreicht

Falls die letzte Variable der Liste gesetzt ist *und* beim gerade durchgeführten Suchvorgang gefunden wurde, wird ihre Nummer am Ausgang NR ausgegeben und zusätzlich dem Ausgang END der Wert 1 zugewiesen.

NR WORT

Am Ausgang NR wird die Listen-Nummer der zuletzt gefundenen Variablen ausgegeben.
Es gilt die Zuordnung:

Variable am Eingang	Listennummer
B0	1
B1	2
.	.
.	.
.	.
B _{n-1}	n

Wird beim Suchvorgang das Ende der Liste erreicht, ohne daß eine neue Variable gefunden wurde, so wird am Ausgang NR weiterhin die Nummer der zuletzt gefundenen Variablen ausgegeben.
Am Ausgang NR wird der Wert 0 ausgegeben, wenn in der Liste *keine* Variable gesetzt ist.

Prioritäten der Eingänge S, R, FSU, FKOP

- Die höchste Priorität hat der Set-Eingang S. Solange der Eingang S ein 1-Signal hat, wird kein anderer Eingang bearbeitet.
- Die zweithöchste Priorität hat der Reset-Eingang R. Solange der Eingang R ein 1-Signal hat, werden die nachfolgenden Eingänge nicht bearbeitet.
- Der Aktualisierungs-Eingang FKOP hat Priorität vor dem Freigabe-Suchlauf-Eingang FSU.

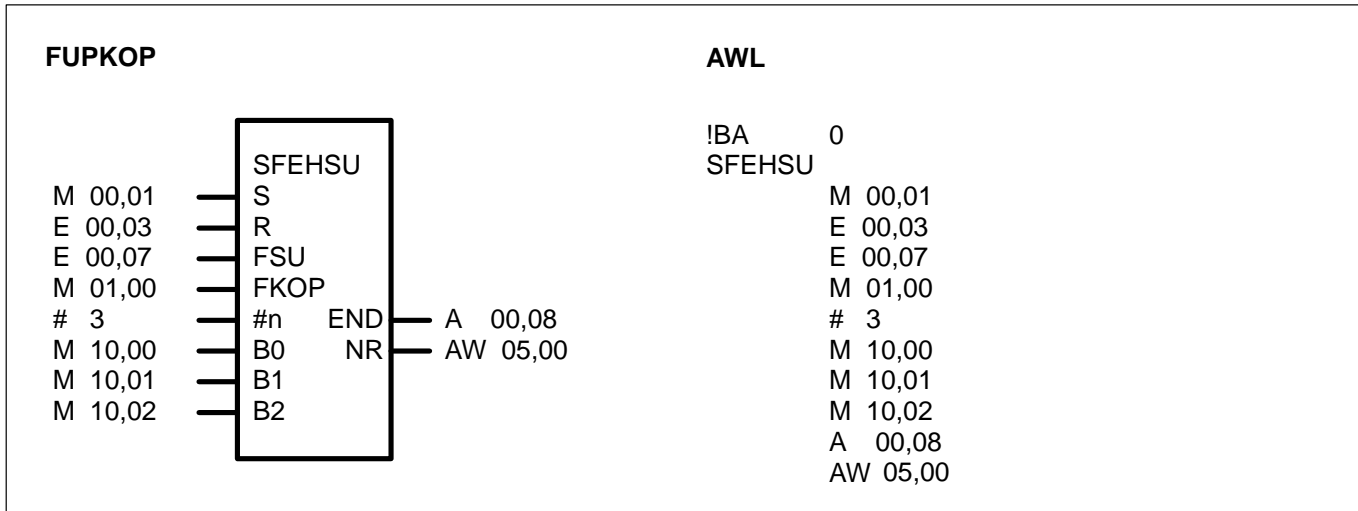
Aktualisierung der internen Liste (Eingang FKOP):

Fall 1: Aktualisierung freigegeben (FKOP = 1)
Ist die Aktualisierung freigegeben, so werden alle gesetzten Binär-Eingangsvariablen zusätzlich in die interne Liste eingetragen. Falls der Suchvorgang am Eingang FSU freigegeben ist, wird nach der Aktualisierung der internen Liste der Suchvorgang sofort durchgeführt.

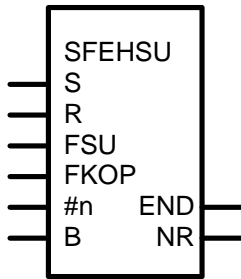
Fall 2: Aktualisierung nicht freigegeben (FKOP = 0)
Ist die Aktualisierung nicht freigegeben, so wird, falls der Suchvorgang am Eingang FSU freigegeben ist, der Suchvorgang sofort durchgeführt.

Ist in beiden Fällen der Suchlauf nicht freigegeben, so werden an den Ausgängen END und NR die Altwerte ausgegeben.

Beispiel



VE-FUP-Definition

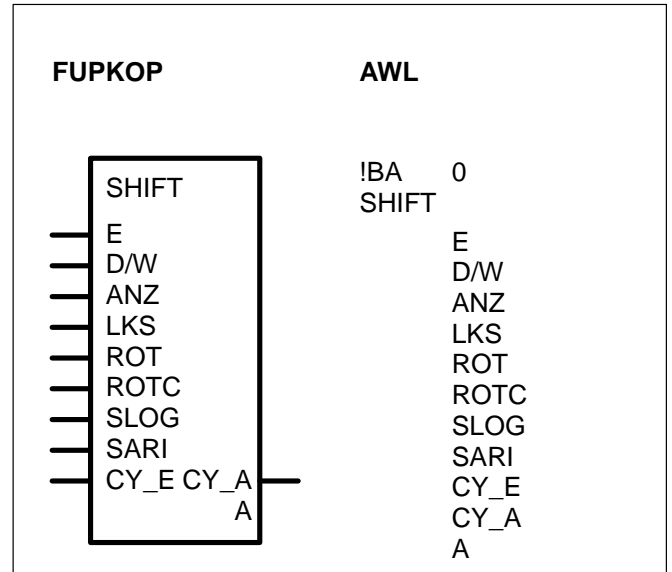


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
S	E	L	N	P	J	0	0
R	E	L	N	P	J	0	0
FSU	E	L	N	P	J	0	0
FKOP	E	L	N	P	J	0	0
#n	K	W	N	P	J	0	0
B0	E	L	N	P	J	1	0
END	A	L	N	P	J	0	0
NR	A	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	SFEHSU			
00002		PP 0	S	Eingang BINÄR (Set)
00003		PP 0	R	Eingang BINÄR (Reset)
00004		PP 0	FSU	Eingang BINÄR (Freigabe suchen)
00005		PP 0	FKOP	Eingang BINÄR (Freigabe kopiere)
00006		PP 0	#n	# KONSTANTE(Anzahl Bit)
	[1		
00007		PP 0	B	Eingang BINÄR (Element Bitfeld)
]	1		
00008		PP 0	END	Ausgang BINÄR (Fehlende erreicht)
00009		PP 0	NR	Ausgang WORT (Nummer)

Der Funktionsbaustein shiftet den am Eingang anliegenden Operanden um eine vorgebbare Anzahl von Bitpositionen nach rechts oder links.



Parameter

E	WORT DOPPELWORT	EW, AW, MW, KW KD, MD	Zu "shiftender" Operand, Wort oder Doppelwort.
D/W	BINÄR	E, A, M, K, S	Formatauswahl, Doppelwort oder Wort D/W = 0 → Wort D/W = 1 → Doppelwort
ANZ	WORT	EW, AW, MW, KW	Anzahl der zu "shiftenden" Bitpositionen
LKS	BINÄR	E, A, M, K, S	Shiftrichtung, links oder rechts LKS = 0 → rechts shiften LKS = 1 → links shiften
ROT	BINÄR	E, A, M, K, S	Shift-Art: ROTIEREN
ROTC	BINÄR	E, A, M, K, S	Shift-Art: ROTIEREN durch das CARRY-FLAG
SLOG	BINÄR	E, A, M, K, S	SHIFT-Art: SHIFTEN LOGISCH
SARI	BINÄR	E, A, M, K, S	SHIFT-Art: SHIFTEN ARITHMETISCH
CY_E	BINÄR	E, A, M, K, S	Anfangswert für das CARRY- FLAG bei Shift-Art ROTC
CY_A	BINÄR	A, M	Zustand des CARRY_FLAG nach dem Shift-Vorgang
A	WORT DOPPELWORT	AW, MW MD	Ergebnis des Shift-Vorgangs

VE-Daten

Laufzeit:	
Grundlaufzeit:	37 ... 78 µs je nach Shiftart
Zusätzliche Laufzeit:	1 µs pro zu "shiftender" Bitposition
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein shiftet den am Eingang anliegenden Operanden um eine vorgebbare Anzahl von Bitpositionen nach rechts oder links.

Das Ergebnis des Shift-Vorgangs und das CARRY-Flag stehen an den Bausteinausgängen zur Verfügung. Der Operand am Eingang bleibt unverändert.

Die gewünschte Shift-Art wird projiziert an den Eingängen:

- ROT
- ROTC
- SLOG
- SARITH

Werden gleichzeitig mehrere Shift-Arten vorgegeben, so gilt die Shift-Art, die in der Reihenfolge der Bausteineingänge am weitesten vorn steht.

Der Baustein kann sowohl Wort-Operanden als auch Doppelwort-Operanden shiften.

Achtung: Am Ausgang dieses Bausteins kann u.U. der für arithmetische Operationen verbotene Wert 8000H bzw. 8000 0000 H entstehen.

Die Ein- und Ausgänge sind weder doppelbar noch invertierbar noch negierbar.

E WORT/DOPPELWORT

Zu "shiftender" Operand

Auf den Eingangsoperand wird die projizierte SHIFT-Operation angewendet. Der Eingangsoperand wird dabei nicht verändert.

D/W BINÄR

Formatauswahl für den Eingangsoperand

D/W = 0 -> WORT

D/W = 1 -> DOPPELWORT

ANZ WORT

Anzahl der zu "shiftenden" Bitpositionen
Sinnvoller Bereich für die Anzahl n:

Wort-Operanden: $0 \leq n \leq 16$

Doppelwort-Operanden: $0 \leq n \leq 32$

LKS BINÄR

Richtung, in die "geschiftet" wird

LKS = 0 -> nach rechts shiften

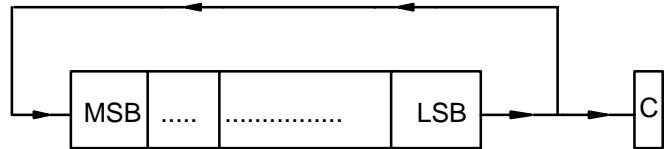
LKS = 1 -> nach links shiften

ROT BINÄR

Shift-Art: ROTIEREN

Die durch den Shift-Vorgang freiwerdende Bitposition wird durch das herausgeschobene Bit ersetzt. Zusätzlich wird der Inhalt des CARRY-FLAGS durch das herausgeschobene Bit ersetzt. Nach dem Shift-Vorgang steht das Ergebnis und der Inhalt des CARRY-FLAGS an den Bausteinausgängen zur Verfügung.

ROT nach rechts:



ROT nach Links:

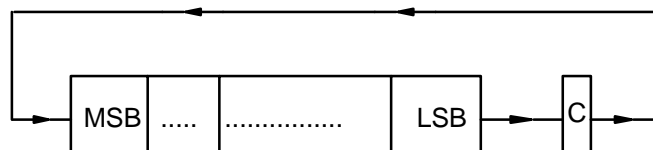


LSB: Niederwertigstes Bit (least significant bit)
MSB: Höchstwertigstes Bit (most significant bit)

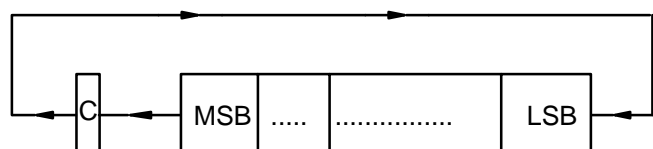
ROTC BINÄR

Shift-Art: ROTIEREN durch das CARRY-FLAG. Die durch den Shift-Vorgang freiwerdende Bitposition wird durch den Inhalt des CARRY-FLAGS ersetzt. Anschließend wird das CARRY-FLAG durch das herausgeschobene Bit ersetzt. Nach dem Shift-Vorgang steht das Ergebnis und der Inhalt des CARRY-FLAGS an den Bausteinausgängen zur Verfügung.

ROTC nach rechts:



ROTC nach links:



LSB: Niederwertigstes Bit (least significant bit)
MSB: Höchstwertigstes Bit (most significant bit)

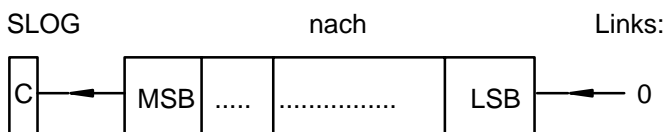
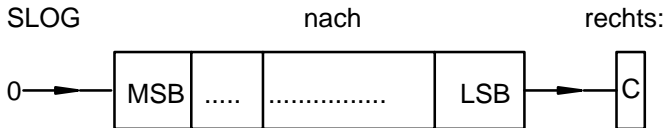
SLOG BINÄR

Shift-Art: LOGISCH SHIFTEN

Die durch den Shift-Vorgang freiwerdende Bitposition wird durch den Wert 0 ersetzt.

Der Inhalt des CARRY-FLAGS wird durch das herausgeschobene Bit ersetzt.

Nach dem Shift-Vorgang stehen das Ergebnis und der Inhalt des CARRY-FLAGS an den Bausteinausgängen zur Verfügung.



LSB: Niederwertigstes Bit (least significant bit)
MSB: Höchstwertigstes Bit (most significant bit)

SARI BINÄR

Shift-Art: ARITHMETISCH SHIFTEN

SHIFTEN ARITHMETISCH nach rechts:

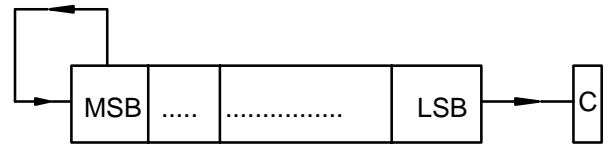
Die durch den Shift-Vorgang freiwerdende Bitposition 15 (MSB, Vorzeichenbit) wird durch sich selbst ersetzt. Der Inhalt des CARRY-FLAGS wird durch das herausgeschobene Bit ersetzt. Nach dem Shift-Vorgang steht das Ergebnis und der Inhalt des CARRY-FLAGS an den Bausteinausgängen zur Verfügung.

SHIFTEN ARITHMETISCH nach links: (Identisch mit SLOG links)

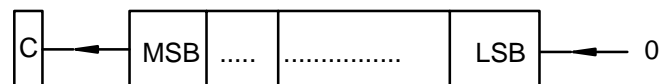
Die durch den Shift-Vorgang freiwerdende Bitposition 0 wird durch den Wert 0 ersetzt.

Der Inhalt des CARRY-FLAGS wird durch das herausgeschobene Bit ersetzt. Nach dem Shift-Vorgang stehen das Ergebnis und der Inhalt des CARRY-FLAGS an den Bausteinausgängen zur Verfügung.

SARI nach rechts:



SARI nach Links:(Identisch SLOG nach links)



LSB: Niederwertigstes Bit (least significant bit)
MSB: Höchstwertigstes Bit (most significant bit)

CY_E BINÄR

Anfangswert für das CARRY-FLAG bei Shift-Art ROTC. Für die Shift-Art "ROTIEREN durch das CARRY-FLAG" wird für das CARRY-FLAG ein Anfangswert benötigt. Dieser Anfangswert wird am Eingang CY_E vorgegeben.

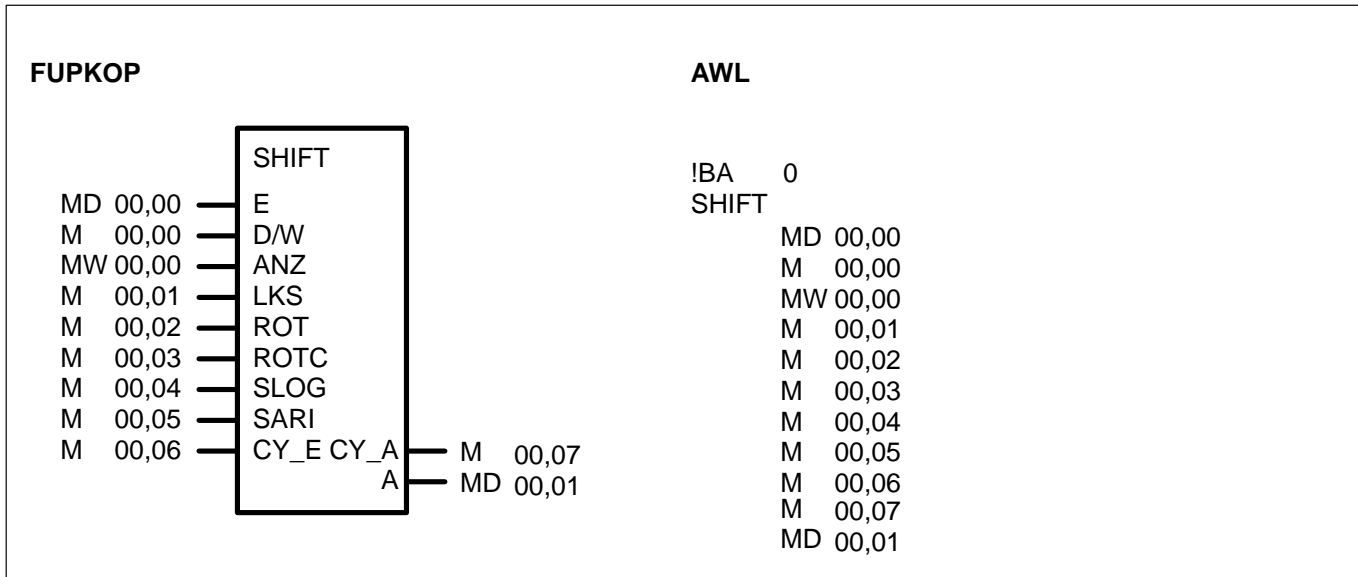
CY_A BINÄR

Zustand des CARRY-FLAGS nach dem Shift-Vorgang. Nach dem Shift-Vorgang steht an diesem Ausgang der aktuelle Wert des CARRY-FLAGS zur Verfügung.

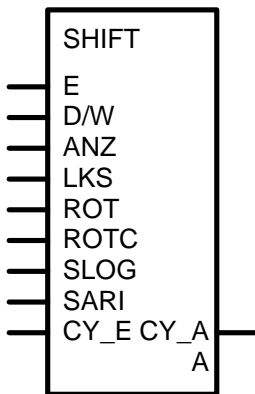
A WORT/DOPPELWORT

Ergebnis des Shift-Vorgangs
Nach dem Shift-Vorgang steht an diesem Ausgang das Ergebnis zur Verfügung.

Beispiel



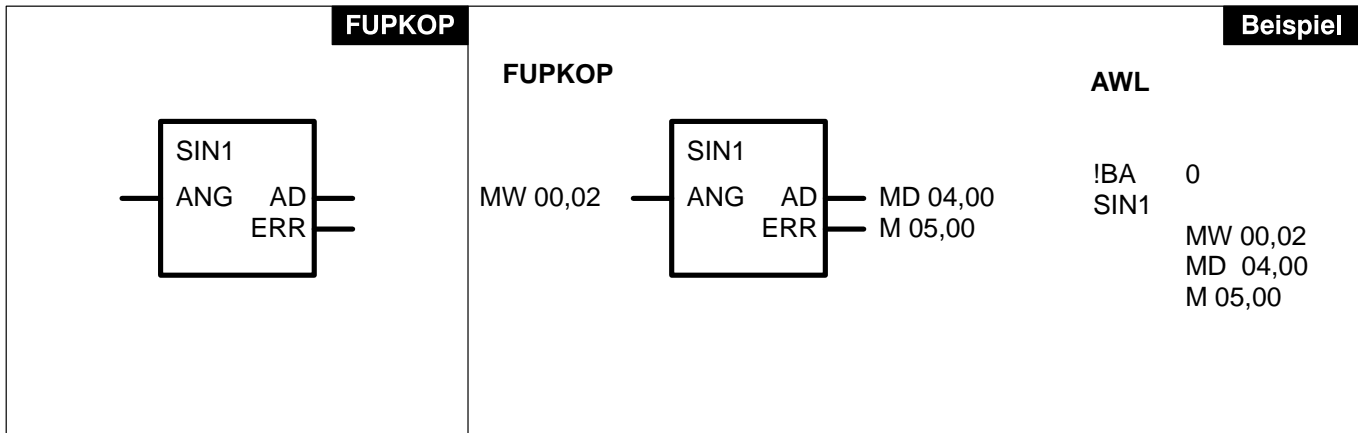
VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E	E	X	N	P	J	0	0
D/W	E	L	N	P	J	0	0
ANZ	E	W	N	P	J	0	0
LKS	E	L	N	P	J	0	0
ROT	E	L	N	P	J	0	0
ROTC	E	L	N	P	J	0	0
SLOG	E	L	N	P	J	0	0
SARI	E	L	N	P	J	0	0
CY_E	E	L	N	P	J	0	0
CY_A	A	L	N	P	J	0	0
A	A	X	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr (vorbelegt 0)
00001	SHIFT			
00002		PP 0	E	zu "shiftender" Operand (WORT/DOPPELWORT)
00003		PP 0	D/W	Formatauswahl (BINÄR)
00004		PP 0	ANZ	Anzahl Bitpositionen (WORT)
00005		PP 0	LKS	Shift-Richtung (BINÄR)
00006		PP 0	ROT	Rotieren (BINÄR)
00007		PP 0	ROTC	Rotieren d. Carry (BINÄR)
00008		PP 0	SLOG	Shift logisch (BINÄR)
00009		PP 0	SARI	Shift arithmetisch (BINÄR)
00010		PP 0	CY_E	Carry-Eingang (BINÄR)
00011		PP 0	CY_A	Carry-Ausgang (BINÄR)
00012		PP 0	A	Ergebnis Shiften (WORT/DOPPELWORT)



Parameter

ANG WORT EW, AW, MW
 AD DOPPELWORT MD
 ERR BINÄR M, A

Winkel 0 ... 3600 (entspricht 0,0° ... 360,0°)
 Sinus des Eingangswertes
 Fehler falls Eingangswert negativ oder größer 3600

Beschreibung

Der Funktionsbaustein berechnet den Sinuswert vom Eingang ANG und weist diesen dem Ausgang AD zu. Das Ergebnis liegt im Bereich -100 000 ... 100 000. Ist der Wert am Eingang ANG negativ oder größer 3600 (360°), wird dem Ausgang AD der Wert 0 und dem Ausgang ERR der Wert 1 zugewiesen. Der max. Fehler des Ergebnisses ist ± 0,5.

ERR BINÄR

Der Ausgang ERR zeigt an, ob der Eingangswert im korrekten Bereich liegt ($0 \leq ANG \leq 3600$).

Eingang $0 \leq ANG \leq 3600$
 -> ERR = 0 und AD = SIN(ANG)
 Eingang $ANG < 0$ oder $ANG > 3600$
 -> ERR = 1 und AD = 0

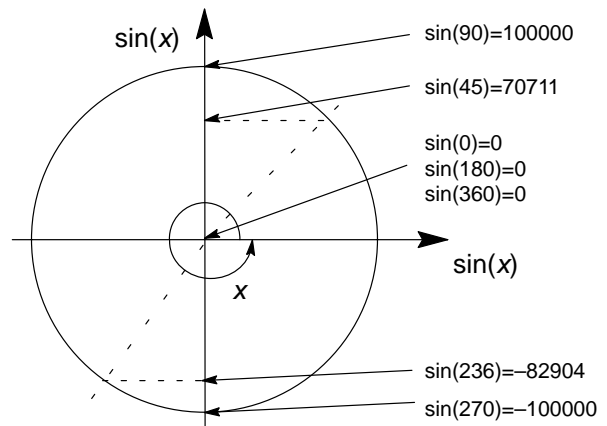
ANG WORT

Es wird der Sinus des Wertes des Eingangsoperanden ANG berechnet. Das Ergebnis ist als Wert des Ausgangsoperanden AD verfügbar.

Eingabe:

0000 für 0 Grad
 0001 für 0,1 Grad
 0010 für 1,0 Grad
 ..
 3600 für 360,0 Grad

Beispiele von Sinuswerten



AD DOPPELWORT

Der Sinus des Eingangswertes ist am Ausgang AD verfügbar.

VE-Daten

Laufzeit:

Grundlaufzeit:

Zusätzliche Laufzeit:

Aktualisierung der Ausgänge:

Anzahl der Vergangenheitswerte:

Verfügbar ab:

07 KR 91 / 07 KT 92 / 07 KT 93 / 07 KP 62

47 – 69 µs

—

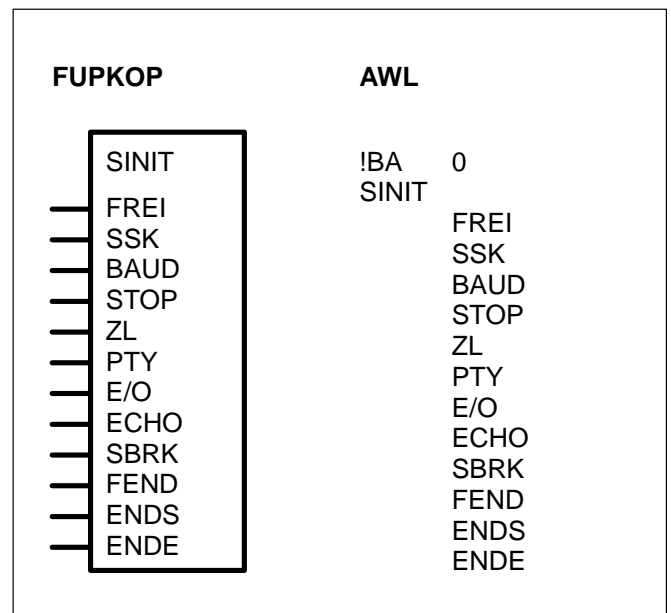
ja

0 Worte

ABB Procontic CS31 07KT92 R202/R262, 07KT93 R171

ABB Procontic T200 07KP62 R202

Der Funktionsbaustein SINIT wird bei jeder 0→1-Flanke am Eingang FREI *einmal* bearbeitet. Er initialisiert dabei die am Eingang SSK angegebene serielle Schnittstelle (COM1, COM2).



Parameter

FREI	BINÄR	A, E, S, M, K	Freigabe der Bausteinbearbeitung, 0→1-Flanke
SSK	WORT	AW, EW, MW, KW	Schnittstellenkennung (1 oder 2)
BAUD	WORT	AW, EW, MW, KW	Baudrate: 300 ... 9600 Baud
STOP	WORT	AW, EW, MW, KW	Anzahl Stoppbits; Eingang ist ohne Wirkung
ZL	WORT	AW, EW, MW, KW	Zeichenlänge, 7 oder 8 Datenbits pro Zeichen
PTY	BINÄR	A, E, S, M, K	Parity, enable/disable
E/O	BINÄR	A, E, S, M, K	Parity even/odd
ECHO	BINÄR	A, E, S, M, K	Echo, on/off
SBRK	BINÄR	A, E, S, M, K	Send Break-character
FEND	BINÄR	A, E, S, M, K	Freigabe Textabschlußzeichen für Senderichtung
ENDS	WORT	AW, EW, MW, KW	Textabschlußzeichen für Senderichtung
ENDE	WORT	AW, EW, MW, KW	Textabschlußzeichen für Empfangsrichtung

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	25 µs	865 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	entfällt	entfällt
Anzahl der Vergangenheitswerte:	1 Wort	1 Wort
Verfügbar ab:	ABB Procontic CS31 / Kommunikationsmodul 07 KP 62	07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein SINIT wird bei jeder 0→1-Flanke am Eingang FREI *einmal* bearbeitet. Er initialisiert dabei die am Eingang SSK angegebene serielle Schnittstelle (COM1, COM2).

Die Eingänge sind weder doppelbar noch negierbar/invertierbar.

Dem Anwender steht die serielle Schnittstelle COM1 zur Verfügung, bei 07 KT 92 und 07 KP 62 zusätzlich noch COM2. Diese Schnittstellen können vom SPS-Programm bedient werden (z. B. mit den Bausteinen DRUCK und EMAS). Beachten Sie dazu bitte auch das Kapitel "Serielle Schnittstellen" in der SPS-Beschreibung. Vor der Benutzung einer dieser Schnittstellen muß diese initialisiert werden. Dazu steht der Funktionsbaustein SINIT zur Verfügung.

Serielle Schnittstellen der SPS

Folgende serielle Schnittstellen können mit dem Baustein SINIT initialisiert werden:

Zentraleinheit 07 KR 31:	COM1
Zentraleinheit 07 KR 91:	COM1
Zentraleinheit 07 KT 92:	COM1, COM2
Kommunikationsprozessor 07 KP 62:	COM1, COM2

Die Synchronisation zwischen den seriellen Schnittstellen und dem angeschlossenen Gerät erfolgt grundsätzlich mit den Signalen RTS und CTS.

FREI BINÄR

Bei Vorgabe einer 0→1-Flanke am Eingang FREI wird der Baustein *einmal* durchlaufen. Dadurch wird die serielle Schnittstelle, deren Nummer am Eingang SSK angegeben ist, initialisiert und die Schnittstelle ist danach betriebsbereit.

SSK WORT

Am Eingang SSK wird die Nummer der zu initialisierenden Schnittstelle vorgegeben.

Es gilt:

COM1:	Nummer = 1
COM2:	Nummer = 2

BAUD WORT

Am Eingang BAUD wird der Wert für die Baudrate angegeben.

Baudrate: 300 ... 9600 Baud

STOP WORT

Die Anzahl der Stoppbits ist fest auf 1 eingestellt und nicht änderbar. Der am Eingang STOP angegebene Wert für die Anzahl der Stoppbits hat keine Bedeutung.

ZL WORT

Am Eingang ZL wird die gewünschte Zeichenlänge angegeben. Zeichenlänge bedeutet die Anzahl der Daten-Bits pro Zeichen.

Möglich sind 7 oder 8 Daten-Bits pro Zeichen.

PTY BINÄR

Am Eingang PTY wird angegeben, ob ein Zeichen mit oder ohne Paritätsbit übertragen wird.

PTY = 0 → Übertragung ohne Paritätsbit

PTY = 1 → Übertragung mit Paritätsbit

E/O BINÄR

Am Eingang E/O wird angegeben, ob ein gerades (even) oder ungerades (odd) Paritätsbit gewünscht wird.

E/O = 0 → Paritätsbit ungerade (odd)

E/O = 1 → Paritätsbit gerade (even)

ECHO BINÄR

Am Eingang ECHO wird angegeben, ob die über die betreffende Schnittstelle empfangenen Zeichen von der SPS reflektiert (geecho) werden sollen. Damit kann z. B. der Sender eines Zeichens feststellen, ob dieses korrekt in der SPS angekommen ist.

ECHO = 0 → kein Echo, Zeichen wird nicht reflektiert

ECHO = 1 → Echo, Zeichen wird reflektiert

SBRK BINÄR

Am Eingang SBRK (Send-Break-Character) kann der Zustand der Sendeleitung TxD beeinflusst werden.

SBRK = 0 → Normalzustand der Sendeleitung TxD zur Übertragung von Zeichen

SBRK = 1 → Sendeleitung TxD wird auf "0" gesetzt

FEND BINÄR

Am Eingang FEND (Freigabe Endezeichen) wird angegeben, ob das am Eingang ENDS projizierte Text-Abschlußzeichen für die Senderichtung mit ausgegeben wird oder nicht.

FEND = 0 → Text-Abschlußzeichen in Senderichtung wird nicht ausgegeben

FEND = 1 → Text-Abschlußzeichen in Senderichtung wird ausgegeben

ENDS WORT

Am Eingang ENDS kann ein frei wählbares Text-Abschlußzeichen für die Senderichtung angegeben werden. Dieses Abschlußzeichen wird dann automatisch an jeden Text (Telegramm) angehängt, den der Baustein DRUCK über die serielle Schnittstelle nach außen sendet. Voraussetzung ist allerdings die Freigabe am Eingang FEND.

Die Angabe des Textabschlußzeichens erfolgt als Zahlenwert.

Bsp.:

3 bzw. 03_H bedeutet <ETX>

4 bzw. 04_H bedeutet <EOT>

13 bzw. 0D_H bedeutet <CR>

10 bzw. 0A_H bedeutet <LF>

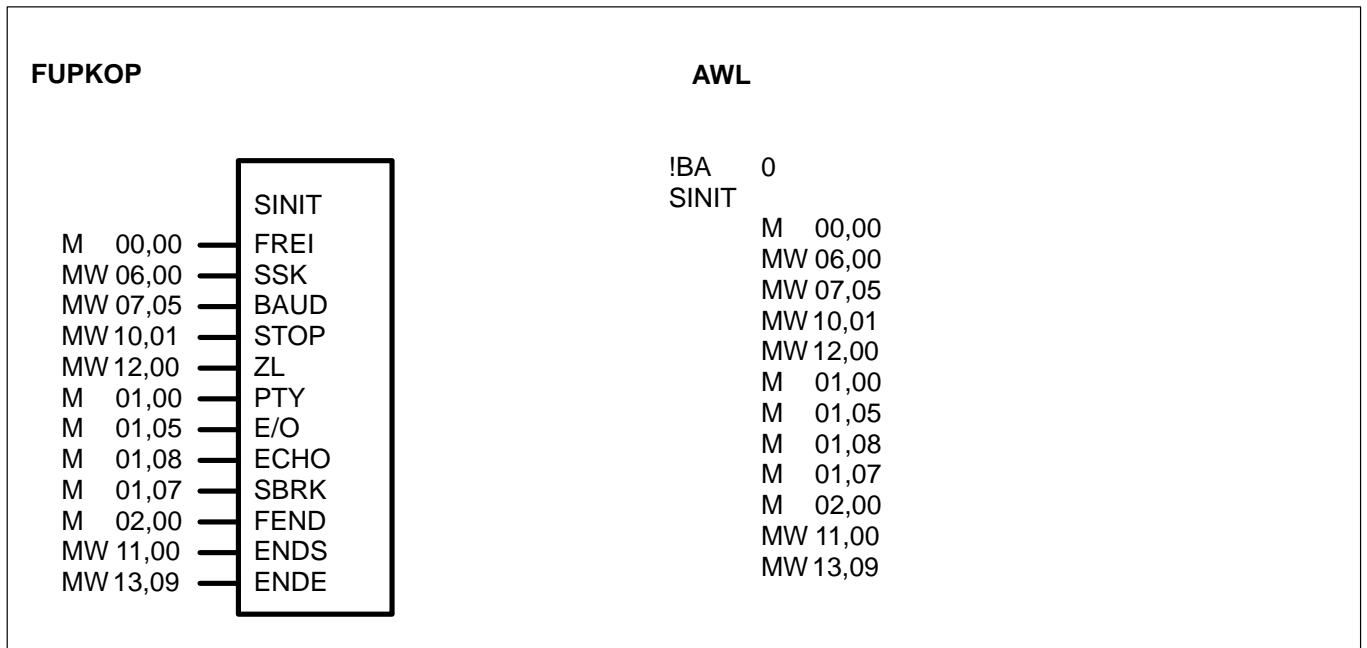
32 bzw. 20_H bedeutet <SP>

.
.
.

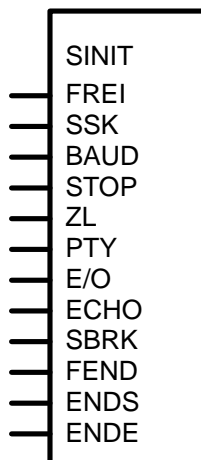
ENDE WORT

Am Eingang ENDE kann ein frei wählbares Text-Abschlußzeichen für die Empfangsrichtung angegeben werden. Beim Empfang eines Telegramms über die serielle Schnittstelle erkennt die SPS an diesem Abschlußzeichen das Ende des Telegramms. Die Angabe des Abschlußzeichens erfolgt in der gleichen Art und Weise wie beim Eingang ENDS.

Beispiel



VE-FUP-Definition

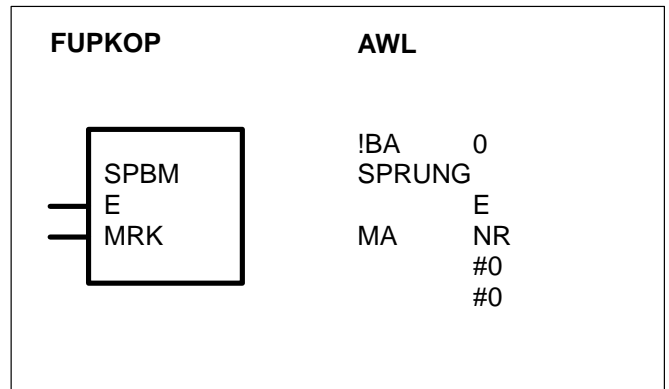


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
SSK	E	W	N	P	J	0	0
BAUD	E	W	N	P	J	0	0
STOP	E	W	N	P	J	0	0
ZL	E	W	N	P	J	0	0
PTY	E	L	N	P	J	0	0
E/O	E	L	N	P	J	0	0
ECHO	E	L	N	P	J	0	0
SBRK	E	L	N	P	J	0	0
FEND	E	L	N	P	J	0	0
ENDS	E	W	N	P	J	0	0
ENDE	E	W	N	P	J	0	0

VE-AWL-Definition

Nr	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	SINIT			
00002		PP 0	FREI	Eingang BINÄR (Baustein-Freigabe)
00003		PP 0	SSK	Eingang WORT (Schnittstellen)
00004		PP 0	BAUD	Eingang WORT (Baudrate)
00005		PP 0	STOP	Eingang WORT (Anzahl Stoppbit)
0006		PP 0	ZL	Eingang WORT (Zeichenlänge)
0007		PP 0	PTY	Eingang BINÄR (Parity)
0008		PP 0	E/O	Eingang BINÄR (parity even/odd)
0009		PP 0	ECHO	Eingang BINÄR (Echo)
00010		PP 0	SBRK	Eingang BINÄR (Send Break)
00011		PP 0	FEND	Eingang BINÄR (ENDS ausgeben)
00012		PP 0	ENDS	Eingang WORT (Textabschlußzeichen senden)
00013		PP 0	ENDE	Eingang WORT (Textabschlußzeichen empfangen)

Der Funktionsbaustein bewirkt einen Sprung zu der am Eingang MRK angegebenen Zielmarke, falls die Sprungbedingung am Eingang E erfüllt ist.



Parameter

E	BINÄR	E, M, A, K, S	Sprungbedingung
MRK	SPEZIAL	MRK	Zielmarke

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	26 µs	9 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	entfällt	entfällt
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein bewirkt einen Sprung zu der am Eingang MRK angegebenen Zielmarke, falls die Sprungbedingung am Eingang E erfüllt ist.

Zur Vermeidung von Endlosschleifen durch fehlerhafte Projektierung werden von der SPS nur VORWÄRTS-SPRÜNGE akzeptiert.

Die Bezeichnung der Zielmarke besteht aus einem Schlüsselwort und einer Nummer.

FUPKOP

Bezeichnung der Zielmarke: MRK Nr
 Dabei ist: MRK Schlüsselwort
 0 ≤ Nr ≤ 999 Nummer der Marke

AWL

Bezeichnung der Zielmarke: MA Nr
 Dabei ist: MA Schlüsselwort
 0 ≤ Nr ≤ 999 Nummer der Marke

Die im Baustein angegebene Marke wird auch an der Stelle im Anwenderprogramm angegeben, zu der der Sprung erfolgen soll. Auf ein und dieselbe Marke kann

von mehreren Sprungbausteinen aus gesprungen werden.

Eine Marke darf mehrmals im Programm vorkommen. Ist dies der Fall, so gehört immer der Sprungbaustein und die ihm als *nächstes* nachfolgende dieser namensgleichen Marken zusammen.

Die SPS errechnet aufgrund der Marken:

- Die Anzahl der zu überspringenden Vergangenheitswerte
- Die Adresse der Zielmarke

Die beiden errechneten Werte werden von der SPS an den dafür vorgesehenen Stellen des Sprung-Bausteins eingetragen. Bei der Projektierung des Sprung-Bausteins in Anweisungsliste können diese beiden Programmworte prinzipiell mit beliebigem Inhalt beschrieben werden. Es wird aber vorgeschlagen hier den Zahlenwert 0 einzutragen.

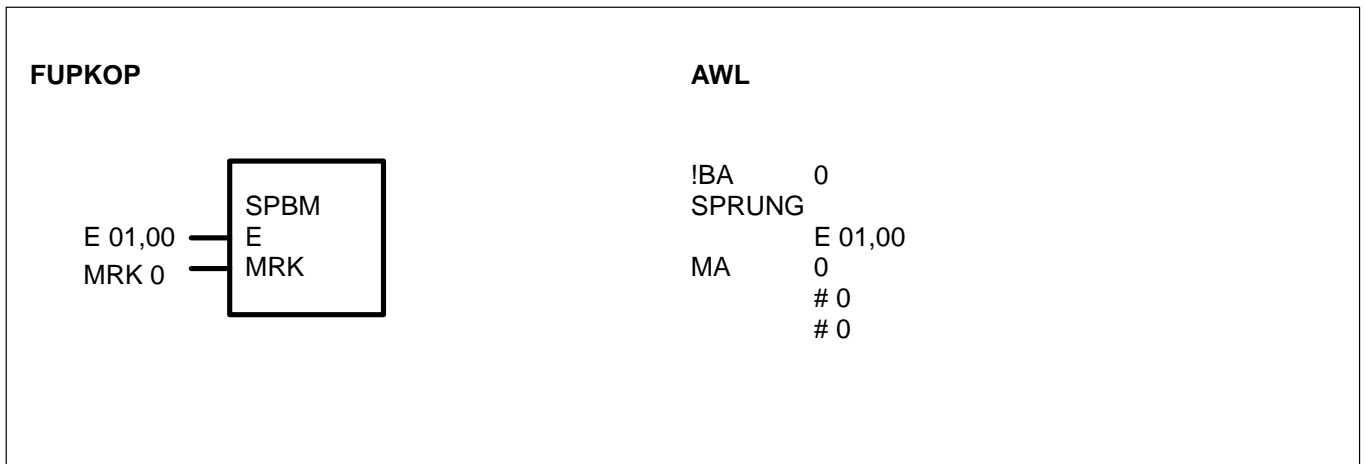
Vorsicht:

Die beiden Programmworte sollten aber nicht mit NOP beschrieben werden, da diese beim Optimieren des Anwenderprogramms entfernt werden.

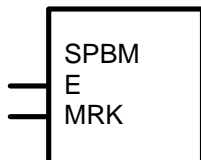
Beispiel für Sprung-Baustein in AWL:

!E0,0
&E0,1
/M0,0
=M0,1
!BA 123
SPRUNG Sprungbaustein
M0,1 Sprungbedingung
MA 99 Name der Zielmarke
#0 Freiplatz für Anzahl der zu über-
springenden Verg.werte (nicht benutzt
in 07 KR 31)
#0 Freiplatz für Zieladresse
!E1,5
&E4,9
=A6,8
=M9,7
.
.
.
MA 99 Zielmarke, Sprungziel für Sprung-
baustein
!E10,12
/E10,13
&E10,14
=A14,15
!PE

Beispiel



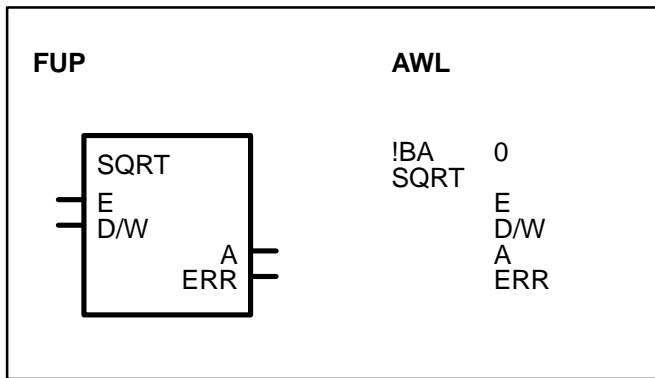
VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E	E	L	N	P	J	0	0
MRK	S	S	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	SPRUNG			
00002		PP 0	E	Sprungbedingung BINÄR
00003	MA	PP 0	MRK	Zielmarke
00004		# 0		
00005		# 0		



Parameter

E	WORT, DOPPELWORT	EW, AW, MW, KW MD, KD	Eingang
D/W	BINÄR	E, A, M, K, S	Format des Ein/Ausgangs
A	WORT, DOPPELWORT	EW, AW, MW, KW, MD, KD	Quadratwurzel des Eingangswerts
ERR	BINÄR	E, A, M, K	Fehler bei negativem Eingangswert

VE-DATEN

Laufzeit:

Grundlaufzeit:	55 µs ... 193 µs
Zusätzliche Laufzeit	keine
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	Keine
Verfügbar mit:	ABB Procontic T320 V8.0 / 35 ZE 94 R0101

Beschreibung

Der Funktionsbaustein SQRT bildet die Quadratwurzel des Wertes am Eingang E. Das Ergebnis steht am Ausgang A zur Verfügung. Das Ergebnis wird grundsätzlich auf eine ganze Zahl *abgerundet*. Der Wert am Eingang E muß eine positive Zahl sein. Ist der Wert am Eingang negativ, so wird am Ausgang A der Wert 'Null' und am Binärausgang ERR der Wert '1' ausgegeben.

E WORT/DOPPELWORT

Aus dem Wert des Eingangsoperanden E wird die Quadratwurzel gebildet und steht als Wert des Ausgangsoperanden A zur Verfügung.

D/W BINÄR

Der Eingang D/W legt das Format des Ein/Ausgangsoperanden fest.
 D/W = 0 → WORT
 D/W = 1 → DOPPELWORT

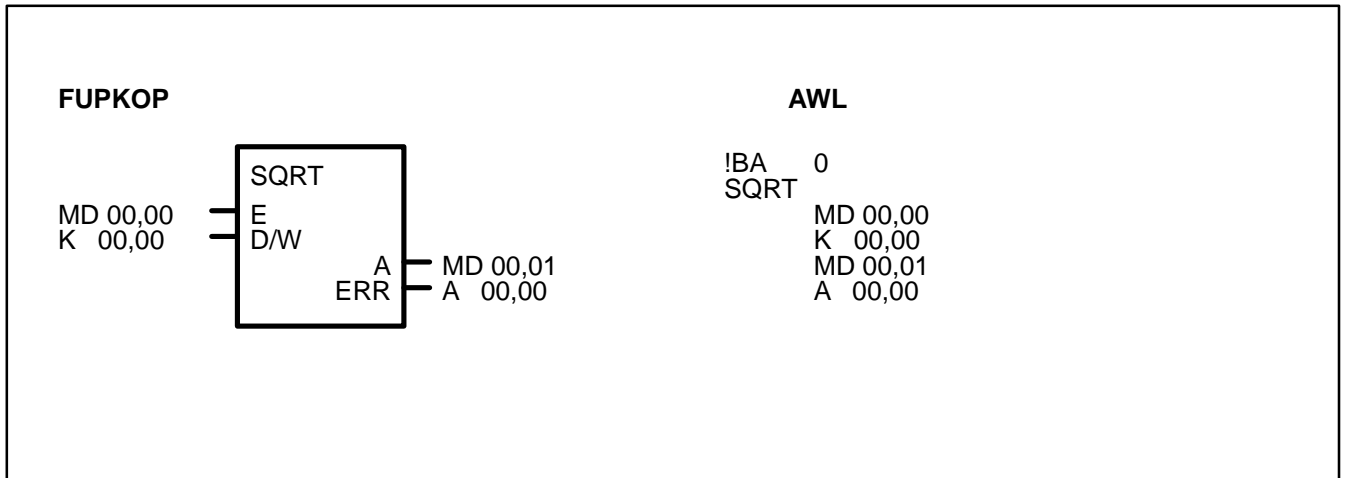
A WORT/DOPPELWORT

Am Ausgang A steht der Wert der Quadratwurzel zur Verfügung.

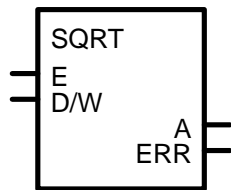
ERR BINÄR

Der Ausgang ERR zeigt an, ob der Wert des Eingangsoperanden E positiv (≥ 0) oder negativ (< 0) ist.
 Eingang $E \geq 0$ → ERR = 0 und A = Quadratwurzel
 Eingang $E < 0$ → ERR = 1 und A = 0

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E	E	X	N	P	J	0	0
D/W	E	L	N	P	J	0	0
A	A	X	N	P	J	0	0
ERR	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	SQRT			
00002		PP 0	E	Eingang (WORT,DOPPELWORT)
00003		PP 0	D/W	Format (BINÄR)
00004		PP 0	A	Ausgang (WORT,DOPPELWORT)
00005		PP 0	ERR	Fehler (BINÄR)

Dieses Verknüpfungselement realisiert einen Binär-Speicher mit der Eigenschaft "dominierend Rücksetzen".

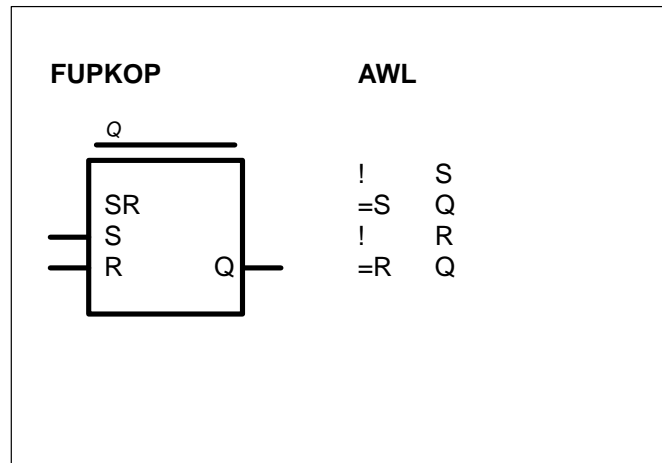
Der Operand Q, der die Speicherfunktion übernimmt, wird über dem Schaltzeichen eingetragen. Der Zustand des Operanden Q wird dem Operanden am Ausgang Q zugewiesen.

Zustand 1 am Eingang S setzt den Operanden Q auf Zustand 1.

Zustand 1 am Eingang R setzt den Operanden Q zurück auf Zustand 0.

Gleichzeitiger 1-Zustand am Eingang S und R setzt den Operanden Q zurück auf Zustand 0 (dominierend Rücksetzen).

Zustand 0 am Eingang S oder R hat keinen Einfluß auf den Operanden Q.



Parameter

S	BINÄR	E, M, A, S, K	Setz-Eingang
R	BINÄR	E, M, A, S, K	Rücksetz-Eingang
Q	BINÄR	M, A	Flip-Flop-Ausgang

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	8,6 µs	30 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Operand Q, der die Speicherfunktion übernimmt, wird über dem Schaltzeichen eingetragen. Der Zustand des Operanden Q wird dem Operanden am Ausgang Q zugewiesen.

Zustand 1 am Eingang S setzt den Operanden Q auf Zustand 1.

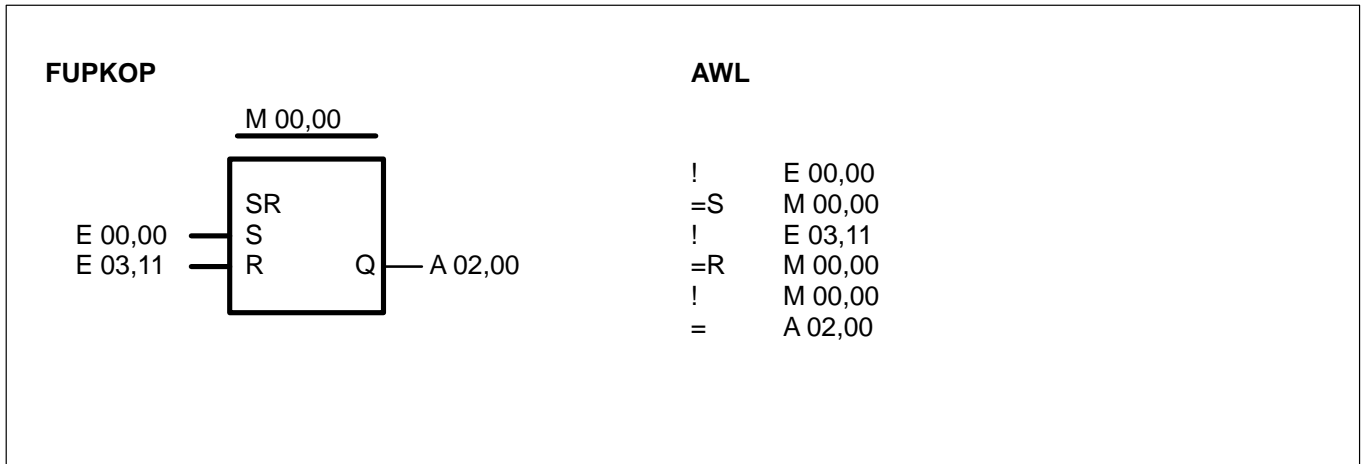
Zustand 1 am Eingang R setzt den Operanden Q zurück auf Zustand 0.

Gleichzeitiger 1-Zustand am Eingang S und R setzt den Operanden Q zurück auf Zustand 0 (dominierend Rücksetzen).

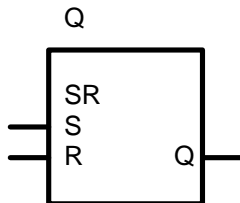
Zustand 0 am Eingang S oder R hat keinen Einfluß auf den Operanden Q.

Die Eingänge und der Ausgang sind nicht doppelbar. Die Eingänge R und S sind invertierbar.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
S	E	L	J	P	J	0	0
R	E	L	J	P	J	0	0
Q	A	L	N	P	J	0	0

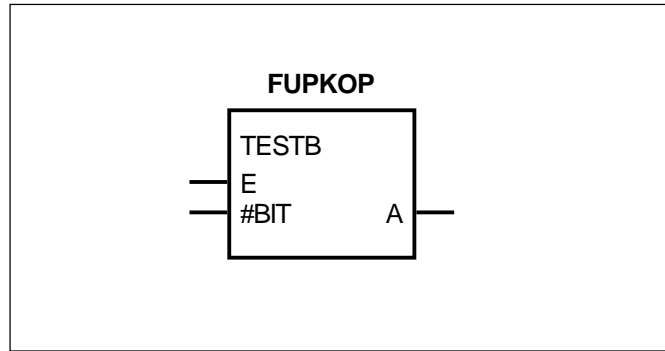
VE-AWL-Definition

00000	!	PP 0	S	Eingang BINÄR
00002	=S	PP 0	Q	Ausgang BINÄR
00004	!	PP 0	R	Eingang BINÄR
00006	=R	PP 0	Q	Ausgang BINÄR

Das Bit am Eingang #Bit (#) wird im Wort/Doppelwort E getestet und der Binärvariablen A zugewiesen. Zulässige Werte am Eingang # Bit: 0...31.

Wird am Eingang E eine Wortvariable eingegeben und #BIT ist größer als 15, so wird das entsprechende Bit der folgenden Wortvariablen getestet und zugewiesen.

Beispiel: E = MW 00,00 und #Bit = 16 → Bit 0 von MW 00,01 wird dem Ausgang A zugewiesen.



Parameter

E	WORT DOPPELWORT	EW, MW MD	Eingang
#BIT	DIREKTE KONSTANTE	#, #H	Bitposition; zulässige Werte: 0...31
A	BINÄR	A, M	Ausgang

VE-Daten

Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine

Verfügbar ab: 07 KR 91 R2 Index c, 07 KT 92 R2 Index c, 07 KT 93 R2 Index f, 07 KT 94

Beschreibung

E WORT/DOPPELWORT

Das Bit am Eingang #BIT (#) wird im Wort/Doppelwort E getestet und der Binärvariablen A zugewiesen.

Wird am Eingang E eine Wortvariable eingegeben und #BIT ist größer als 15, so wird das entsprechende Bit der folgenden Wortvariablen getestet und dem Ausgang A zugewiesen.

Beispiel: E = MW 00,02 und #BIT = 18 → Bit 2 von MW 00,03 wird dem Ausgang A zugewiesen.

#BIT Direkte Konstante (#,#H)

Am Eingang #BIT wird die Position des entsprechenden Bits angegeben.

Zulässige Werte: 0...31

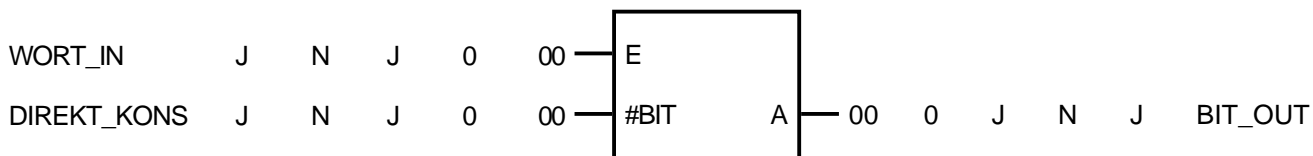
A BINÄR

Der Ausgang A wird in Abhängigkeit des entsprechenden Bits in E gesetzt (Bit = 1) bzw. zurückgesetzt (Bit = 0).

Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr		Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	--	----	-----	-----	-----	-----	--------------

TESTB



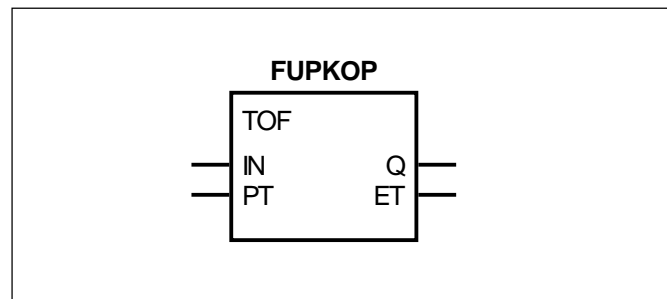
Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	TESTB		
	00001	PP 00	E	
	00002	PP 00	#BIT	
	00003	PP 00	A	
	00004			

Die 1-0-Flanke des Eingangs IN wird um die Zeitdauer PT verzögert und als 1-0-Flanke am Ausgang Q ausgegeben.

Der Ausgang ET zeigt die aktuelle Zeit an.

Geht der Eingang IN vor Ablauf der Zeit PT wieder auf 1-Pegel, so bleibt der Ausgang Q auf 1-Pegel.



Parameter

IN	BINÄR	E, M, A, S, K	Eingangssignal
PT	WORT/DOPPELWORT	MW&MW+1, MD, KW&KW+1, KD EW&EW+1, AW&AW+1	Verzögerungszeit
Q	BINÄR	M, A	Verzögertes Signal
ET	WORT/DOPPELWORT	MW&MW+1, MD	Zeitanzeige

Verfügbar ab: Advant Controller 31, Serie 40 und 50

Beschreibung

Die 1-0-Flanke des Eingangs IN wird um die Zeitdauer PT verzögert und als 1-0-Flanke am Ausgang Q ausgegeben.

Geht der Eingang IN vor Ablauf der Zeit PT wieder auf 1-Pegel, so bleibt der Ausgang Q auf 1-Pegel.

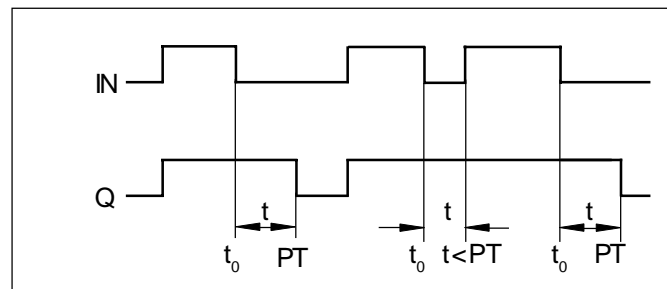
Die Istzeit wird am Ausgang ET angezeigt.

Für Zeiten, die kleiner als 65 s sind, können am Eingang PT bzw. Ausgang ET Variablen vom Typ WORT verwendet werden: Merker, Konstanten, andere WORT-Funktionen (+, -, AWT) und die Potentiometer-Eingänge der CPU.

In diesem Falle belegen die MW, KW, EW und AW zwei aufeinander folgende Adressen. MW+1, KW+1, EW+1 und AW+1 dürfen im Programm nicht mehr verwendet werden.

Die Zeitangabe erfolgt in Millisekunden.
Angebbarer Zeitbereich: 5 ms ... 24,8 Tage.

Bei der indirekten Konstanten KD wird der Wert der Impulslänge im Langtext angegeben.



Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit

Sinnvoller Bereich für PT: > 1 Zykluszeit

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar.

Allgemeines Verhalten

- Gestartete Zeitwerke werden vom Betriebssystem der SPS bearbeitet und sind deshalb vollkommen unabhängig von der Bearbeitung des SPS-Programms. Erst nach Ablauf des Zeitwerks erfolgt eine entsprechende Meldung des Betriebssystems an den zugehörigen Zeitbaustein im SPS-Programm.
- Die Bearbeitung eines Zeitwerks im Betriebssystem der SPS wird durch folgende Befehle nicht beeinflusst:
 - Programm abbrechen
 - Programm starten
 - Programm stoppen
 - Programm fortsetzen

D. h. ist ein Zeitwerk gestartet, so wird es im Betriebssystem der SPS auch dann weiter bearbeitet, wenn das zugehörige SPS-Programm abgebrochen, wieder gestartet oder gestoppt und wieder fortgesetzt wird.

also stets durch einen Kalt- oder Warmstart der SPS abgebrochen.

Initialisierung

Die Zeitwerke werden grundsätzlich bei jedem Kalt- oder Warmstart der SPS initialisiert. Ein laufendes Zeitwerk wird

Kaltstart:

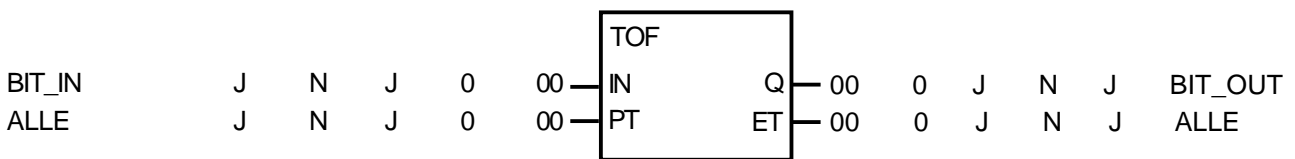
- Befehl KALT <CR>
- Erstmaliges Zuschalten der Spannung

Warmstart:

- Befehl WARM <CR>
- Zuschalten der Spannung

Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr	Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	----	-----	-----	-----	-----	--------------



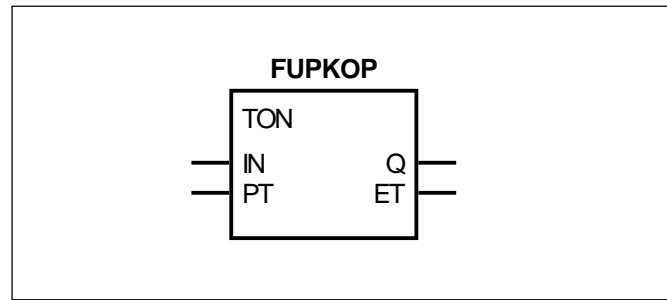
Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	!BA	000	
	00001	TOF		
	00002		PP 00	IN
	00003		PP 00	PT
	00004		PP 00	ET
	00005		PP 00	Q

Die 0-1-Flanke des Eingangs IN wird um die Zeitdauer PT verzögert und als 0-1-Flanke am Ausgang Q ausgegeben.

Der Ausgang ET zeigt die aktuelle Zeit an.

Geht der Eingang IN vor Ablauf der Zeit PT wieder auf 0-Pegel, so bleibt der Ausgang Q auf 0-Pegel.



Parameter

IN	BINÄR	E, M, A, S, K	Eingangssignal
PT	WORT/DOPPELWORT	MW&MW+1, MD,, KW&KW+1, KD, EW&EW+1, AW&AW+1	Verzögerungszeit
Q	BINÄR	M, A	Verzögertes Signal
ET	WORT/DOPPELWORT	MW&MW+1, MD	Zeitanzeige

Verfügbar ab: Advant Controller 31, Serie 40 und 50

Beschreibung

Die 0-1-Flanke des Eingangs IN wird um die Zeitdauer PT verzögert und als 0-1-Flanke am Ausgang Q ausgegeben.

Geht der Eingang IN vor Ablauf der Zeit PT wieder auf 0-Pegel, so bleibt der Ausgang Q auf 0-Pegel.

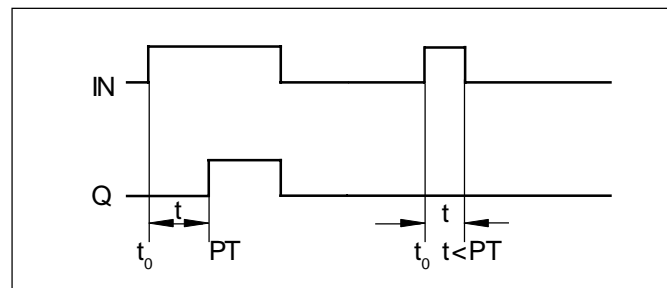
Die Istzeit wird am Ausgang ET angezeigt.

Für Zeiten, die kleiner als 65 s sind, können am Eingang PT bzw. Ausgang ET Variablen vom Typ WORT verwendet werden: Merker, Konstanten, andere WORT-Funktionen (+, -, AWT) und die Potentiometer-Eingänge der CPU.

In diesem Falle belegen die MW, KW, EW und AW zwei aufeinander folgende Adressen. MW+1, KW+1, EW+1 und AW+1 dürfen im Programm nicht mehr verwendet werden.

Die Zeitangabe erfolgt in Millisekunden.
Angebbarer Zeitbereich: 5 ms ... 24,8 Tage.

Bei der indirekten Konstanten KD wird der Wert der Impulslänge im Langtext angegeben.



Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit

Sinnvoller Bereich für PT: > 1 Zykluszeit

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar.

Allgemeines Verhalten

- Gestartete Zeitwerke werden vom Betriebssystem der SPS bearbeitet und sind deshalb vollkommen unabhängig von der Bearbeitung des SPS-Programms. Erst nach Ablauf des Zeitwerks erfolgt eine entsprechende Meldung des Betriebssystems an den zugehörigen Zeitbaustein im SPS-Programm.
- Die Bearbeitung eines Zeitwerks im Betriebssystem der SPS wird durch folgende Befehle nicht beeinflusst:
 - Programm abbrechen
 - Programm starten
 - Programm stoppen
 - Programm fortsetzen

D. h. ist ein Zeitwerk gestartet, so wird es im Betriebssystem der SPS auch dann weiter bearbeitet, wenn das zugehörige SPS-Programm abgebrochen, wieder gestartet oder gestoppt und wieder fortgesetzt wird.

also stets durch einen Kalt- oder Warmstart der SPS abgebrochen.

Initialisierung

Die Zeitwerke werden grundsätzlich bei jedem Kalt- oder Warmstart der SPS initialisiert. Ein laufendes Zeitwerk wird

Kaltstart:

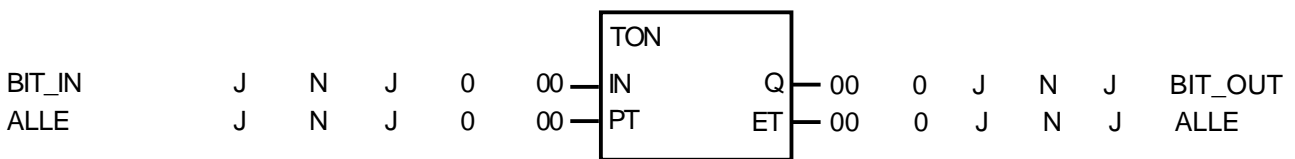
- Befehl KALT <CR>
- Erstmaliges Zuschalten der Spannung

Warmstart:

- Befehl WARM <CR>
- Zuschalten der Spannung

Grafik

Plaus-gruppe	Pf1	Inv	Anz	Art	Nr	Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	----	-----	-----	-----	-----	--------------



Source

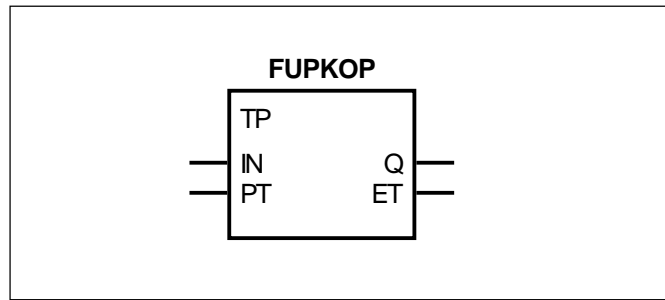
SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	!BA	000	
	00001	TON		
	00002	PP 00	IN	
	00003	PP 00	PT	
	00004	PP 00	ET	
	00005	PP 00	Q	

Die 0-1-Flanke des Eingangs IN bewirkt eine 0-1-Flanke am Ausgang Q. Nach Ablauf der Zeitdauer PT wird der Ausgang Q wieder auf 0-Pegel zurückgesetzt.

Der Ausgang ET zeigt die aktuelle Zeit an.

Eine zweite 0-1-Flanke des Eingangs IN vor Ablauf der Zeit PT wird ignoriert.

Der Ausgang ET zeigt die abgelaufene Zeit an.



Parameter

IN	BINÄR	E, M, A, S, K	Eingangssignal
PT	WORT/DOPPELWORT	MW&MW+1, MD, KW&KW+1, KD, EW&EW+1, AW&AW+1	Impulsdauer
Q	BINÄR	M, A	Impuls
ET	WORT/DOPPELWORT	MW&MW+1, MD	Zeitanzeige

Verfügbar ab: Advant Controller 31, Serie 40 und 50

Beschreibung

Die 0-1-Flanke des Eingangs IN bewirkt eine 0-1-Flanke am Ausgang Q. Nach Ablauf der Zeitdauer PT wird der Ausgang Q wieder auf 0-Pegel zurückgesetzt. Eine zweite 0-1-Flanke des Eingangs IN vor Ablauf der Zeit PT wird ignoriert.

Die Istzeit wird am Ausgang ET angezeigt.

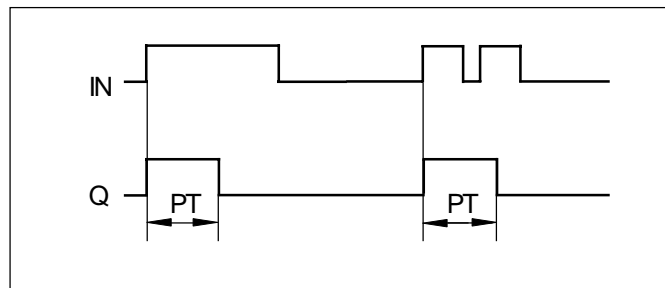
Für Zeiten, die kleiner als 65 s sind, können am Eingang PT bzw. Ausgang ET Variablen vom Typ WORT verwendet werden: Merker, Konstanten, andere WORT-Funktionen (+, -, AWT) und die Potentiometer-Eingänge der CPU.

In diesem Falle belegen die MW, KW, EW und AW zwei aufeinander folgende Adressen. MW+1, KW+1, EW+1 und AW+1 dürfen im Programm nicht mehr verwendet werden.

Die Zeitangabe erfolgt in Millisekunden.

Angebbarer Zeitbereich: 5 ms ... 24,8 Tage

Bei der indirekten Konstanten KD wird der Wert der Impulslänge im Langtext angegeben.



Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit

Sinnvoller Bereich für PT: > 1 Zykluszeit

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar.

Allgemeines Verhalten

- Gestartete Zeitwerke werden vom Betriebssystem der SPS bearbeitet und sind deshalb vollkommen unabhängig von der Bearbeitung des SPS-Programms. Erst nach Ablauf des Zeitwerks erfolgt eine entsprechende Meldung des Betriebssystems an den zugehörigen Zeitbaustein im SPS-Programm.
- Die Bearbeitung eines Zeitwerks im Betriebssystem der SPS wird durch folgende Befehle nicht beeinflusst:
 - Programm abbrechen
 - Programm starten
 - Programm stoppen
 - Programm fortsetzen

D. h. ist ein Zeitwerk gestartet, so wird es im Betriebssystem der SPS auch dann weiter bearbeitet, wenn das zugehörige SPS-Programm abgebrochen, wieder gestartet oder gestoppt und wieder fortgesetzt wird.

also stets durch einen Kalt- oder Warmstart der SPS abgebrochen.

Initialisierung

Die Zeitwerke werden grundsätzlich bei jedem Kalt- oder Warmstart der SPS initialisiert. Ein laufendes Zeitwerk wird

Kaltstart:

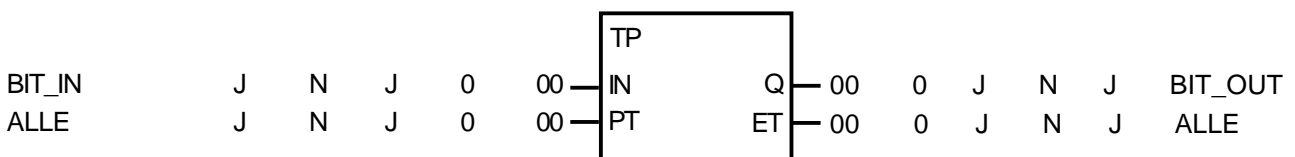
- Befehl KALT <CR>
- Erstmaliges Zuschalten der Spannung

Warmstart:

- Befehl WARM <CR>
- Zuschalten der Spannung
- RESET-Schalter

Grafik

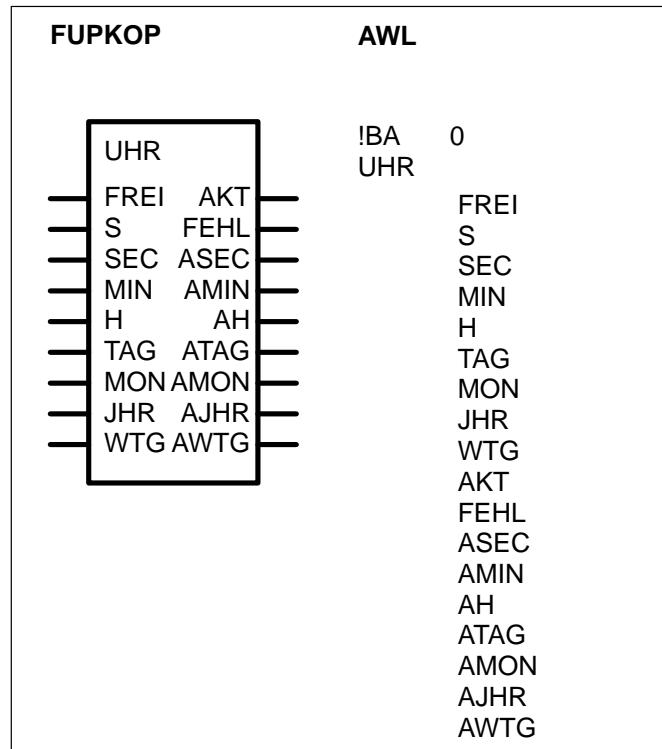
Plaus-gruppe	Pf1	Inv	Anz	Art	Nr	Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	----	-----	-----	-----	-----	--------------



Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	!BA	000	
	00001	TP		
	00002		PP 00	IN
	00003		PP 00	PT
	00004		PP 00	ET
	00005		PP 00	Q

Der Funktionsbaustein ermöglicht das Stellen und Anzeigen der aktuellen Uhrzeit und des aktuellen Datums.



Parameter

FREI	BINÄR	E, A, M, K, S	Freigabe für die Bearbeitung des Bausteins
S	BINÄR	E, A, M, K, S	0/1–Flanke bewirkt das Stellen von Uhrzeit und Datum
SEC	WORT	EW, AW, MW, KW	Stelleingang für die Sekunden
MIN	WORT	EW, AW, MW, KW	Stelleingang für die Minuten
H	WORT	EW, AW, MW, KW	Stelleingang für die Stunden
TAG	WORT	EW, AW, MW, KW	Stelleingang für die Tage
MON	WORT	EW, AW, MW, KW	Stelleingang für die Monate
JHR	WORT	EW, AW, MW, KW	Stelleingang für die Jahre
WTG	WORT	EW, AW, MW, KW	Stelleingang für die Wochentage
AKT	BINÄR	A, M	Aktualität (Brauchbarkeit) der Daten an den Ausgängen
FEHL	WORT	AW, MW	Fehlerkennung
ASEC	WORT	AW, MW	Ausgang Sekunden
AMIN	WORT	AW, MW	Ausgang Minuten
AH	WORT	AW, MW	Ausgang Stunden
ATAG	WORT	AW, MW	Ausgang Tage
AMON	WORT	AW, MW	Ausgang Monate
AJHR	WORT	AW, MW	Ausgang Jahre
AWTG	WORT	AW, MW	Ausgang Nr. des Wochentages

VE–Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	60 µs – Anzeigen von Zeit und Datum	409 µs
	200 µs – Stellen von Zeit und Datum	4697 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	1 Wort	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein ermöglicht das Stellen und Anzeigen der aktuellen Uhrzeit und des aktuellen Datums.

Die Eingänge und Ausgänge sind weder doppelbar noch invertierbar noch negierbar.

Das Stellen der Uhr erfolgt über die Stelleingänge für die Uhrzeit und das Datum. Die an den Stelleingängen anliegenden Werte werden durch eine 0/1-Flanke am Eingang S übernommen. Solange am Eingang FREI ein 1-Signal anliegt, werden an den Bausteinausgängen die aktuelle Uhrzeit und das Datum angezeigt.

FREI BINÄR

Freigabe des Bausteins

FREI = 0: Der Baustein wird nicht bearbeitet. Die Ausgänge AKT und FEHL werden auf 0 gesetzt. Die Zeit- und Datumsausgänge werden vom Baustein nicht mehr verändert.

FREI = 1: Baustein wird bearbeitet

S BINÄR

0/1-Flanke → die Uhr wird auf die an den Zeit- und Datumseingängen anstehenden Werte gestellt.

Während des Stellvorgangs sind die Uhrzeit und das Datum an den Bausteinausgängen ungültig (Ausgang AKT = 0).

Stelleingänge für Datum und Uhrzeit

Bei einer 0/1-Flanke am Eingang S wird die Uhr auf die an den Stelleingängen anliegenden Werte gesetzt. Sind die angegebenen Stellwerte unzulässig, wird der Ausgang AKT auf 0 gesetzt und am Ausgang FEHL erfolgt eine Fehlermeldung. Die an den Zeit- und Datumsausgängen anliegenden Werte sind in diesem Fall ungültig. Die Uhr muß noch einmal gestellt werden.

SEC WORT

Stelleingang für die Sekunden.
Wertebereich: 0 ... 59.

MIN WORT

Stelleingang für die Minuten.
Wertebereich: 0 ... 59.

H WORT

Stelleingang für die Stunden.

Die Uhr arbeitet im 24-h-Mode, d. h. sie springt von 23:59:59 Uhr auf 0:0:0 Uhr.
Wertebereich: 0 ... 23.

TAG WORT

Stelleingang für die Tage (wievielter Tag des Monats)

Nicht für 07 KR 31 Die Uhr kennt die Anzahl der Tage in Abhängigkeit von den Monaten und Schaltjahren. Ein Schaltjahr besteht für die Uhr dann, wenn die Jahreszahl ein ganzzahliges Vielfaches von 4 ist. Der maximale Wert für die Tage (28, 29, 30, 31) ist abhängig vom Monat.
Wertebereich: 1...28, 29, 30, 31.

MON WORT

Stelleingang für die Monate.
Wertebereich: 1 ... 12.

JHR WORT

Stelleingang für die Jahre.

Die Uhr zeigt nur die Jahre und Jahrzehnte an.
Wertebereich: 0 ... 99.

WTG WORT

Stelleingang für die Nummer des Wochentags.

An diesem Eingang wird angegeben, der wievielte Wochentag der Tag ist, an dem die Eingabe erfolgt. D. h. man kann festlegen, welcher Wochentag (z. B. Sonntag oder Montag) der Tag mit der Nummer 1 sein soll.
Wertebereich: 1 ... 7.

Beispiel:

Die Uhr wird am Freitag, den 01.07.88 gestellt. Wird für die Wochentagsnummer der Wert 6 eingegeben, so ist nun der Freitag als 6. Tag der Woche und damit auch der Sonntag als 1. Tag der Woche definiert.

AKT BINÄR

Anzeige der Aktualität (Brauchbarkeit) der Ausgänge.

AKT ist 1, wenn:

- die Datums- und Zeitausgänge im aktuellen Zyklus aktualisiert wurden
- die Werte an den Ausgängen konsistent sind, d. h. keiner der Werte an den Datums- bzw. Zeitausgängen hat sich während des Aktualisierungsvorgangs geändert. Sie stammen alle vom selben Uhrentakt ab
- die Uhr richtig gestellt wurde

AKT = 1 → FEHL = 0: Datum / Uhrzeit sind *gültig*

AKT = 0 → FEHL > 0: Datum / Uhrzeit sind *ungültig*
Der Grund wird als Fehlerkennung am Ausgang FEHL angezeigt

FEHL WORT

Am Ausgang FEHL steht im Fehlerfall die relevante Fehlerkennung zur Verfügung.

Bedeutung der Fehlerkennungen:

– **Kein Fehler aufgetreten:**

FEHL=0: kein Fehler aufgetreten *oder* FREI = 0
d. h. Baustein gesperrt

– **Fehler beim Stellen der Uhr:**

FEHL=1: $0 \leq \text{SEC} \leq 59$ ist nicht eingehalten

FEHL=2: $0 \leq \text{MIN} \leq 59$ ist nicht eingehalten

FEHL=3: $0 \leq \text{H} \leq 23$ ist nicht eingehalten

FEHL=4: $1 \leq \text{TAG} \leq 28, 29, 30, 31$ (je nach Monat, in 07 KR 31 nicht getestet) ist nicht eingehalten

FEHL=5: $1 \leq \text{MON} \leq 12$ ist nicht eingehalten

FEHL=6: $0 \leq \text{JHR} \leq 99$ ist nicht eingehalten

Die folgenden Fehler werden in 07 KR 31 nicht benutzt:

FEHL=7: $1 \leq \text{WTG} \leq 7$ ist nicht eingehalten

FEHL=8: Die Sendemailbox ist momentan durch einen anderen Auftraggeber belegt. Der Baustein wartet bis die Mailbox frei wird und führt dann den Stellvorgang für Datum / Uhrzeit aus.

FEHL=9: Datum / Uhrzeit an den Ausgängen sind *ungültig*

FEHL=10: Der Stellvorgang von Datum / Uhrzeit läuft gerade; er kann mehrere SPS-Zyklen dauern

FEHL=11: Stellvorgang war nicht erfolgreich, bitte wiederholen (unbekannter Requestcode)

FEHL=12: Stellvorgang war nicht erfolgreich, bitte wiederholen (Mail-Parameter ungültig)

FEHL=13: Stellvorgang war nicht erfolgreich, bitte wiederholen (Requestcode nicht ausführbar)

– **Fehler beim Anzeigen des Datums und der Uhrzeit:**

FEHL=9: Datum / Uhrzeit an den Ausgängen sind *ungültig*

Ausgänge für Datum und Uhrzeit

Die Aktualisierung der Ausgänge erfolgt immer dann, wenn am Eingang FREI ein 1-Signal anliegt *und* die Uhr einmal gestellt worden ist. *Während* des Stellvorgangs sind die Ausgänge für das Datum und die Uhrzeit *ungültig*.

Wenn der Ausgang AKT gleich 1 ist, sind die Ausgänge für Datum und Uhrzeit gültig. Im Fehlerfall wird am Ausgang FEHL eine Fehlerkennung ausgegeben.

ASEC WORT

Ausgang Sekunden.
Wertebereich: 0 ... 59.

AMIN WORT

Ausgang Minuten.
Wertebereich: 0 ... 59.

AH WORT

Ausgang Stunden.
Wertebereich: 0 ... 23.

ATAG WORT

Ausgang Tage.
Wertebereich: 1 ... 28, 29, 30, 31.

AMON WORT

Ausgang Monate.
Wertebereich: 1 ... 12.

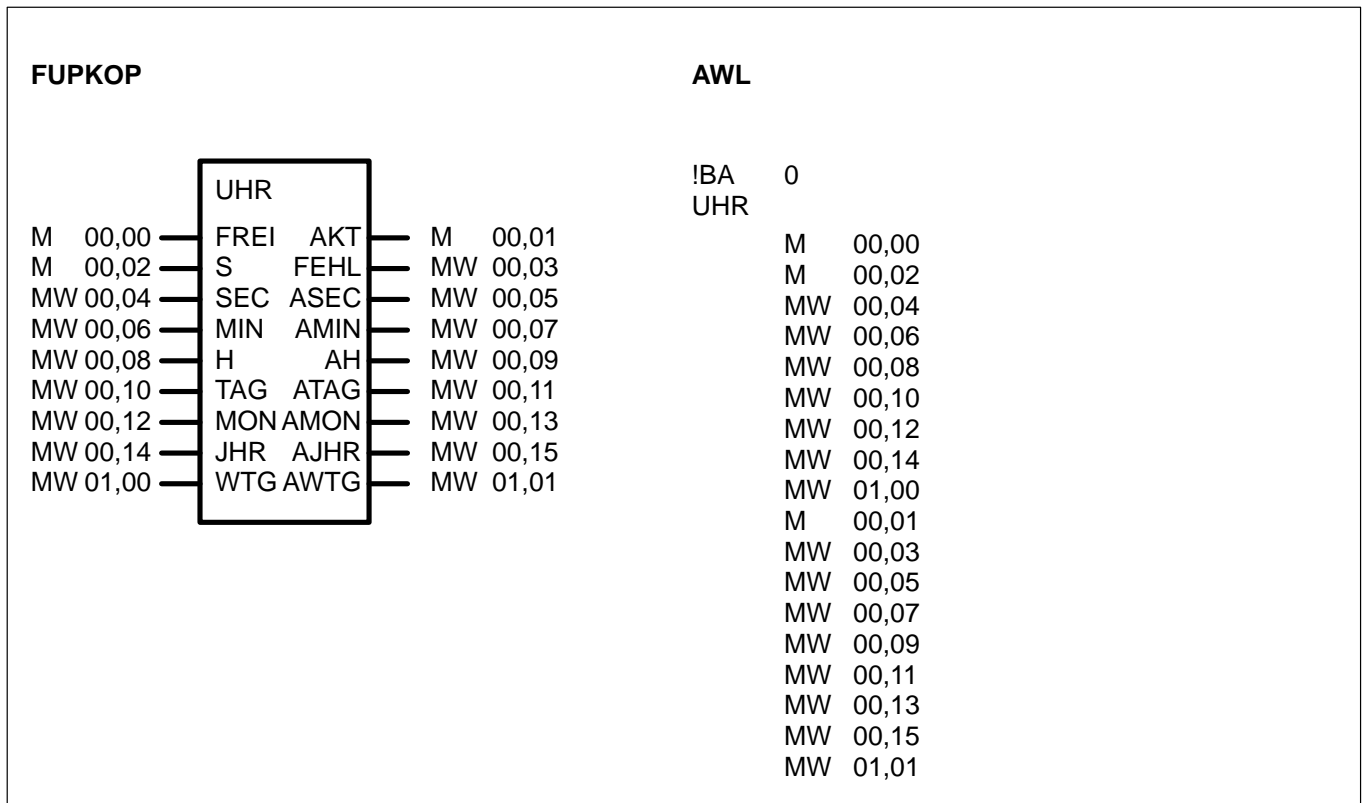
AJHR WORT

Ausgang Jahre.
Wertebereich: 0 ... 99.

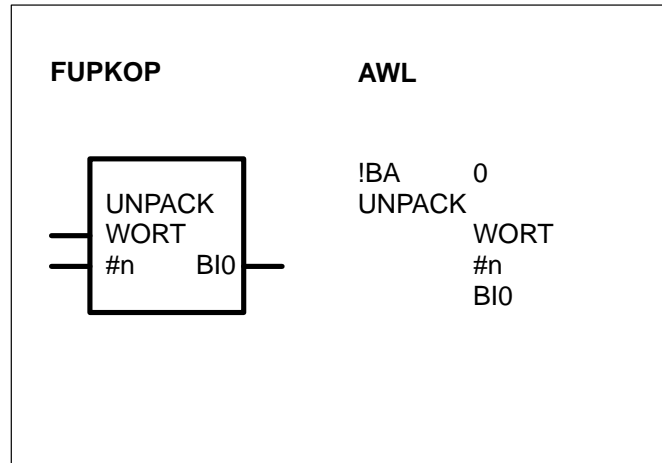
AWTG WORT

Ausgang Nr. des Wochentages.
Wertebereich: 1 ... 7.

Beispiel



Der Funktionsbaustein entpackt die Wortvariable am Eingang WORT. Jedes Bit der Eingangsvariablen wird jeweils einer Binär-Variablen (BI0 ... BIn) am Ausgang zugewiesen.



Parameter

WORT	WORT	EW, AW, MW, KW	zu entpackende Wortvariable
#n	DIREKTE KONSTANTE	#, #H	Anzahl der Ausgangsvariablen BI0 ... BIn-1
BI0	BINÄR	A, M	1.binäre Ausgangsvariable; Ausgang ist doppelbar

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

Zusätzliche Laufzeit:

23,5 µs

9,5 µs

pro projektiertes
Binär-Variablen

32 µs

33 µs

pro projektiertes
Binär-Variablen

Aktualisierung der Ausgänge:

ja

ja

Anzahl der Vergangenheitswerte:

keine

keine

Verfügbar ab:

ABB Procontic CS31 / 907 PC 331

907 PC 331 / 07 KR 31 V 1.0,

07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein entpackt die Wortvariable am Eingang WORT. Jedes Bit der Eingangsvariablen wird jeweils einer Binär-Variablen (BI0 ... BIn-1) am Ausgang zugewiesen.

WORT WORT

Am Eingang WORT wird die zu entpackende Variable angegeben. Jedes Bit (Bit0 ... Bit15) dieser Eingangsvariablen wird der zugeordneten Ausgangsvariablen (BI0 ... BIn-1) zugewiesen.

#n DIREKTE KONSTANTE (#,#H)

Am Eingang #n wird die Anzahl der projektierten binären Ausgänge (BI0 ... BIn-1) angegeben. Die Angabe erfolgt als direkte Konstante.

Es gilt: $1 \leq n \leq 16$

BI0 BINÄR

Der Ausgang BI0 ist doppelbar (BI0 ... BIn-1). Den binären Ausgängen BI0 ... BIn werden die zugeordneten Bits der Variablen am Eingang WORT zugewiesen.

Zuordnung

Eingangsvariable Bit0 → BI0

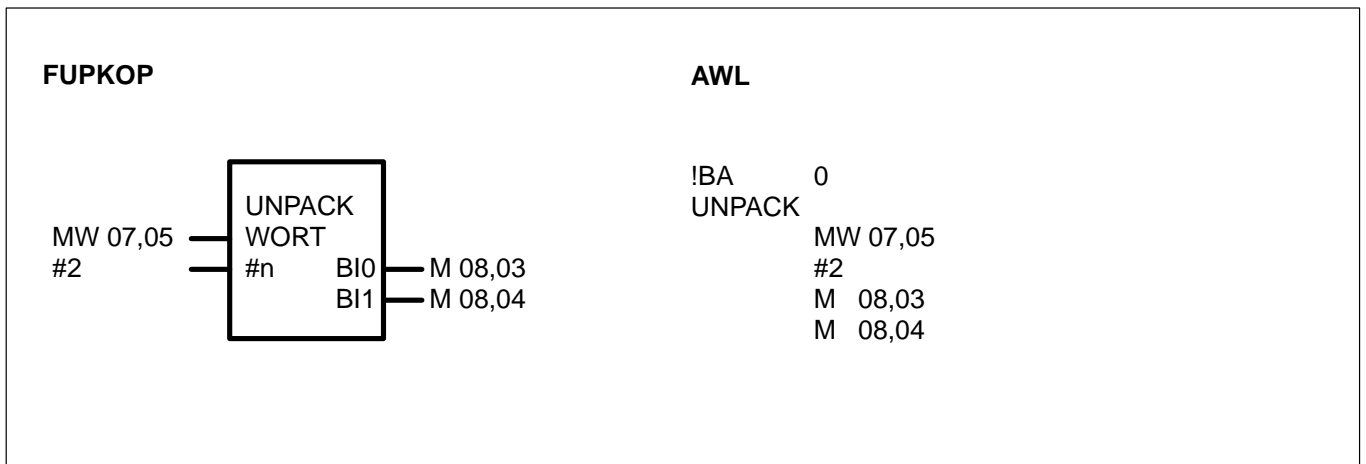
Eingangsvariable Bit1 → BI1

. . . .

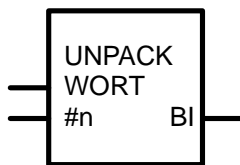
. . . .

Eingangsvariable Bit15 → BI15

Beispiel



VE-FUP-Definition

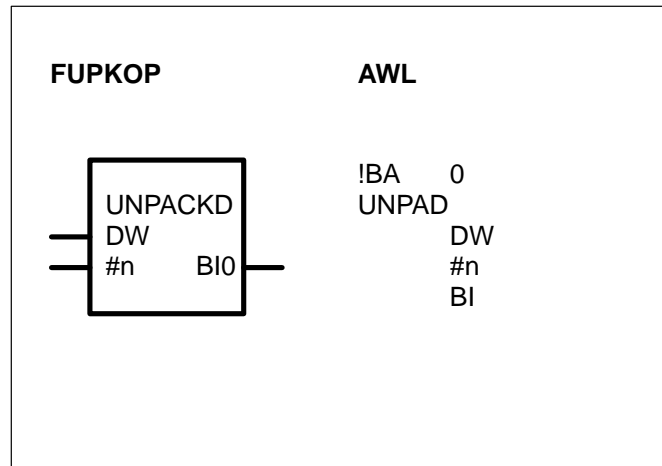


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
WORT	E	W	N	P	J	0	0
#n	K	W	N	P	J	0	0
BI	A	L	N	P	J	1	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	UNPACK			
00002		PP 0	WORT	Eingang WORT
00003		PP 0	#n	# KONSTANTE (Anzahl BIT)
	[1		
00004		PP 1	BI	Ausgang BINÄR
]	1		

Der Funktionsbaustein entpackt die Doppelwortvariable am Eingang DW. Jedes Bit der Eingangsvariablen wird jeweils einer Binär-Variablen (BI0 ... BI_n) am Ausgang zugewiesen.



Parameter

DW DOPPELWORT MD, KD
 #n WORT #,#H
 BI0 BINÄR A, M

zu entpackende Doppelwortvariable
 Anzahl der Ausgangsvariablen BI0 ... BI_{n-1}
 1. binäre Ausgangsvariable; Ausgang ist doppelbar

VE-Daten

Laufzeit:
 Grundlaufzeit: 40 µs
 Zusätzliche Laufzeit: 9 µs, pro Ausgang
 Aktualisierung der Ausgänge: ja
 Anzahl der Vergangenheitswerte: keine
 Verfügbar ab: ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein entpackt die Doppelwortvariable am Eingang DW. Jedes Bit der Eingangsvariablen wird jeweils einer Binär-Variablen (BI0 ... BI_{n-1}) am Ausgang zugewiesen.

Zuordnung

Eingangsvariable Bit0 → BI0
 Eingangsvariable Bit1 → BI1
 : :
 : :
 : :
 Eingangsvariable Bit31 → BI31

DW DOPPELWORT

Am Eingang DW wird die zu entpackende Variable angegeben. Jedes Bit (Bit0 ... Bit31) dieser Eingangsvariablen wird der zugeordneten Ausgangsvariablen (BI0 ... BI_{n-1}) zugewiesen.

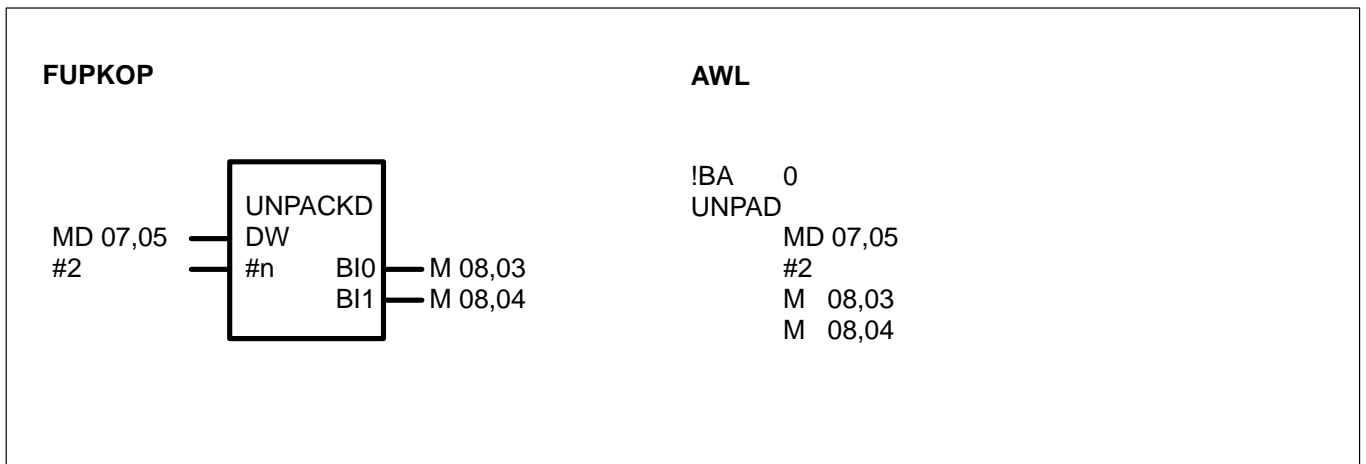
#n DIREKTE KONSTANTE (#,#H)

Am Eingang #n wird die Anzahl der projektierten binären Ausgänge (BI0 ... BI_{n-1}) angegeben. Die Angabe erfolgt als direkte Konstante.
 Es gilt: 1 ≤ n ≤ 32

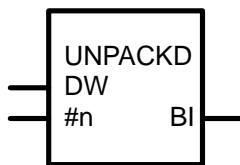
BI0 BINÄR

Der Ausgang BI0 ist doppelbar (BI0 ... BI_{n-1}). Den binären Ausgängen BI0 ... BI_n werden die zugeordneten Bits der Variablen am Eingang DW zugewiesen.

Beispiel



VE-FUP-Definition



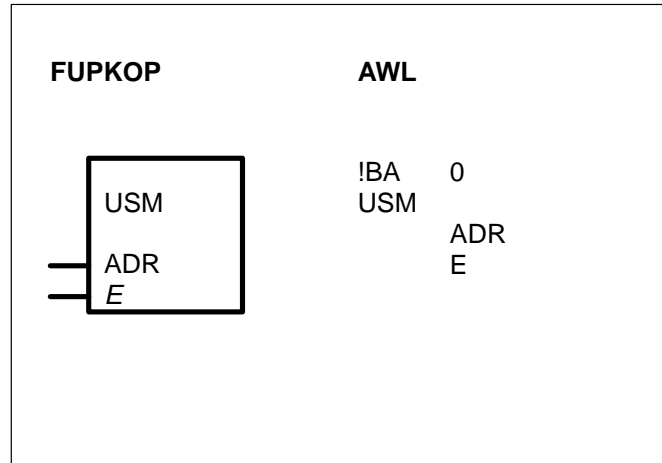
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
DW	E	D	N	P	J	0	0
#n	K	W	N	P	J	0	0
BI	A	L	N	P	J	1	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	UNPAD			
00002		PP 0	DW	Eingang DOPPELWORT
00003		PP 0	#n	# KONSTANTE (Anzahl BIT)
	[1		
00004		PP 1	BI	Ausgang BINÄR
]	1		

Der Funktionsbaustein weist den Wert vom Eingang E einem Operanden zu, wobei das Verfahren der indirekten Adressierung benutzt wird.

Hinweis: Der Baustein USM kann nur sinnvoll in Verbindung mit dem Baustein ADRWA benutzt werden.



Parameter

ADR	WORT	EW, AW, MW, KW	Indirekte Adresse des zu schreibenden Operanden Wert, der dem Operanden zuzuweisen ist
E	WORT	AW, MW	

VE-Daten

Laufzeit:	
Grundlaufzeit:	225 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein weist den Wert vom Eingang E einem Operanden zu, wobei das Verfahren der indirekten Adressierung benutzt wird.

Hinweis: Der Baustein USM kann nur sinnvoll in Verbindung mit dem Baustein ADRWA benutzt werden.

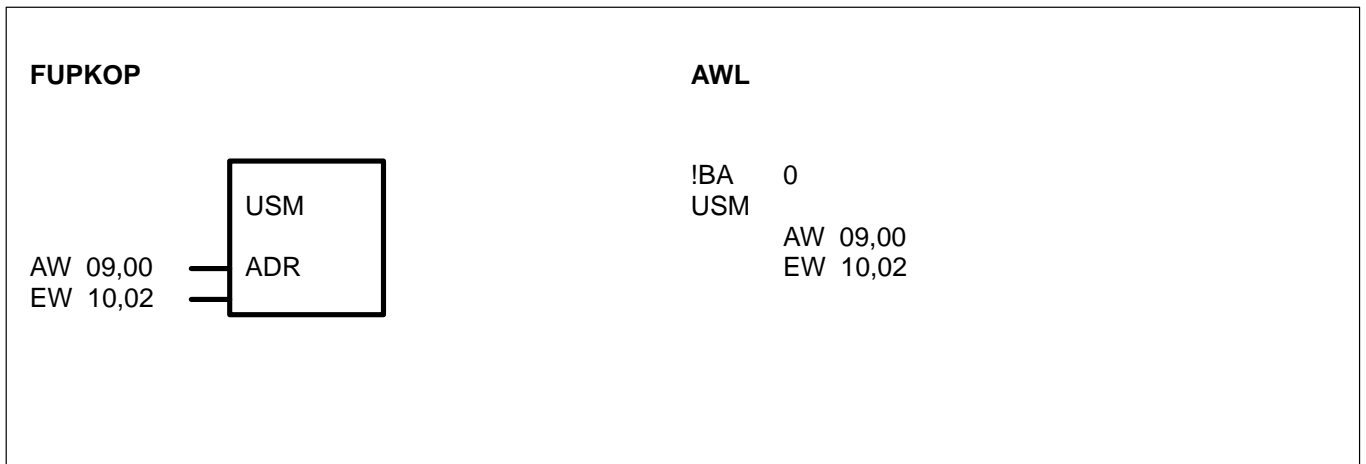
Der Wert des Operanden am Eingang ADR wird als Adresse des zu schreibenden Operanden interpretiert (indirekte Adressierung).

Der Operand am Eingang ADR stellt also zusammen mit seinem Wert eine indirekte Adresse dar. Diese indirekte Adresse wird vom Funktionsbaustein ADRWA erzeugt.

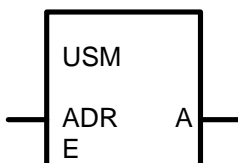
Die Eingänge sind weder doppelbar noch negierbar.

Hinweis: Die Erläuterung des Verfahrens der indirekten Adressierung und die Verwendungsmöglichkeit des Funktionsbausteins USM ist beim Funktionsbaustein ADRWA beschrieben.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
ADR	E	W	N	P	J	0	0
E	E	W	N	P	N	0	0

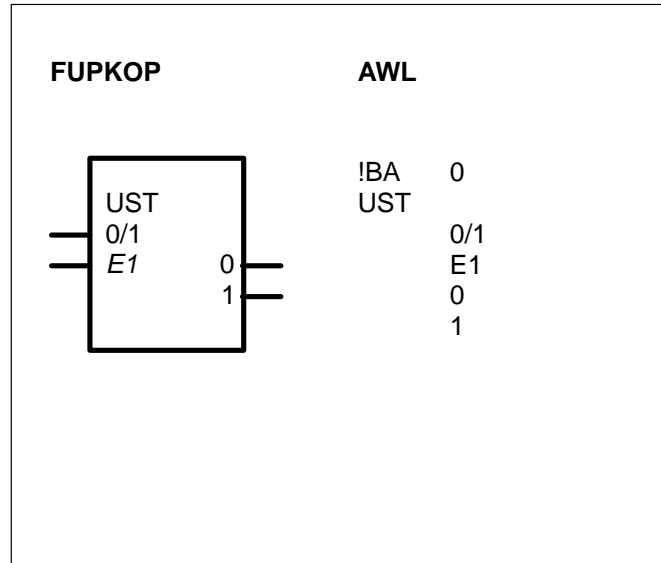
VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	USM			
00002		PP 0	ADR	Eingang WORT (Operandenadresse)
00003		PP 0	E	Eingang WORT (Operandenwert)

0-Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang 0 den Wert des Wortoperanden am Eingang E1 zu.

1-Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang 1 den Wert des Wortoperanden am Eingang E1 zu.

Der jeweils nicht zugewiesene Ausgang behält seinen alten Wert bei, wobei aber keine Aktualisierung des alten Wertes erfolgt.



Parameter

0/1	BINÄR	E, M, A, S, K	Umschalteingang
E1	WORT	EW, MW, AW, KW	Eingang
0	WORT	AW, MW	Ausgang für 0/1 = 0
1	WORT	AW, MW	Ausgang für 0/1 = 1

VE-DATEN

Laufzeit:	
Grundlaufzeit:	31 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	nein; Empfehlung: Ausgang direkt an Merker zuweisen oder Zwischenmerker global vergeben; siehe Griff 3, Kapitel 2.5.3
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

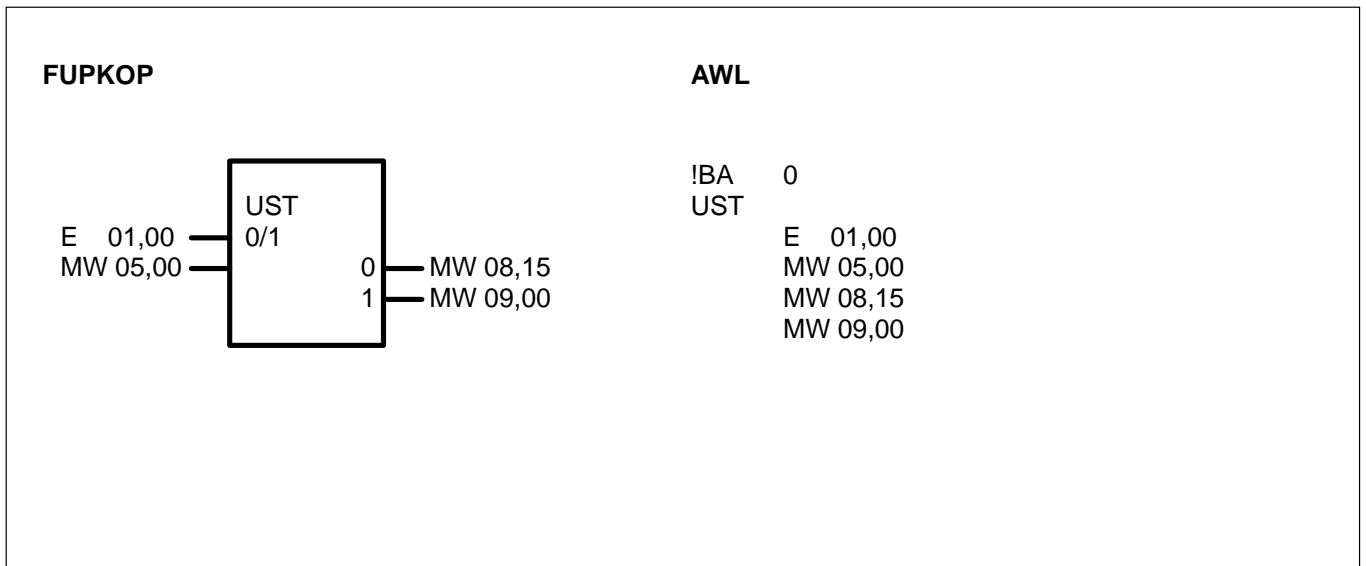
0-Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang 0 den Wert des Wortoperanden am Eingang E1 zu.

1-Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang 1 den Wert des Wortoperanden am Eingang E1 zu.

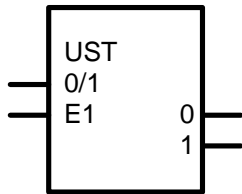
Der jeweils nicht zugewiesene Ausgang behält seinen alten Wert bei, wobei aber keine Aktualisierung des alten Wertes erfolgt.

Die Ein- und Ausgänge sind weder doppelbar noch negierbar/invertierbar.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0/1	E	L	N	P	J	0	0
E1	E	W	N	P	N	0	0
0	A	W	N	P	J	0	0
1	A	W	N	P	J	0	0

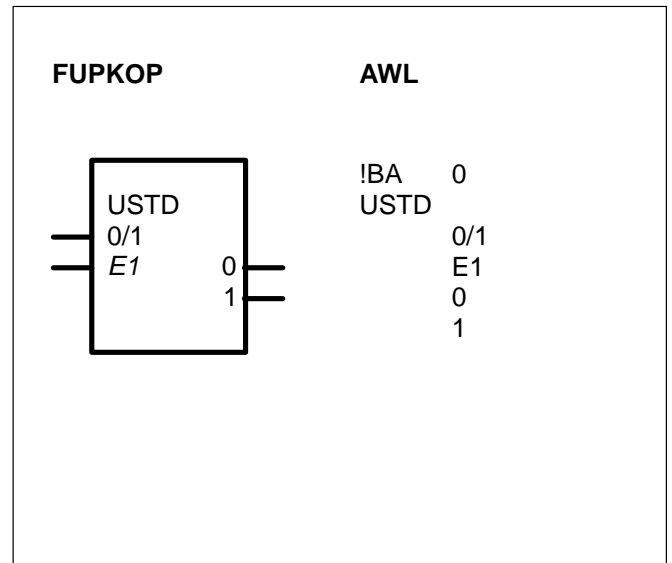
VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	UST			
00002		PP 0	0/1	Eingang BINÄR (Selektor)
00003		PP 0	E1	Eingang WORT
00004		PP 0	0	Ausgang WORT
00005		PP 0	1	Ausgang WORT

0–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang 0 den Wert des Doppelwortoperanden am Eingang E1 zu.

1–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang 1 den Wert des Doppelwortoperanden am Eingang E1 zu.

Der jeweils nicht zugewiesene Ausgang behält seinen alten Wert bei, wobei aber keine Aktualisierung des alten Wertes erfolgt.



Parameter

0/1	BINÄR	E, M, A, S, K	Umschalteingang
E1	DOPPELWORT	MD, KD	Eingang
0	DOPPELWORT	MD	Ausgang für 0/1 = 0
1	DOPPELWORT	MD	Ausgang für 0/1 = 1

VE-DATEN

Laufzeit:		
Grundlaufzeit:		35 µs
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:	nein;	Empfehlung: Ausgang direkt an Merker zuweisen oder Zwischenmerker global vergeben; siehe Griff 3, Kapitel 2.5.3
Anzahl der Vergangenheitswerte:	keine	
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

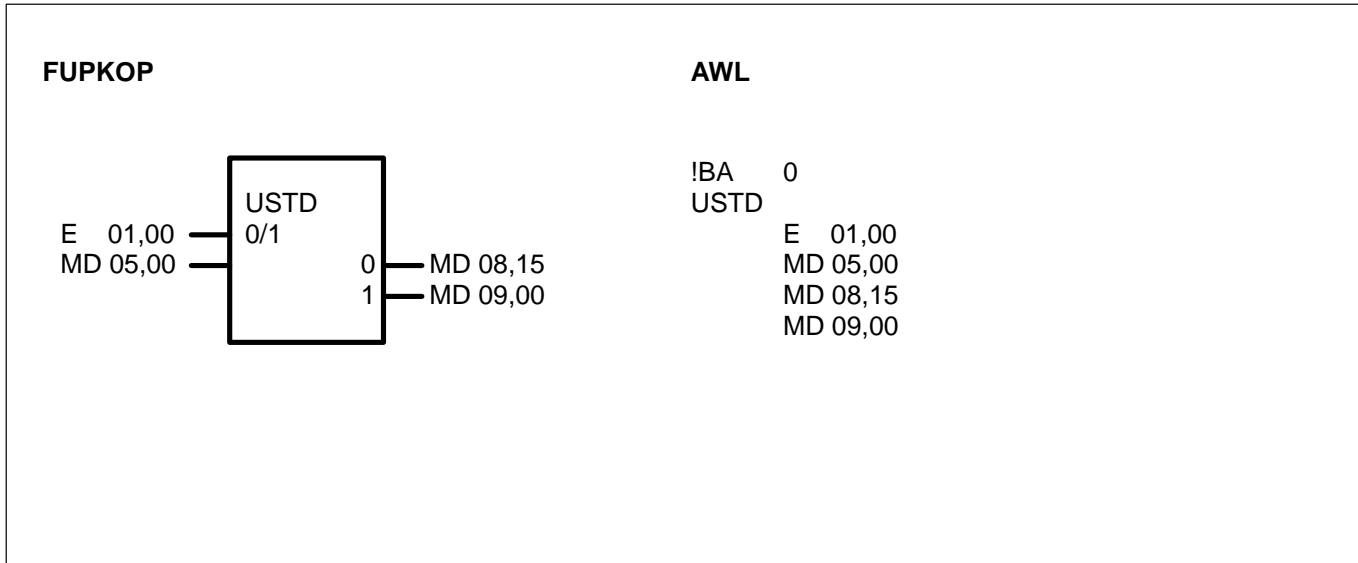
0–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang 0 den Wert des Doppelwortoperanden am Eingang E1 zu.

1–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang 1 den Wert des Doppelwortoperanden am Eingang E1 zu.

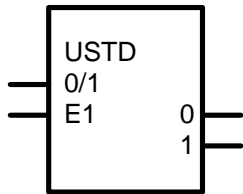
Der jeweils nicht zugewiesene Ausgang behält seinen alten Wert bei, wobei aber keine Aktualisierung des alten Wertes erfolgt.

Die Ein- und Ausgänge sind weder doppelbar noch negierbar/invertierbar.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0/1	E	L	N	P	J	0	0
E1	E	D	N	P	N	0	0
0	A	D	N	P	J	0	0
1	A	D	N	P	J	0	0

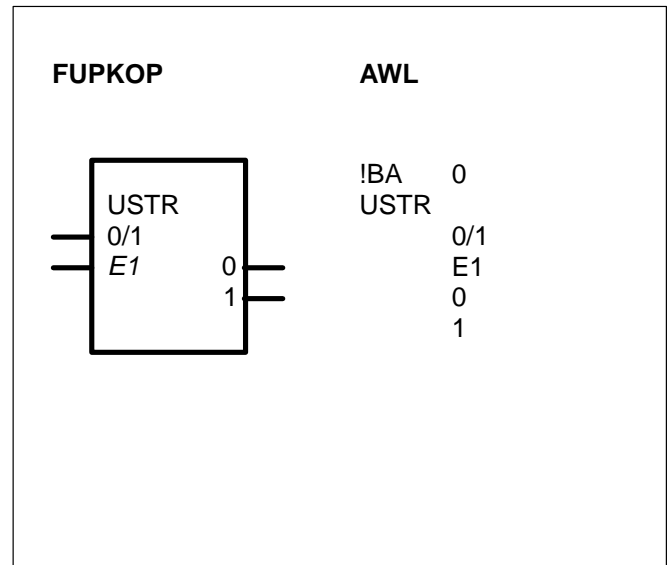
VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	USTD			
00002		PP 0	0/1	Eingang BINÄR (Selektor)
00003		PP 0	E1	Eingang DOPPELWORT
00004		PP 0	0	Ausgang DOPPELWORT
00005		PP 0	1	Ausgang DOPPELWORT

0–Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang 0 den Wert des Wortoperanden am Eingang E1 zu.

1–Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang 1 den Wert des Wortoperanden am Eingang E1 zu.

Der jeweils nicht zugewiesene Ausgang wird zu Null gesetzt.



Parameter

0/1	BINÄR	E, M, A, S, K	Umschalteingang
E1	WORT	EW, MW, AW, KW	Eingang
0	WORT	AW, MW	Ausgang für 0/1 = 0
1	WORT	AW, MW	Ausgang für 0/1 = 1

VE-DATEN

Laufzeit:	
Grundlaufzeit:	35 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

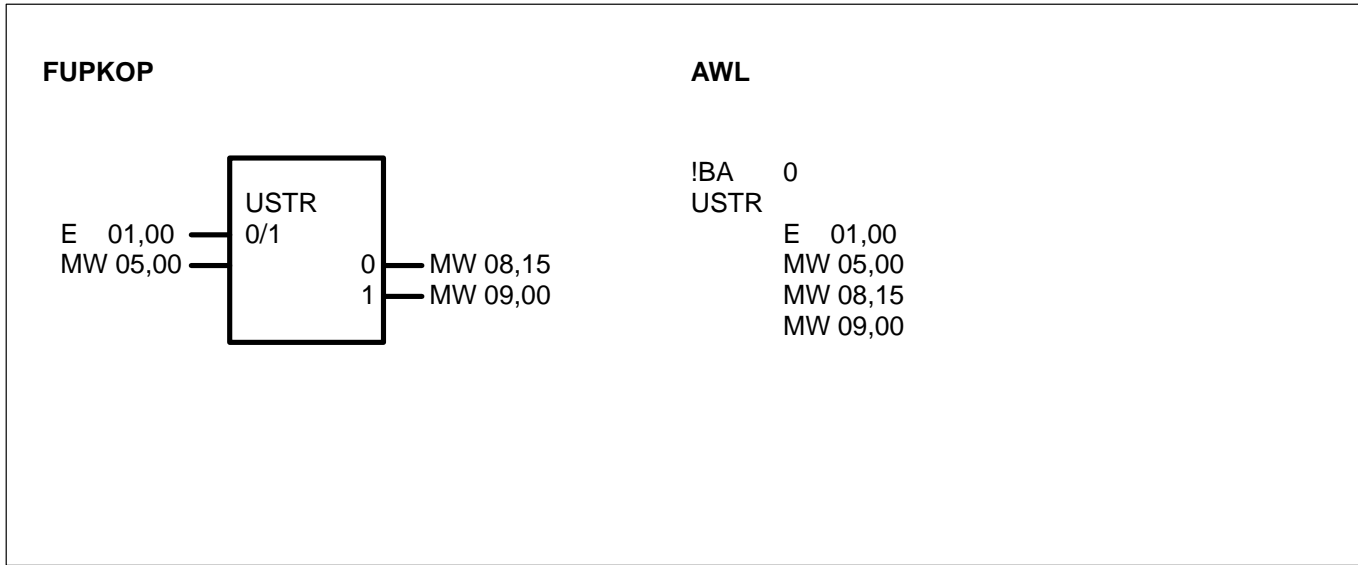
0–Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang 0 den Wert des Wortoperanden am Eingang E1 zu.

1–Signal am binären Eingang 0/1 weist dem Wortoperanden am Ausgang 1 den Wert des Wortoperanden am Eingang E1 zu.

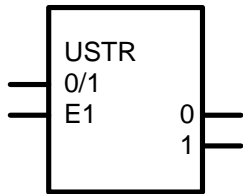
Der jeweils nicht zugewiesene Ausgang wird zu Null gesetzt.

Die Ein- und Ausgänge sind weder doppelbar noch negierbar/invertierbar.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0/1	E	L	N	P	J	0	0
E1	E	W	N	P	N	0	0
0	A	W	N	P	J	0	0
1	A	W	N	P	J	0	0

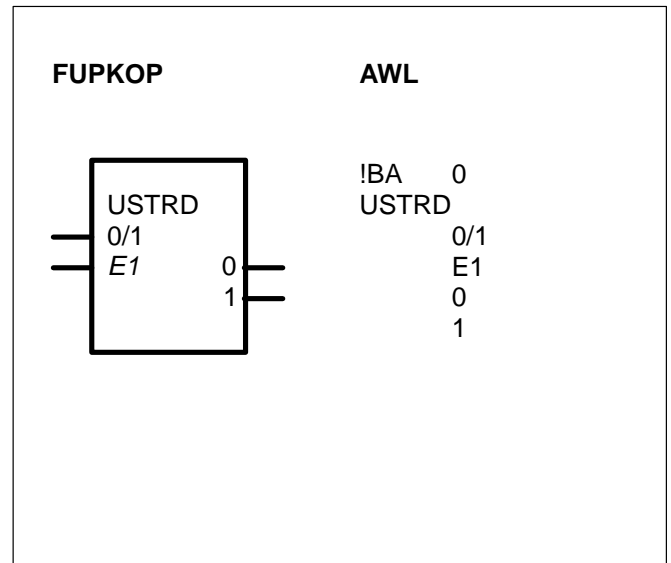
VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	USTR			
00002		PP 0	0/1	Eingang BINÄR (Selektor)
00003		PP 0	E1	Eingang WORT
00004		PP 0	0	Ausgang WORT
00005		PP 0	1	Ausgang WORT

0–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang 0 den Wert des Doppelwortoperanden am Eingang E1 zu.

1–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang 1 den Wert des Doppelwortoperanden am Eingang E1 zu.

Der jeweils nicht zugewiesene Ausgang wird zu Null gesetzt.



Parameter

0/1	BINÄR	E, M, A, S, K	Umschalteingang
E1	DOPPELWORT	MD, KD	Eingang
0	DOPPELWORT	MD	Ausgang für 0/1 = 0
1	DOPPELWORT	MD	Ausgang für 0/1 = 1

VE-DATEN

Laufzeit:		
Grundlaufzeit:		43 µs
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

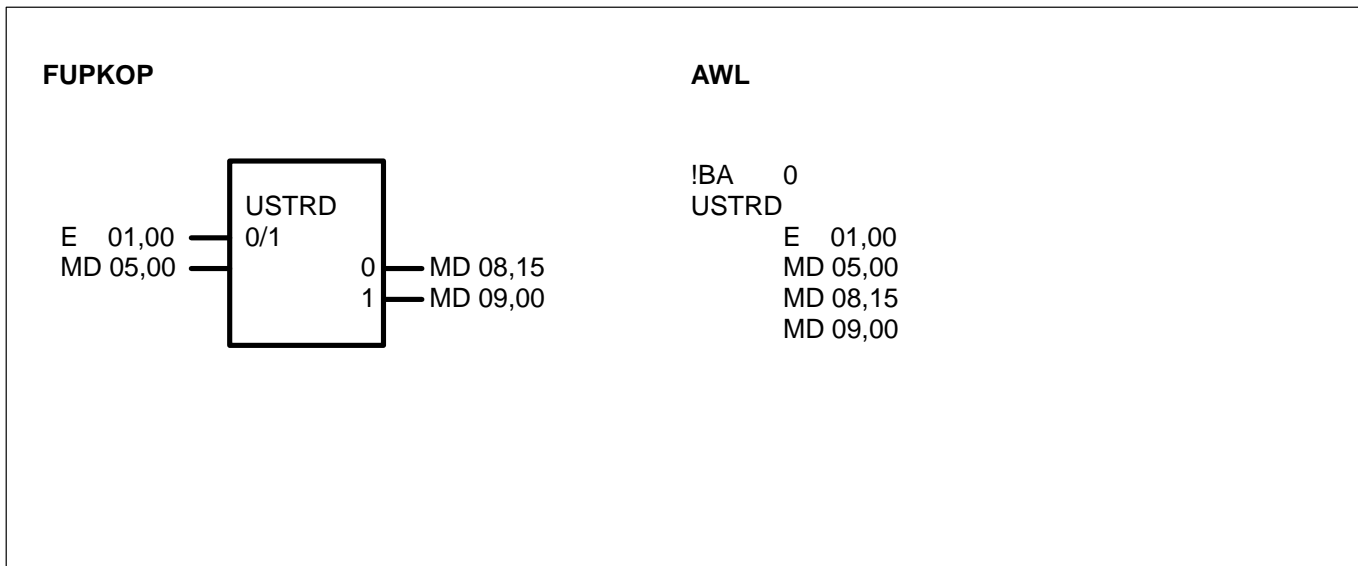
0–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang 0 den Wert des Doppelwortoperanden am Eingang E1 zu.

1–Signal am binären Eingang 0/1 weist dem Doppelwortoperanden am Ausgang 1 den Wert des Doppelwortoperanden am Eingang E1 zu.

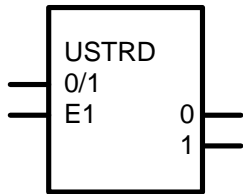
Der jeweils nicht zugewiesene Ausgang wird zu Null gesetzt.

Die Ein- und Ausgänge sind weder doppelbar noch negierbar/invertierbar.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
0/1	E	L	N	P	J	0	0
E1	E	D	N	P	N	0	0
0	A	D	N	P	J	0	0
1	A	D	N	P	J	0	0

VE-AWL-Definition

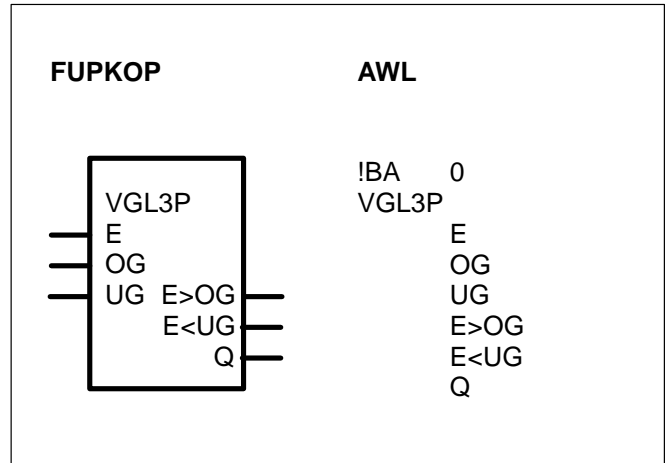
00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	USTRD			
00002		PP 0	0/1	Eingang BINÄR (Selektor)
00003		PP 0	E1	Eingang DOPPELWORT
00004		PP 0	0	Ausgang DOPPELWORT
00005		PP 0	1	Ausgang DOPPELWORT

Der Wert des Operanden am Eingang E wird mit den Werten der Operanden am Eingang OG und UG verglichen.

Die möglichen Ergebnisse werden an den Ausgängen E>OG, E<UG und Q signalisiert.

Es gilt:

- E < UG —> E>OG = 0, E<UG = 1, Q = 0
- UG ≤ E ≤ OG —> E>OG = 0, E<UG = 0, Q = 1
- E > OG —> E>OG = 1, E<UG = 0, Q = 0



Parameter

E	WORT	EW, MW, AW, KW	Eingangswert
OG	WORT	EW, MW, AW, KW	obere Grenze
UG	WORT	EW, MW, AW, KW	untere Grenze
E>OG	BINÄR	A, M	Wert > obere Grenze
E<UG	BINÄR	A, M	Wert < untere Grenze
Q	BINÄR	A, M	untere Grenze ≤ Eingangswert ≤ obere Grenze

VE-Daten

Laufzeit:	
Grundlaufzeit:	48 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

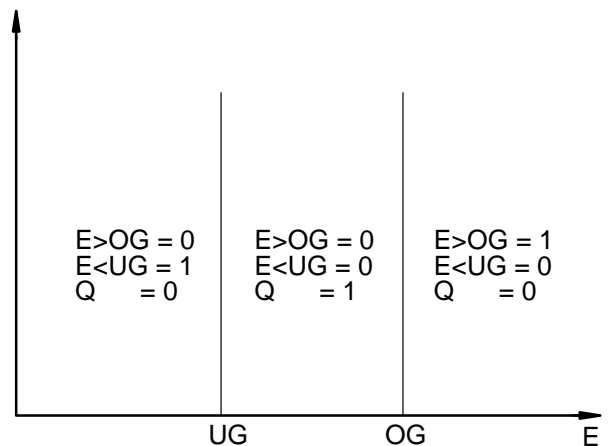
Der Wert des Operanden am Eingang E wird mit den Werten der Operanden am Eingang OG und UG verglichen.

Die möglichen Ergebnisse werden an den Ausgängen E>OG, E<UG und Q signalisiert.

Es gilt:

- E < UG —> E>OG = 0, E<UG = 1, Q = 0
- UG ≤ E ≤ OG —> E>OG = 0, E<UG = 0, Q = 1
- E > OG —> E>OG = 1, E<UG = 0, Q = 0

Die Eingänge sind weder doppelbar noch negierbar. Die Ausgänge sind weder doppelbar noch invertierbar.



Zahlenbereich

Integer Wort (16 Bit)

Speziell gilt hier für die nicht negierten Eingänge:

- untere Grenze 8000_{H} -32768
- obere Grenze $7FFF_{\text{H}}$ $+32767$

Allgemein gilt:

- untere Grenze: 8001_{H} -32767
- obere Grenze: $7FFF_{\text{H}}$ $+32767$
- unzulässiger Wert: 8000_{H} —

Der Wert 8000_{H} (-32768) liegt bei der Zweierkomplement-Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

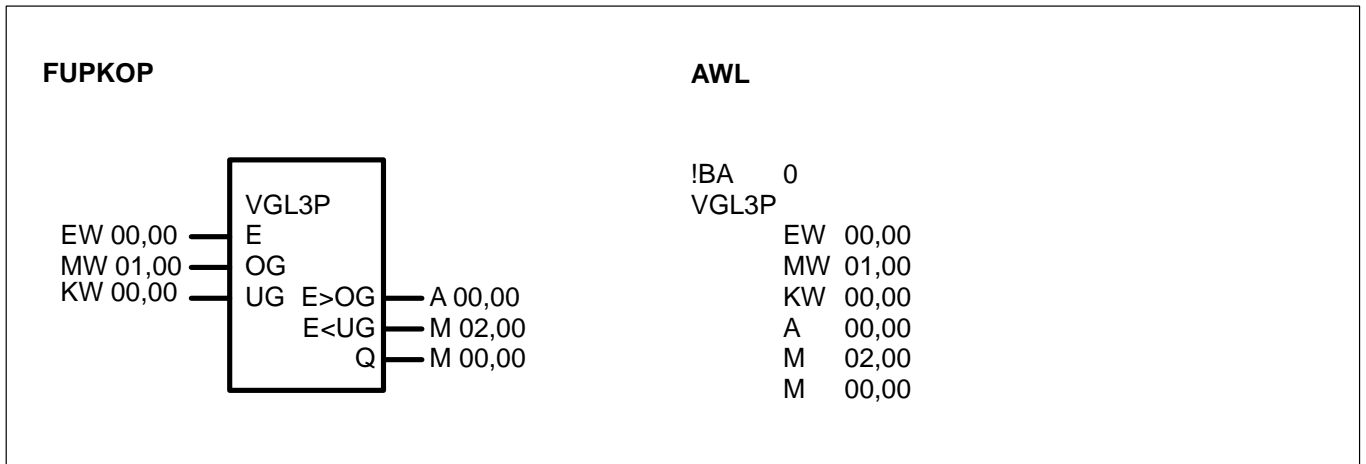
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

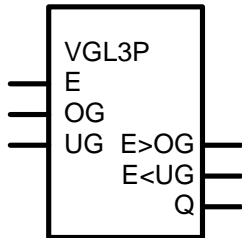
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_{H} (-32768) auf den erlaubten Wert 8001_{H} (-32767) korrigiert.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E	E	W	N	P	J	0	0
OG	E	W	N	P	J	0	0
UG	E	W	N	P	J	0	0
E>OG	A	L	N	P	J	0	0
E<UG	A	L	N	P	J	0	0
Q	A	L	N	P	J	0	0

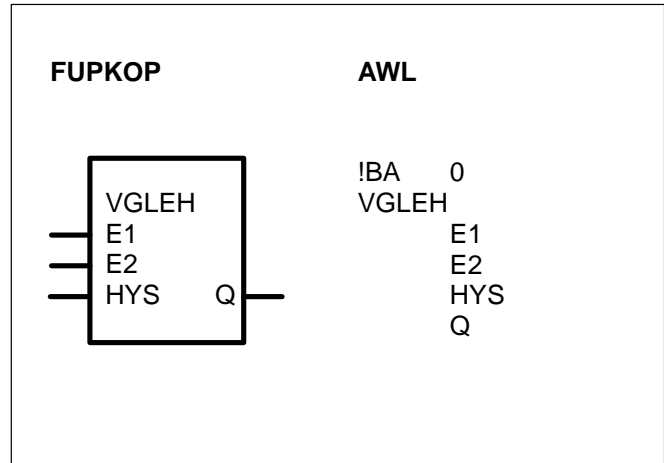
VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr (vorbelegt 0)
00001	VGL3P			
00002		PP 0	E	Eingang WORT(Eingangswert)
00003		PP 0	OG	Eingang WORT(Obere Grenze)
00004		PP 0	UG	Eingang WORT(Untere Grenze)
00005		PP 0	E>OG	Ausgang BINÄR (Wert > Obere Gr.)
00006		PP 0	E<UG	Ausgang BINÄR (Wert < Untere Gr.)
00007		PP 0	Q	Ausgang BINÄR (UG ≤ E ≤ OG)

Die Werte der Operanden an den Eingängen E1 und E2 werden miteinander verglichen. Das Ergebnis wird, unter Berücksichtigung der Hysterese am Eingang HYS, am Ausgang Q signalisiert.

Es gilt:

- $E1 \geq E2 \rightarrow Q = 1$
- $E1 < E2 - HYS \rightarrow Q = 0$
- $E2 - HYS \leq E1 < E2 \rightarrow Q$ wie im vorhergehenden Zyklus



Parameter

E1	WORT	EW, MW, AW, KW	Eingangswert 1
E2	WORT	EW, MW, AW, KW	Eingangswert 2
HYS	WORT	EW, MW, AW, KW	Hysterese
Q	BINÄR	A, M	Ergebnis des Vergleichs

VE-Daten

Laufzeit:

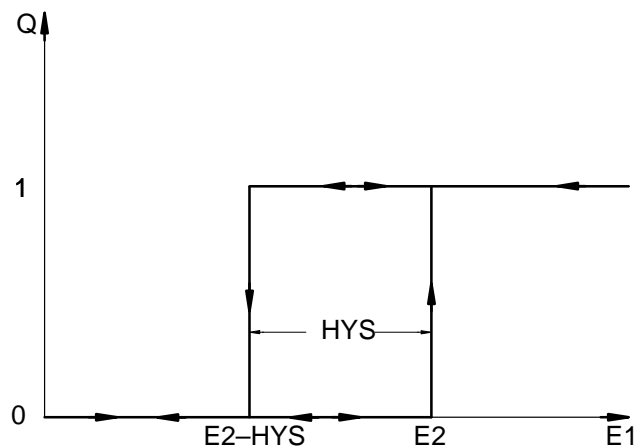
Grundlaufzeit:	56 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Die Werte der Operanden an den Eingängen E1 und E2 werden miteinander verglichen. Das Ergebnis wird, unter Berücksichtigung der Hysterese am Eingang HYS, am Ausgang Q signalisiert.

Es gilt:

- $E1 \geq E2 \rightarrow Q = 1$
- $E1 < E2 - HYS \rightarrow Q = 0$
- $E2 - HYS \leq E1 < E2 \rightarrow Q$ wie im vorhergehenden Zyklus



HYS = Hysterese

Die Eingänge sind weder doppelbar noch negierbar. Der Ausgang ist weder doppelbar noch invertierbar.

Zahlenbereich

Integer Wort (16 Bit)

– untere Grenze:	8001 _H	–32767
– obere Grenze:	7FFF _H	+32767
– unzulässiger Wert:	8000 _H	—

Speziell gilt hier noch für den linken Rand der Hysterese folgende Vorschrift:

$$E2 - HYS \geq -32\,767 \text{ (} 8001_{\text{H}} \text{)}$$

Der Wert 8000_H (–32768) liegt bei der Zweierkomplement–Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

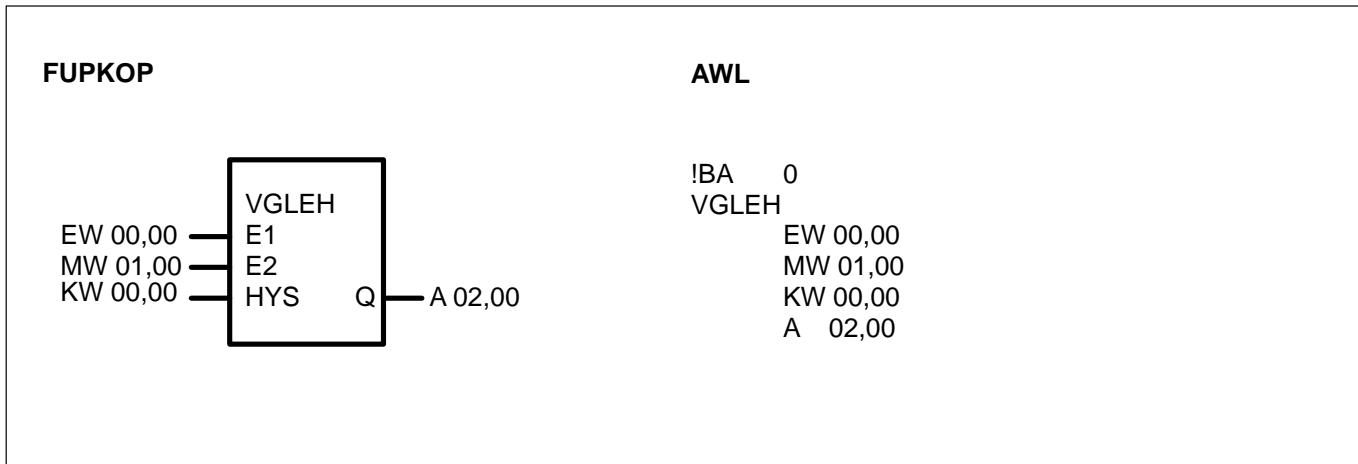
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

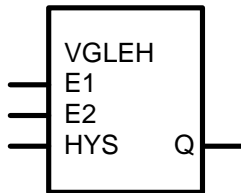
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (–32768) auf den erlaubten Wert 8001_H (–32767) korrigiert.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	J	0	0
E2	E	W	N	P	J	0	0
HYS	E	W	N	P	J	0	0
Q	A	L	N	P	J	0	0

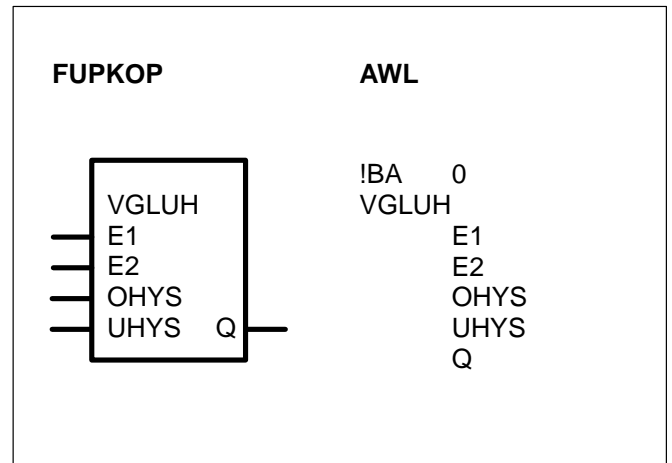
VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	VGLEH			
00002		PP 0	E1	Eingang WORT
00003		PP 0	E2	Eingang WORT
00004		PP 0	HYS	Eingang WORT(Hysterese)
00005		PP 0	Q	Ausgang BINÄR

Die Werte der Operanden an den Eingängen E1 und E2 werden miteinander verglichen. Das Ergebnis wird, unter Berücksichtigung der Hysterese an den Eingängen OHYS (obere Hysterese) und UHYS (untere Hysterese), am Ausgang Q signalisiert.

Es gilt:

- $E1 < E2 - UHYS \rightarrow Q = 0$
- $E1 \geq E2 + OHYS \rightarrow Q = 1$
- $E2 - UHYS \leq E1 \leq E2 + OHYS \rightarrow Q \text{ wie im vorhergehenden Zyklus}$



Parameter

E1	WORT	EW, MW, AW, KW	Eingangswert 1
E2	WORT	EW, MW, AW, KW	Eingangswert 2
OHYS	WORT	EW, MW, AW, KW	obere Hysterese
UHYS	WORT	EW, MW, AW, KW	untere Hyterese
Q	BINÄR	A, M	Ausgang

VE-Daten

Laufzeit:	
Grundlaufzeit:	58 ... 63 µs
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

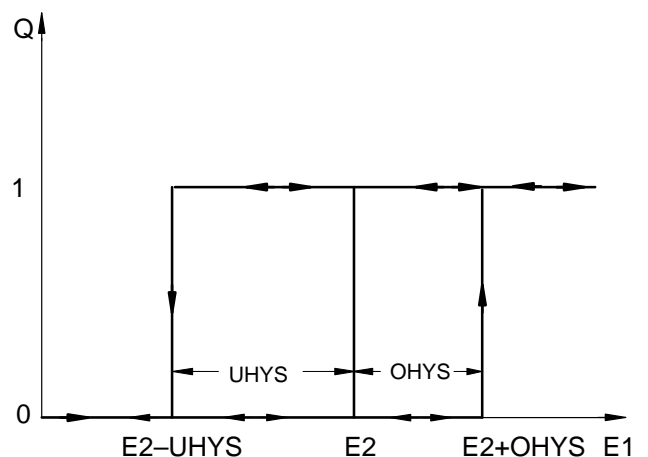
Beschreibung

Die Werte der Operanden an den Eingängen E1 und E2 werden miteinander verglichen. Das Ergebnis wird, unter Berücksichtigung der Hysterese an den Eingängen OHYS (obere Hysterese) und UHYS (untere Hysterese), am Ausgang Q signalisiert.

Es gilt:

- $E1 < E2 - UHYS \rightarrow Q = 0$
- $E1 \geq E2 + OHYS \rightarrow Q = 1$
- $E2 - UHYS \leq E1 \leq E2 + OHYS \rightarrow Q \text{ wie im vorhergehenden Zyklus}$

Die Eingänge sind weder doppelbar noch negierbar. Der Ausgang ist weder doppelbar noch invertierbar.



Zahlenbereich

Integer Wort (16 Bit)

Es gilt hier speziell:

E1 $\geq 8000_{\text{H}}$ (-32768)

E2 – UHYS $\geq 8001_{\text{H}}$ (-32767)

Allgemein gilt:

– untere Grenze: 8001_{H} -32767

– obere Grenze: $7FFF_{\text{H}}$ $+32767$

– unzulässiger Wert: 8000_{H} —

Der Wert 8000_{H} (-32768) liegt bei der Zweierkomplement–Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

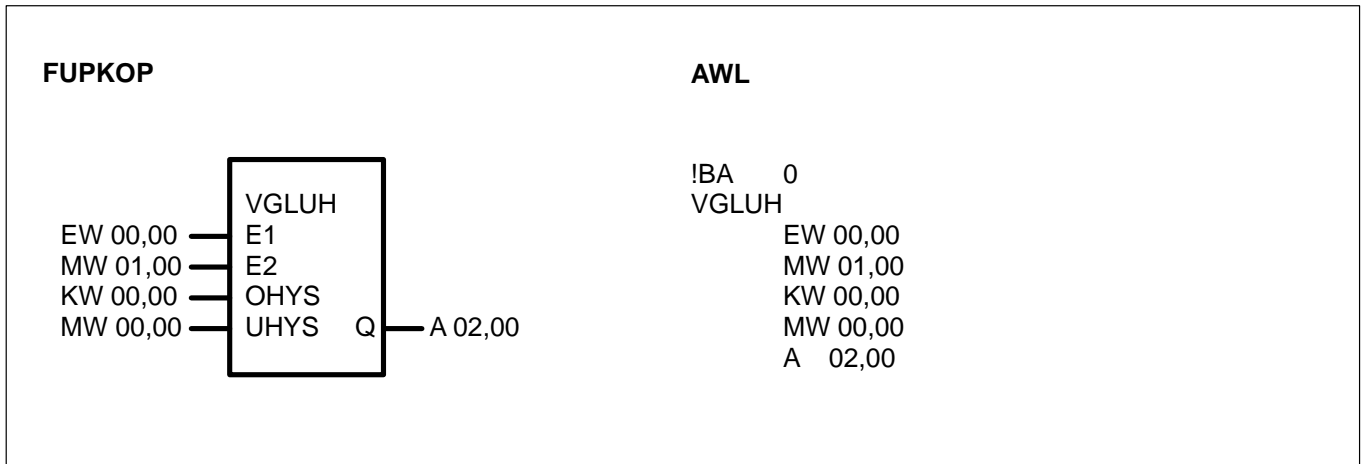
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

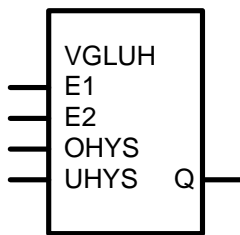
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_{H} (-32768) auf den erlaubten Wert 8001_{H} (-32767) korrigiert.

Beispiel



VE-FUP-Definition

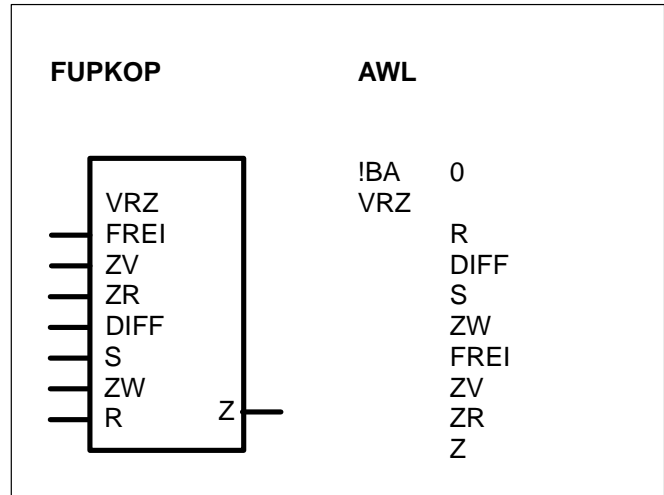


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	J	0	0
E2	E	W	N	P	J	0	0
OHYS	E	W	N	P	J	0	0
UHYS	E	W	N	P	J	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	VGLUH			
00002		PP 0	E1	Eingang WORT
00003		PP 0	E2	Eingang WORT
00004		PP 0	OHYS	Eingang WORT(Obere Hysterese)
00005		PP 0	UHYS	Eingang WORT (Untere Hysterese)
00006		PP 0	Q	Ausgang BINÄR

Der Funktionsbaustein dient zum Zählen von Impulsen. Beim Zählvorgang wird jeweils die positive Flanke des Impulses ausgewertet. Der Zähler kann sowohl vorwärts als auch rückwärts zählen, wobei die Schrittweite pro Zählvorgang vorgegeben werden kann. Die Voreinstellung des Zählerstandes auf einen Zwischenwert ist möglich.



Parameter

FREI	BINÄR	E, M, A, K, S	Freigabe der Bausteinbearbeitung
ZV	BINÄR	E, M, A, K, S	Impulseingang vorwärtszählen
ZR	BINÄR	E, M, A, K, S	Impulseingang rückwärtszählen
DIFF	WORT	EW, MW, AW, KW	Änderung Zählerstand pro positiver Flanke (Schrittweite)
S	BINÄR	E, M, A, K, S	Setzen des Zählers auf einen Zwischenwert
ZW	WORT	EW, MW, AW, KW	Zwischenwert
R	BINÄR	E, M, A, K, S	Rücksetzen des Zählers
Z	WORT	AW, MW	Ausgang für Zählerstand

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	92 µs	241 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja, falls FREI = 1	ja, falls FREI = 1
Anzahl der Vergangenheitswerte:	2 Worte	3 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

Der Funktionsbaustein dient zum Zählen von Impulsen. Beim Zählvorgang wird jeweils die positive Flanke des Impulses ausgewertet. Der Zähler kann sowohl vorwärts als auch rückwärts zählen, wobei die Schrittweite pro Zählvorgang vorgegeben werden kann. Die Voreinstellung des Zählerstandes auf einen Zwischenwert ist möglich.

FREI BINÄR

Mit dem Eingang FREI wird der Zählvorgang freigegeben oder gesperrt.

Es gilt: FREI = 0 → Zählvorgang gesperrt
FREI = 1 → Zählvorgang freigegeben

ZV BINÄR

Jede positive Flanke (0→1-Flanke) am Eingang ZV erhöht den momentanen Zählerstand um die am Eingang DIFF angegebene Schrittweite.

ZR BINÄR

Jede positive Flanke (0→1-Flanke) am Eingang ZR erniedrigt den momentanen Zählerstand um die am Eingang DIFF angegebene Schrittweite.

DIFF WORT

Am Eingang DIFF wird die Schrittweite für den Zählvorgang angegeben. Die Schrittweite ist der Wert, um den der Zähler bei jeder positiven Flanke am Eingang ZV bzw. ZR verändert wird.

S BINÄR

Mit einem 1–Signal am Eingang S wird der Zählerstand auf den am Eingang ZW angegebenen Wert gesetzt. Solange am Eingang S ein 1–Signal anliegt, ist der Zählvorgang blockiert. Der Setzvorgang ist auch wirksam, wenn am Eingang FREI ein 1–Signal anliegt.

ZW WORT

Am Eingang ZW wird der Wert angegeben, auf den der Zählerstand durch ein 1–Signal am Eingang S gesetzt wird.

R BINÄR

Ein 1–Signal am Eingang R setzt den Zählerstand auf den Wert 0. Der Reset–Eingang R hat die höchste Priorität aller Eingänge.

Z WORT

Am Ausgang Z steht der aktuelle Zählerstand zur Verfügung.

Erreicht der Zählerstand die positive bzw. negative Grenze des Zahlenbereichs, so wird der Zählerstand auf diesen Wert begrenzt.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar/invertierbar.

Zahlenbereich

Integer Wort (16 Bit)

- untere Grenze: 8001_H –32767
- obere Grenze: 7FFF_H +32767
- unzulässiger Wert: 8000_H —

Der Wert 8000_H (–32768) liegt bei der Zweierkomplement–Arithmetik *außerhalb* des Zahlenbereichs und wird von der SPS weder erzeugt noch korrekt verarbeitet. Falls dieser *verbotene* Wert

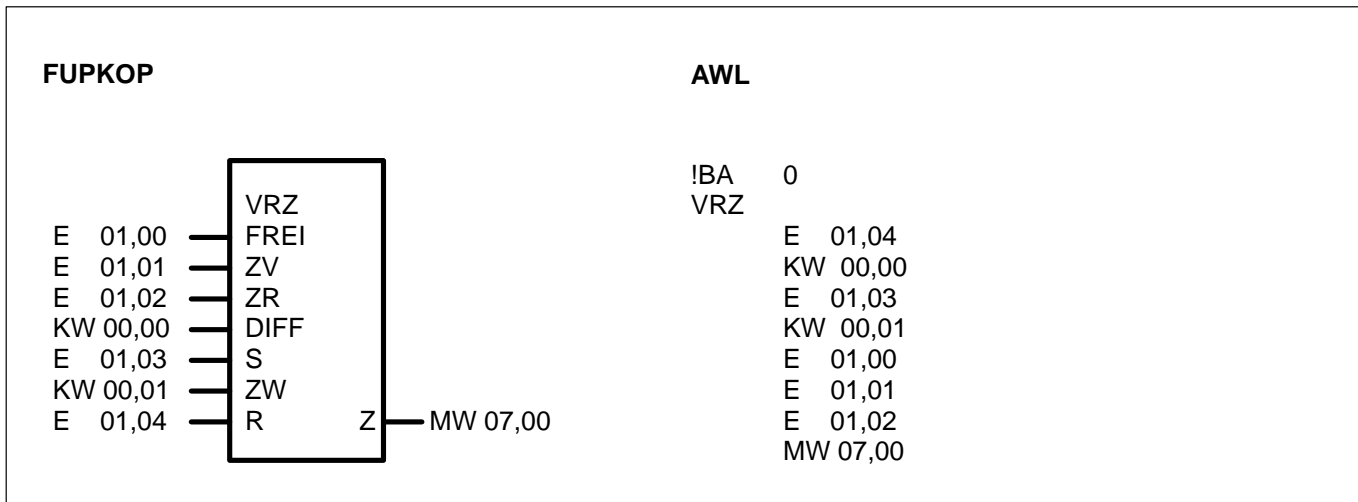
- durch Bitmanipulationen des Anwenders oder
- durch Einlesen von außerhalb der SPS oder
- durch eine indirekte Wortkonstante

in die SPS gelangt, darf auf diesen Wert auf *keinen* Fall eine Negation oder Subtraktion angewendet werden.

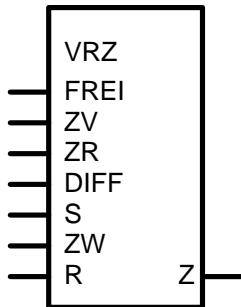
Durch eine Zuweisung (=), Addition (+), Multiplikation (*) oder Division (:) wird wieder ein zulässiger Wert erzeugt.

Bei der Zuweisung (=) wird der *verbotene* Wert 8000_H (–32768) auf den erlaubten Wert 8001_H (–32767) korrigiert.

Beispiel



VE-FUP-Definition

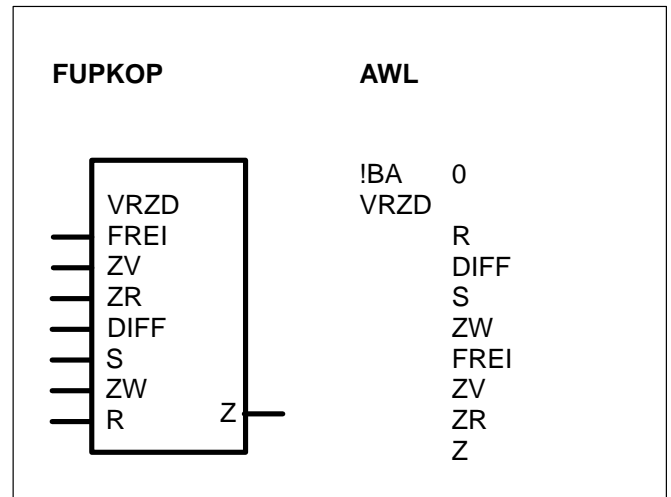


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
ZV	E	L	N	P	J	0	0
ZR	E	L	N	P	J	0	0
DIFF	E	W	N	P	J	0	0
S	E	L	N	P	J	0	0
ZW	E	W	N	P	J	0	0
R	E	L	N	P	J	0	0
Z	A	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr (vorbelegt 0)
00001	VRZ			
00002		PP 0	R	Eingang BINÄR (Reset Zähler = 0)
00003		PP 0	DIFF	Eingang WORT (Änderung je Flanke)
00004		PP 0	S	Eingang BINÄR (Setze Zähler =ZW)
00005		PP 0	ZW	Eingang WORT (Initialwert)
00006		PP 0	FREI	Eingang BINÄR (Baustein-Freigabe)
00007		PP 0	ZV	Eingang BINÄR (Zähleingang vorw.)
00008		PP 0	ZR	Eingang BINÄR (Zähleingang rückw.)
00009		PP 0	Z	Ausgang WORT (Zählerstand)

Der Funktionsbaustein dient zum Zählen von Impulsen. Beim Zählvorgang wird jeweils die positive Flanke des Impulses ausgewertet. Der Zähler kann sowohl vorwärts als auch rückwärts zählen, wobei die Schrittweite pro Zählvorgang vorgegeben werden kann. Die Voreinstellung des Zählerstandes auf einen Zwischenwert ist möglich.



Parameter

FREI	BINÄR	E, M, A, K, S	Freigabe der Bausteinbearbeitung
ZV	BINÄR	E, M, A, K, S	Impulseingang vorwärtszählen
ZR	BINÄR	E, M, A, K, S	Impulseingang rückwärtszählen
DIFF	DOPPELWORT	MD, KD	Änderung Zählerstand pro positiver Flanke (Schrittweite)
S	BINÄR	E, M, A, K, S	Setzen des Zählers auf einen Zwischenwert
ZW	DOPPELWORT	MD, KD	Zwischenwert
R	BINÄR	E, M, A, K, S	Rücksetzen des Zählers
Z	DOPPELWORT	MD	Ausgang für Zählerstand

VE-Daten

Laufzeit:	
Grundlaufzeit:	38 ... 113 µs (je nach Betriebsart)
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja, falls FREI = 1
Anzahl der Vergangenheitswerte:	3 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein dient zum Zählen von Impulsen. Beim Zählvorgang wird jeweils die positive Flanke des Impulses ausgewertet. Der Zähler kann sowohl vorwärts als auch rückwärts zählen, wobei die Schrittweite pro Zählvorgang vorgegeben werden kann. Die Voreinstellung des Zählerstandes auf einen Zwischenwert ist möglich.

FREI BINÄR

Mit dem Eingang FREI wird der Zählvorgang freigegeben oder gesperrt.

Es gilt: FREI = 0 → Zählvorgang gesperrt
 FREI = 1 → Zählvorgang freigegeben

ZV BINÄR

Jede positive Flanke (0→1-Flanke) am Eingang ZV erhöht den momentanen Zählerstand um die am Eingang DIFF angegebene Schrittweite.

ZR BINÄR

Jede positive Flanke (0→1-Flanke) am Eingang ZR erniedrigt den momentanen Zählerstand um die am Eingang DIFF angegebene Schrittweite.

DIFF DOPPELWORT

Am Eingang DIFF wird die Schrittweite für den Zählvorgang angegeben. Die Schrittweite ist der Wert, um den der Zähler bei jeder positiven Flanke am Eingang ZV bzw. ZR verändert wird.

S BINÄR

Mit einem 1–Signal am Eingang S wird der Zählerstand auf den am Eingang ZW angegebenen Wert gesetzt. Solange am Eingang S ein 1–Signal anliegt, ist der Zählvorgang blockiert. Der Setzvorgang ist auch wirksam, wenn am Eingang FREI ein 1–Signal anliegt.

ZW DOPPELWORT

Am Eingang ZW wird der Wert angegeben, auf den der Zählerstand durch ein 1–Signal am Eingang S gesetzt wird.

R BINÄR

Ein 1–Signal am Eingang R setzt den Zählerstand auf den Wert 0. Der Reset–Eingang R hat die höchste Priorität aller Eingänge.

Z DOPPELWORT

Am Ausgang Z steht der aktuelle Zählerstand zur Verfügung.

Erreicht der Zählerstand die positive bzw. negative Grenze des Zahlenbereichs, so wird der Zählerstand auf diesen Wert begrenzt.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar/invertierbar.

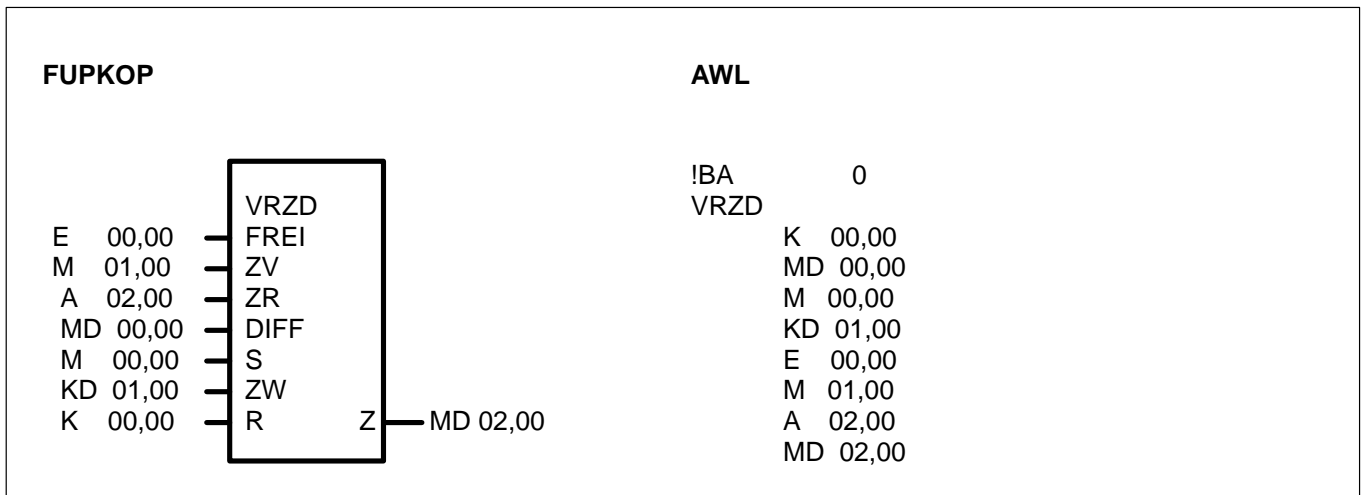
Zahlenbereich

Integer Doppelwort (32 Bit)

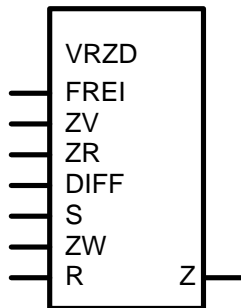
– untere Grenze:	8000 0001 _H	–2 147 483 647
– obere Grenze:	7FFF FFFF _H	+2 147 483 647
– unzulässiger Wert:	8000 0000 _H	—

Alle Bausteine für die Doppelwort–Arithmetik prüfen die zu bearbeitenden Werte auf Zulässigkeit. Tritt der unzulässige Wert auf, so wird er auf den zulässigen Wert 8000 0001_H (–2 147 483 647) korrigiert.

Beispiel



VE-FUP-Definition



Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
ZV	E	L	N	P	J	0	0
ZR	E	L	N	P	J	0	0
DIFF	E	D	N	P	J	0	0
S	E	L	N	P	J	0	0
ZW	E	D	N	P	J	0	0
R	E	L	N	P	J	0	0
Z	A	D	N	P	J	0	0

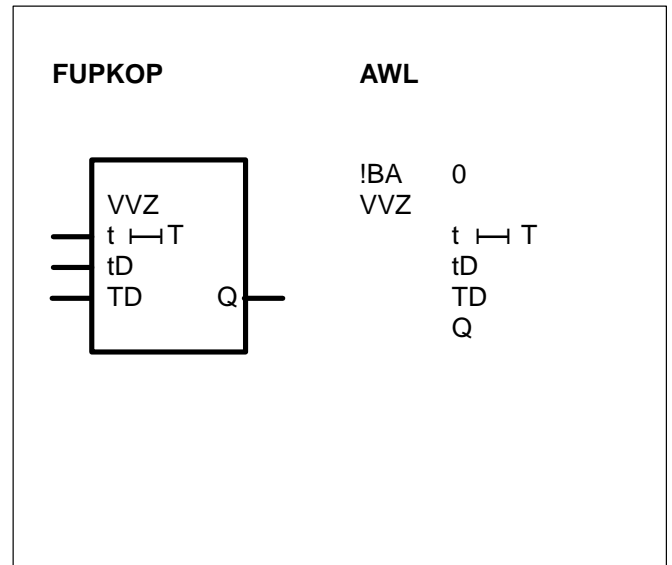
VE-AWL-Definition

Nr	Param.	Param. Gruppe	Param. Typ	Inv.	Baustein-Nr. (vorbelegt 0)
00000	!BA		0	Nr	Baustein-Nr. (vorbelegt 0)
00001	VRZD				
00002		PP	0	R	Eingang BINÄR (Reset Zähler = 0)
00003		PP	0	DIFF	Eingang DOPPELWORT (Änderung/Puls)
00004		PP	0	S	Eingang BINÄR (Setze Zähler =ZW)
00005		PP	0	ZW	Eingang DOPPELWORT (Initialwert)
00006		PP	0	FREI	Eingang BINÄR (Baustein-Freigabe)
00007		PP	0	ZV	Eingang BINÄR (Zähleingang vorw.)
00008		PP	0	ZR	Eingang BINÄR (Zähleingang rückw.)
00009		PP	0	Z	Ausgang DOPPELWORT (Zählerstand)

Eine 0–1–Flanke am Eingang t_{HT} wird um die Zeitdauer t_D verzögert am Ausgang Q ausgegeben. Eine nachfolgende 1–0–Flanke am Eingang t_{HT} wird um die Zeitdauer T_D verzögert am Ausgang Q ausgegeben.

Geht der Eingang t_{HT} vor Ablauf der Zeit t_D wieder auf 0–Pegel, so bleibt der Ausgang Q auf 0–Pegel.

Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit



Parameter

t_{HT}	BINÄR	E, M, A, S, K	Eingangssignal
t_D	DOPPELWORT	MD, KD	Verzögerungszeit 0–1–Flanke
T_D	DOPPELWORT	MD, KD	Verzögerungszeit 1–0–Flanke
Q	BINÄR	M, A	Ausgangssignal

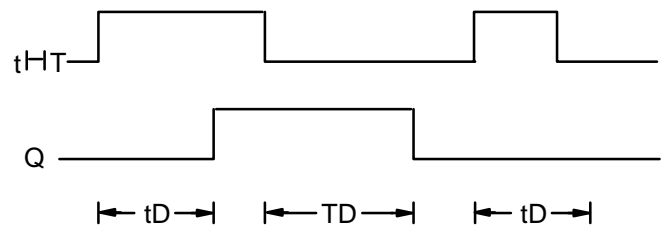
VE-DATEN

Laufzeit:	
Grundlaufzeit:	46 μ s, 0–1–Flanke 566 μ s, 1–0–Flanke 283 μ s
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenhheitswerte:	2 Worte
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Eine 0–1–Flanke am Eingang t_{HT} wird um die Zeitdauer t_D verzögert am Ausgang Q ausgegeben. Eine nachfolgende 1–0–Flanke am Eingang t_{HT} wird um die Zeitdauer T_D verzögert am Ausgang Q ausgegeben.

Geht der Eingang t_{HT} vor Ablauf der Zeit t_D wieder auf 0–Pegel, so bleibt der Ausgang Q auf 0–Pegel.



Die Zeitangabe erfolgt in Millisekunden, wobei nur ganzzahlige Vielfache von 5 ms zulässig sind (Bsp.: 5 ms, 10 ms, 500 ms, 100 000 ms ...). Angebbbarer Zeitbereich: 5 ms ... 24,8 Tage.

Bei den indirekten Konstanten KD wird der Wert der Verzögerungszeit im Langtext angegeben.

Maximaler Zeitversatz am Ausgang: < 1 Zykluszeit

Sinnvoller Bereich für t_D und T_D : > 1 Zykluszeit

Die Eingänge und der Ausgang sind weder doppelbar noch invertierbar.

Allgemeines Verhalten

- Gestartete Zeitwerke werden vom Betriebssystem der SPS bearbeitet und sind deshalb vollkommen unabhängig von der Bearbeitung des SPS-Programms. Erst nach Ablauf des Zeitwerks erfolgt eine entsprechende Meldung des Betriebssystems an den zugehörigen Zeitbaustein im SPS-Programm.
- Die Bearbeitung eines Zeitwerks im Betriebssystem der SPS wird durch folgende Befehle *nicht* beeinflusst:
 - Programm abbrechen
 - Programm starten
 - Programm stoppen
 - Programm fortsetzen

D.h. ist ein Zeitwerk gestartet, so wird es im Betriebssystem der SPS auch dann weiter bearbeitet, wenn das zugehörige SPS-Programm abgebrochen, wieder gestartet oder gestoppt und wieder fortgesetzt wird.

Initialisierung

Die Zeitwerke werden grundsätzlich bei jedem Kalt- oder Warmstart der SPS initialisiert. Ein laufendes Zeitwerk wird also stets durch einen Kalt- oder Warmstart der SPS abgebrochen.

Kaltstart:

- Befehl KALT <CR>
- erstmaliges Zuschalten der Spannung

Warmstart:

- Befehl WARM <CR>
- Zuschalten der Spannung
- RESET-Schalter

Wichtiger Hinweis

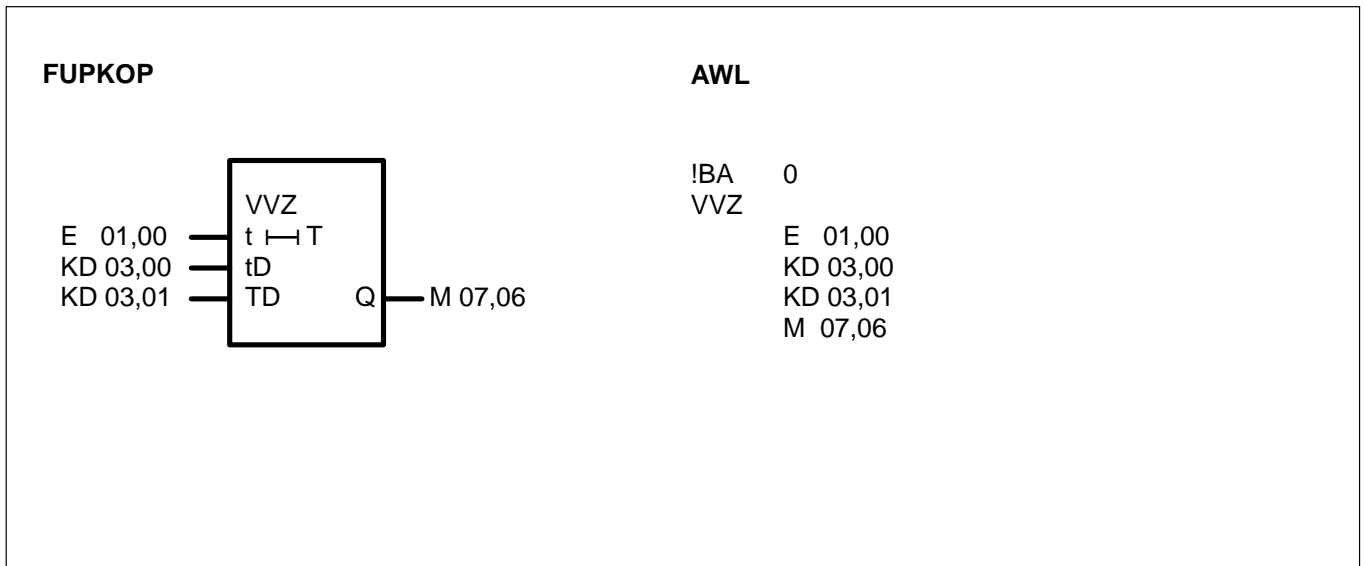
Das Verhalten der Zeitwerke ist geändert worden. Alle laufenden Zeitwerke werden bei folgenden Aktionen automatisch gestoppt:

- SPS-Programm abbrechen
- RUN/STOP-Schalter von RUN -> STOP

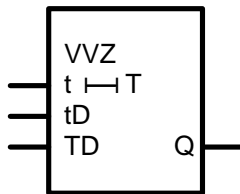
Diese Änderung gilt ab:

07 KR 91 Index g
07 KT 92 Index g
07 KT 93

Beispiel



VE-FUP-Definition

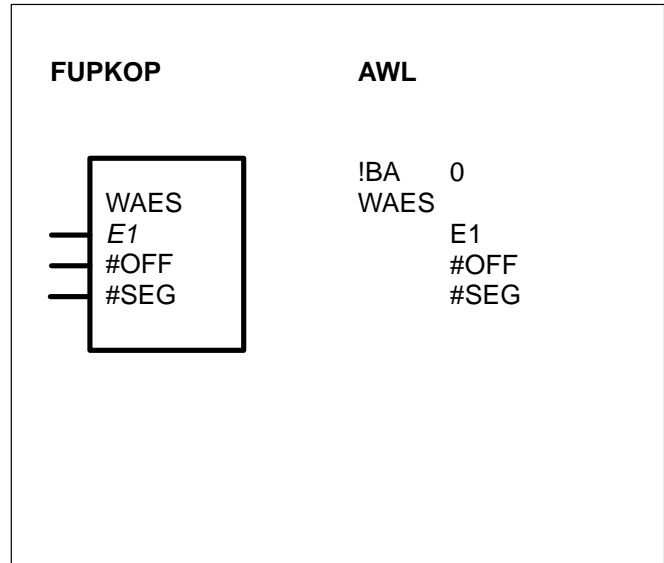


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
t \dashv T	E	L	N	P	J	0	0
tD	E	D	N	P	J	0	0
TD	E	D	N	P	J	0	0
Q	A	L	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	VVZ			
00002		PP 0	t \dashv T	Eingang BINÄR (Starte Zeit)
00003		PP 0	tD	Konstante DOPPELWORT (Zeitwert)
00004		PP 0	TD	Konstante DOPPELWORT (Zeitwert)
00005		PP 0	Q	Ausgang BINÄR (Zeit)

Ändert sich der Wert des Operanden am Eingang E1 verglichen mit dem Wert bei der vorhergehenden Bearbeitung des Bausteins, so wird der Wert des Operanden am Eingang E1 auf die angegebene physikalische Adresse geschrieben.



Parameter

E1	WORT	EW, MW, AW, KW
#OFF	DIREKTE KONSTANTE	#, #H
#SEG	DIREKTE KONSTANTE	#, #H

Eingang für den zu lesenden Operanden
 Offset-Adresse des Speicherplatzes, auf den der Wert von E1 bei Änderung zu schreiben ist.
 Segment-Adresse des Speicherplatzes, auf den der Wert von E1 bei Änderung zu schreiben ist.

VE-DATEN

Laufzeit:	
Grundlaufzeit:	23 µs, kein Lesen; 33 µs, Lesen
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	1 Wort
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Ändert sich der Wert des Operanden am Eingang E1 verglichen mit dem Wert bei der vorhergehenden Bearbeitung des Bausteins, so wird der Wert des Operanden am Eingang E1 auf die angegebene physikalische Adresse geschrieben.

Die physikalische Adresse besteht aus Segment und Offset.

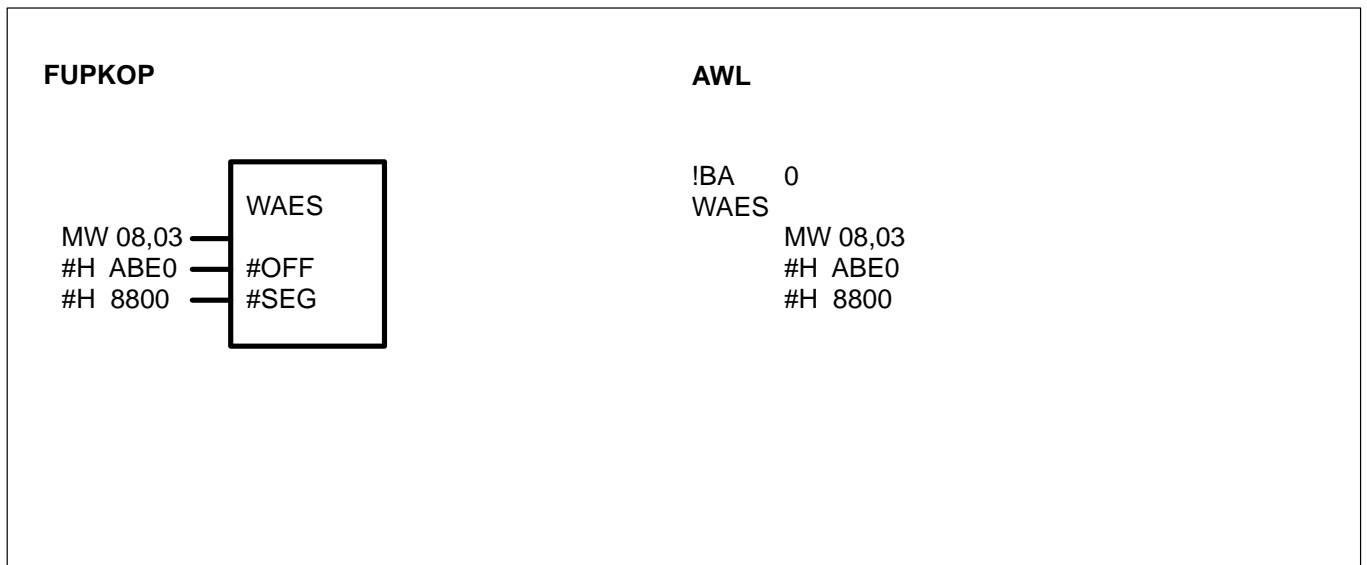
Die Eingänge sind weder doppelbar noch negierbar.

E1 WORT
 Bei Änderung des Operanden am Eingang E1 wird dessen Wert auf die Adresse, die an den Eingängen #OFF und #SEG angegeben ist, geschrieben.

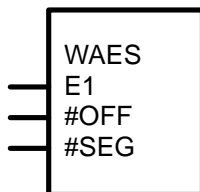
#OFF DIREKTE KONSTANTE (#, #H)
 Am Eingang #OFF wird der Offset-Anteil der zu schreibenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

#SEG DIREKTE KONSTANTE (#, #H)
 Am Eingang #SEG wird der Segment-Anteil der zu schreibenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

Beispiel



VE-FUP-Definition

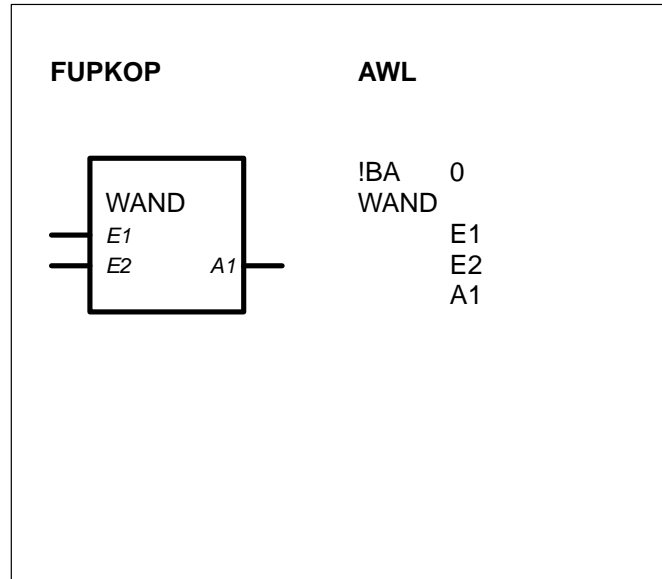


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
#OFF	K	W	N	P	J	0	0
#SEG	K	W	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	WAES			
00002		PP 0	E1	Eingang WORT (Wert)
00003		PP 0	#OFF	# KONSTANTE (Offset-Adresse)
00004		PP 0	#SEG	# KONSTANTE (Segment-Adresse)

Der Funktionsbaustein bildet die bitweise UND-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	WORT	EW, MW, AW, KW	Operand 1
E2	WORT	EW, MW, AW, KW	Operand 2
A1	WORT	MW, AW	Ergebnis der UND-Verknüpfung

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:

Grundlaufzeit:

Zusätzliche Laufzeit:

Aktualisierung der Ausgänge:

Anzahl der Vergangenheitswerte:

Verfügbar ab:

28 µs

—

ja

keine

ABB Procontic CS31 / 907 PC 331

46 µs

—

ja

keine

907 PC 331 / 07 KR 31 V 1.0,
07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

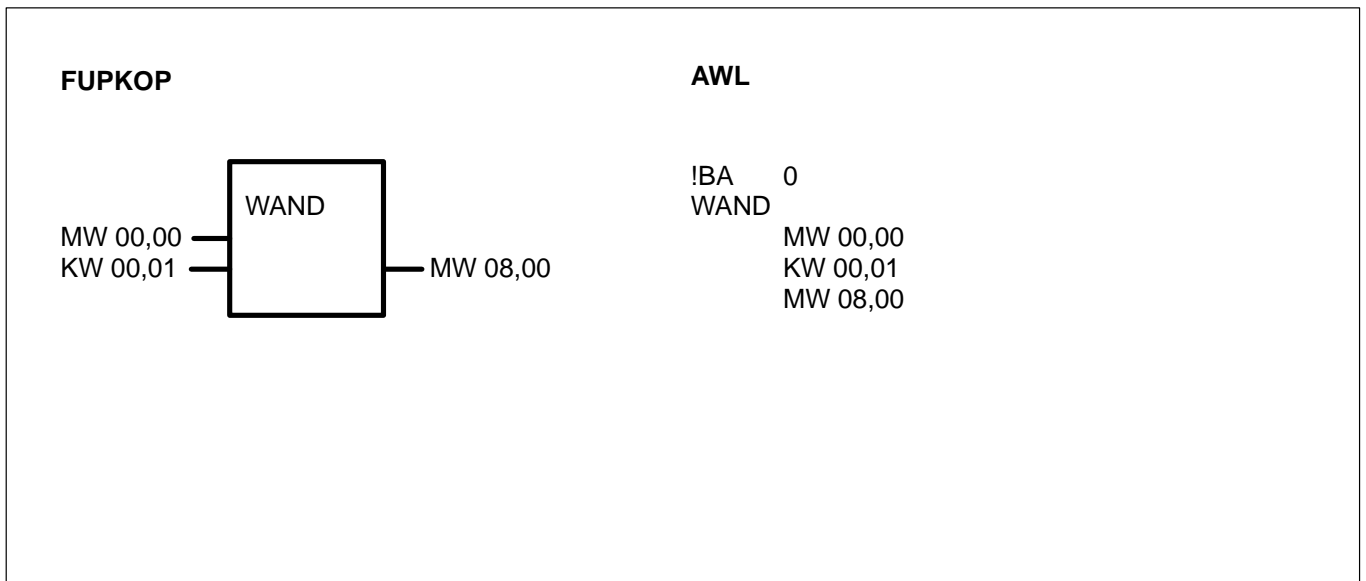
Der Funktionsbaustein bildet die bitweise UND-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar noch invertierbar.

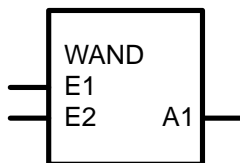
Beispiel:

E1	0.0.0.0	0.0.1.1	0.0.1.0	0.1.1.0
E2	1.0.0.1	0.0.0.0	0.0.1.0	1.1.1.1
A1	0.0.0.0	0.0.0.0	0.0.1.0	0.1.1.0

Beispiel



VE-FUP-Definition

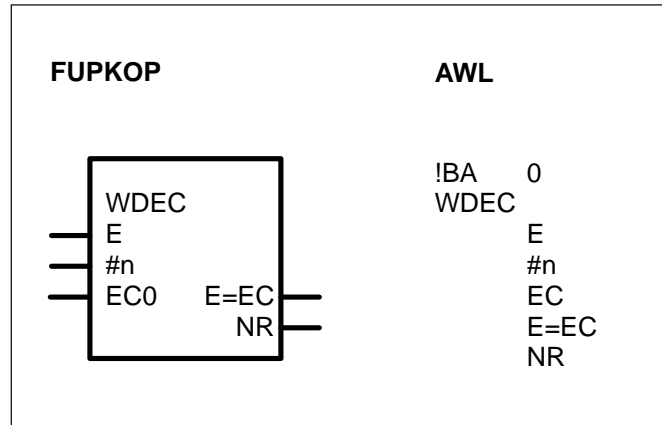


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
E2	E	W	N	P	N	0	0
A1	A	W	N	P	N	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	WAND			
00002		PP 0	E1	Eingang WORT
00003		PP 0	E2	Eingang WORT
00004		PP 0	A1	Ausgang WORT

Der Funktionsbaustein vergleicht den Wert des Operanden am Eingang E mit den Referenzwerten der Operanden an den Eingängen EC0 ... ECn-1. Das Ergebnis des Vergleichs wird an den Ausgängen signalisiert.



Parameter

E	WORT	EW, AW, MW, KW	Eingang
#n	DIREKTE KONSTANTE	#, #H	Anzahl der Referenzwerte
EC0	WORT	EW, AW, MW, KW	Referenzwert; Eingang ist doppelbar
E=EC	BINÄR	A, M	Koinzidenzanzeige
NR	WORT	AW, MW	Nummer des Referenzwertes bei Koinzidenz

VE-Daten

Laufzeit:	
Grundlaufzeit:	22 µs
Zusätzliche Laufzeit:	4 µs (max), pro eingegebenem Referenzwert
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenhheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein vergleicht den Wert des Operanden am Eingang E mit den Referenzwerten der Operanden an den Eingängen EC0 ... ECn-1. Das Ergebnis des Vergleichs wird an den Ausgängen signalisiert.

Stimmt der Eingang E mit mindestens einem der n Referenzwerte EC überein, so wird der Ausgang E=EC auf 1 gesetzt. Dem Operanden am Ausgang NR wird die Nummer des ersten Referenzwertes EC zugewiesen, der mit dem Eingang E übereinstimmt. Die Nummer kann also den Wert 1 bis n annehmen.

Wird keine Übereinstimmung des Eingangswertes E mit einem Referenzwert EC festgestellt, so werden die Ausgänge E=EC und NR auf 0 gesetzt.

Der Eingang EC (Eingangscode) ist doppelbar. Die Anzahl der Eingänge EC ist am Eingang #n als direkte Konstante anzugeben.

E WORT
Am Eingang E wird der Operand angegeben, dessen Wert mit den Referenzwerten an den Eingängen EC0 ... ECn-1 verglichen werden soll.

#n DIREKTE KONSTANTE (#, #H)
Am Eingang #n wird die Anzahl n der projektierten Referenzwerte angegeben. Die Angabe erfolgt als direkte Konstante.

EC0 ... ECn-1 WORT
Der Eingang EC0 ist entsprechend der benötigten Anzahl an Referenzwerten zu doppelbar. An den Eingängen EC0 ... ECn-1 werden die Operanden für die Referenzwerte angegeben. Der Wert des Operanden am Eingang E wird mit diesen Referenzwerten verglichen.

E=EC BINÄR
Am Ausgang E=EC wird die Übereinstimmung (Koinzidenz) des Operandenwertes des Eingangs E mit einem der Referenzwerte signalisiert.
Es gilt: E=EC = 0 -> keine Übereinstimmung
 E=EC = 1 -> Übereinstimmung

NR **WORT**

Am Ausgang NR wird die Nummer des Referenzwertes signalisiert, der mit dem Wert des Operanden am Eingang E übereinstimmt.

Es gilt:

keine Übereinstimmung \rightarrow NR = 0

Übereinstimmung vorhanden \rightarrow NR = n mit $n \geq 1$

E = EC0 \rightarrow NR = 1

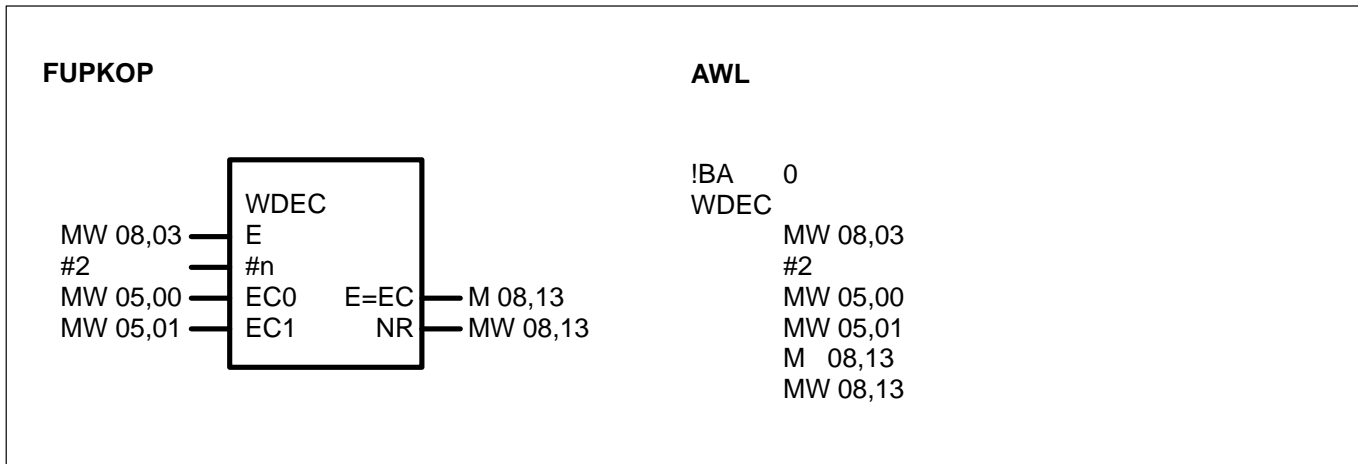
E = EC1 \rightarrow NR = 2

· · ·

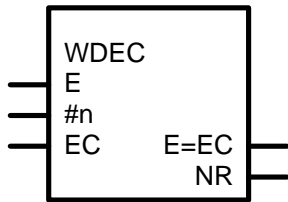
· · ·

E = EC_{n-1} \rightarrow NR = n

Beispiel



VE-FUP-Definition

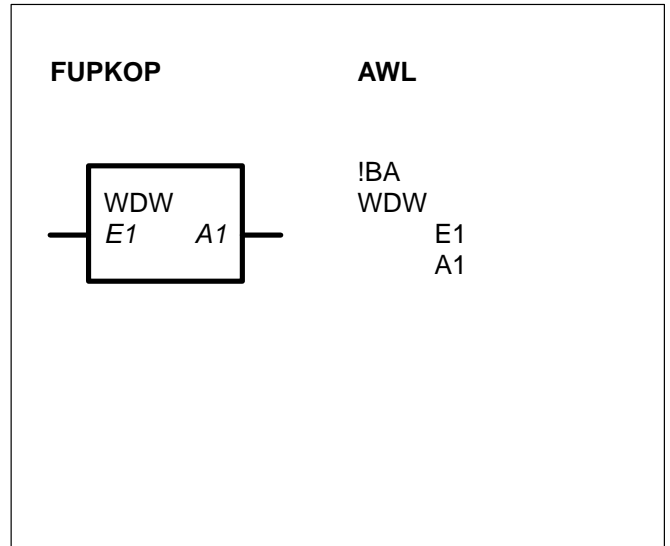


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E	E	W	N	P	J	0	0
#n	K	W	N	P	J	0	0
EC	E	W	N	P	J	1	0
E=EC	A	L	N	P	J	0	0
NR	A	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	WDEC			
00002		PP 0	E	Eingang WORT (Eingangswert)
00003		PP 0	#n	# KONSTANTE (Anzahl EC's)
	[1		
00004		PP 1	EC	Eingang WORT (Eingangscodex)
]	1		
00005		PP 0	E=EC	Ausgang BINÄR (Eing.wert=Ecode)
00006		PP 0	NR	Ausgang WORT (Nummer)

Der Wert des Wort-Operanden am Eingang E1 wird in eine Doppelwortgröße gewandelt, und das Ergebnis dem Doppelwort-Operanden am Ausgang A1 zugewiesen.



Parameter

E1	WORT	EW, AW, MW, KW	zu wandelnde Wortgröße
A1	DOPPELWORT	MD	Ergebnis der Wandlung, Doppelwortgröße

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	26 µs	51 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenhheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 2.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

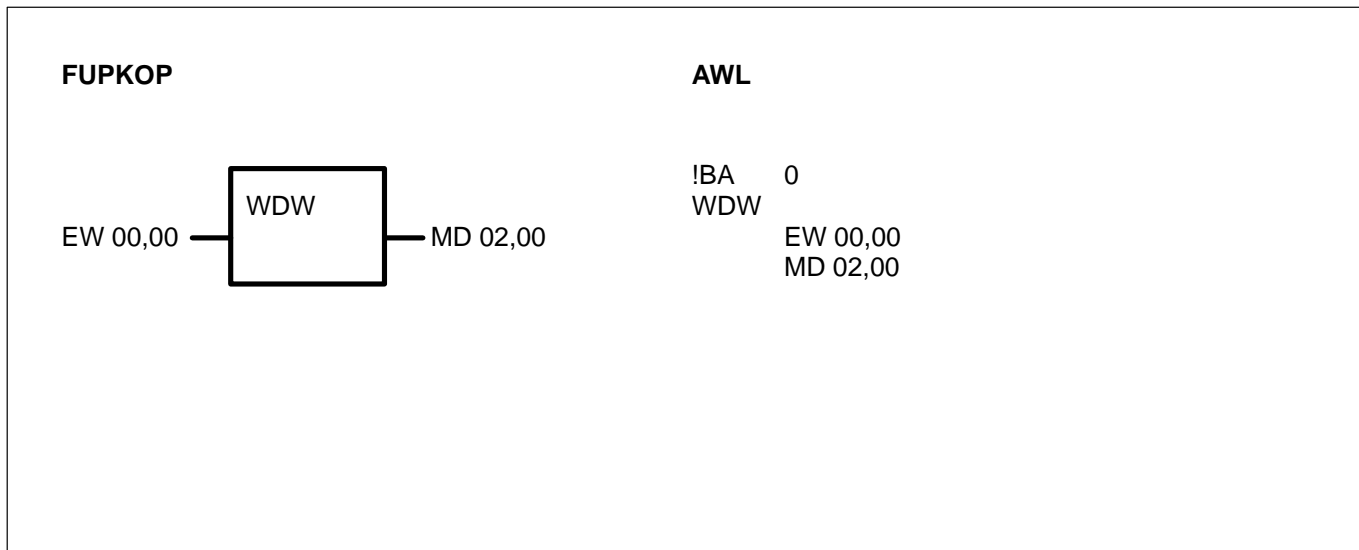
Beschreibung

Der Wert des Wort-Operanden am Eingang E1 wird in eine Doppelwortgröße gewandelt, und das Ergebnis dem Doppelwort-Operanden am Ausgang A1 zugewiesen.

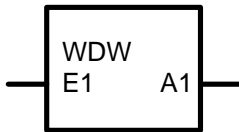
Wertebereich für E1:
 $8000_H \leq E1 \leq 7FFF_H$
 $-32768 \leq E1 \leq 32768$

Der Eingang und der Ausgang sind weder doppelbar noch negierbar.

Beispiel



VE-FUP-Definition

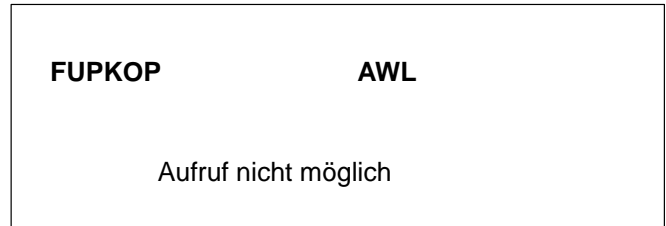


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
A1	A	D	N	P	N	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	WDW			
00002		PP 0	E1	Eingang WORT
00003		PP 0	A1	Ausgang DOPPELWORT

WENN DANN ist eine Grundfunktion in der Hersteller– Bibliothek. Der Name des Elementes in der Bibliothek ist := Das Element kann nicht im VE– und VE–AWL–Editor aufgerufen werden.



Parameter

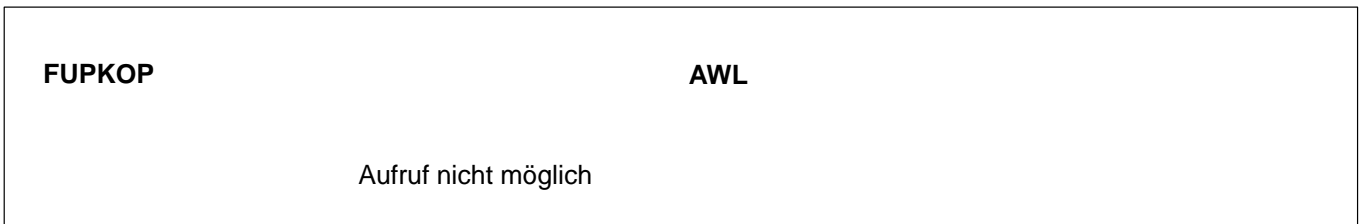
VE–Daten

- Laufzeit:
 - Grundlaufzeit: 4 µs
 - Zusätzliche Laufzeit: —
- Aktualisierung der Ausgänge: —
- Anzahl der Vergangenheitswerte: —
- Verfügbar ab: ABB Procontic CS31 / 907 PC 331

Beschreibung

WENN DANN ist eine Grundfunktion in der Hersteller– Bibliothek. Der Name des Elementes in der Bibliothek ist := Das Element kann nicht im VE– und VE–AWL–Editor aufgerufen werden.
 Das Programmiergerät benutzt dieses Element zur Realisierung von Verbindungslinien.

Beispiel



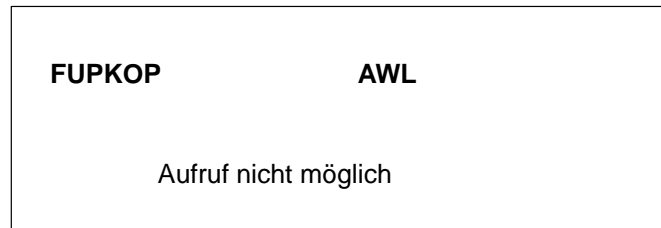
VE–FUP–Definition

nicht definiert

VE–AWL–Definition

00000 ! PP 0 inp
 00002 = PP 0 outp

WENN DANN WORT ist eine Grundfunktion in der Hersteller-Bibliothek. Der Name des Elementes in der Bibliothek ist :=WO
 Das Element kann nicht im VE- und VE-AWL-Editor aufgerufen werden.



Parameter

—

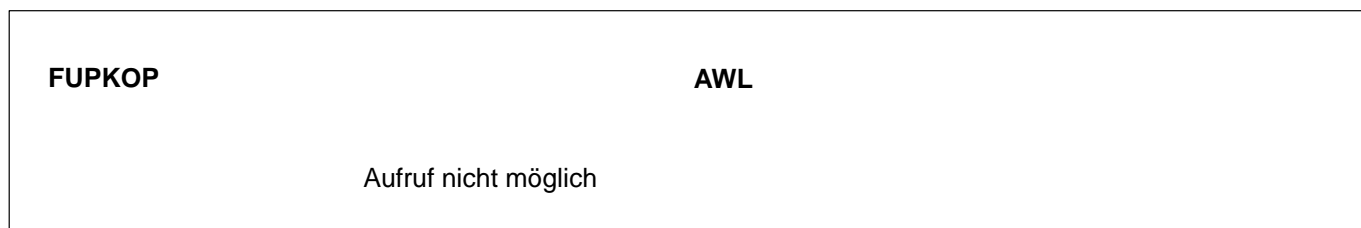
VE-Daten

Laufzeit:
 Grundlaufzeit: 4 µs
 Zusätzliche Laufzeit: —
 Aktualisierung der Ausgänge: —
 Anzahl der Vergangenheitswerte: —
 Verfügbar ab: ABB Procontic CS31 / 907 PC 331

Beschreibung

WENN DANN WORT ist eine Grundfunktion in der Hersteller- Bibliothek. Der Name des Elementes in der Bibliothek ist :=WO
 Das Element kann nicht im VE- und VE-AWL-Editor aufgerufen werden.
 Das Programmiergerät benutzt dieses Element zur Realisierung von Verbindungslinien.

Beispiel



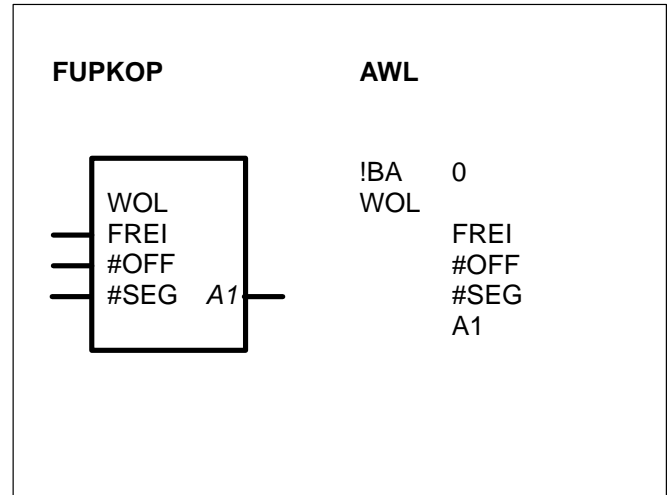
VE-FUP-Definition

nicht definiert

VE-AWL-Definition

00000	!	PP 0	inp
00002	=	PP 0	outp

1–Signal am Eingang FREI bewirkt, daß der Wert der angegebenen physikalischen Adresse gelesen und dem Operanden am Ausgang A1 zugewiesen wird.



Parameter

FREI	BINÄR	E, M, A, S, K	Freigabe des Bausteins
#OFF	DIREKTE KONSTANTE	#, #H	Offset–Adresse des Speicherplatzes, dessen Wert zu lesen ist
#SEG	DIREKTE KONSTANTE	#, #H	Segment–Adresse des Speicherplatzes, dessen Wert zu lesen ist
A1	WORT	AW, MW	Ausgang, dem der gelesene Wert zugewiesen wird.

VE–Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	20 µs, kein Lesen; 33 µs, Lesen	48 µs, Lesen
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 2.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

1–Signal am Eingang FREI bewirkt, daß der Wert der angegebenen physikalischen Adresse gelesen und dem Operanden am Ausgang A1 zugewiesen wird.

Bei 0–Signal am Eingang FREI erfolgt kein Lesen und keine Zuweisung.

Die physikalische Adresse besteht aus Segment und Offset. Der damit erreichbare Adressraum beträgt 1 MByte.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar.

FREI BINÄR

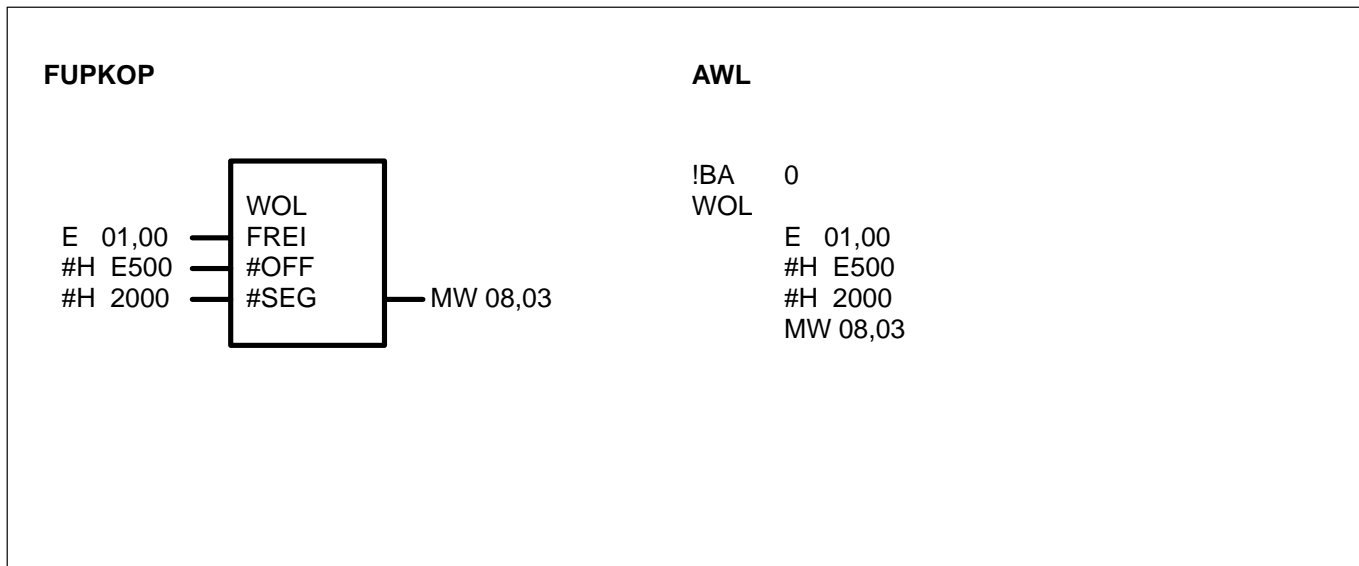
Mit dem Operanden am Eingang FREI wird die Bearbeitung des Bausteins freigegeben oder gesperrt. Es gilt:
 FREI = 0 → Bearbeitung gesperrt
 FREI = 1 → Bearbeitung freigegeben

#OFF DIREKTE KONSTANTE (##H)
 Am Eingang #OFF wird der Offset–Anteil der zu lesenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

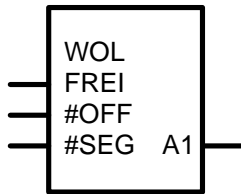
#SEG DIREKTE KONSTANTE (##H)
 Am Eingang #SEG wird der Segment–Anteil der zu lesenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

A1 WORT
 Dem Operanden am Ausgang A1 wird der gelesene Wert zugewiesen.

Beispiel



VE-FUP-Definition

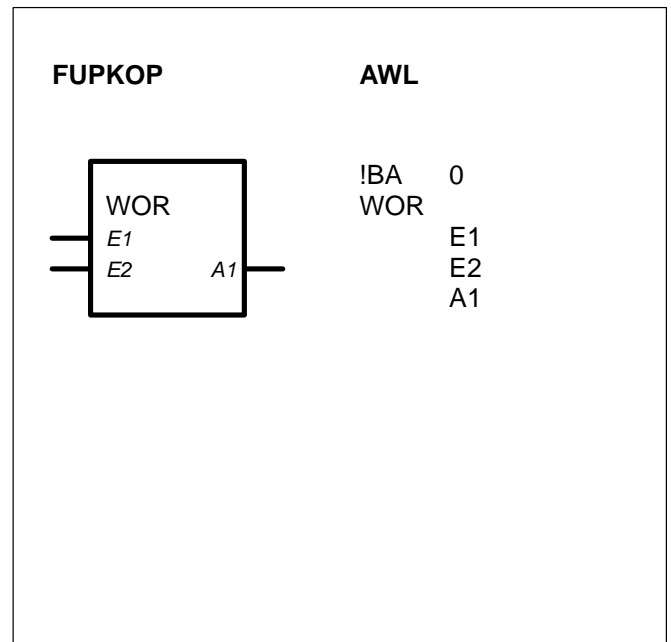


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
#OFF	K	W	N	P	J	0	0
#SEG	K	W	N	P	J	0	0
A1	A	W	N	P	N	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	WOL			
00002		PP 0	FREI	Eingang BINÄR (Baustein-Freigabe)
00003		PP 0	#OFF	# KONSTANTE (Offset-Adresse)
00004		PP 0	#SEG	# KONSTANTE (Segment-Adresse)
00005		PP 0	A1	Ausgang WORT (Wert)

Der Funktionsbaustein bildet die bitweise ODER-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	WORT	EW, MW, AW, KW	Operand 1
E2	WORT	EW, MW, AW, KW	Operand 2
A1	WORT	MW, AW	Ergebnis der ODER-Verknüpfung

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	29 µs	46 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

BESCHREIBUNG

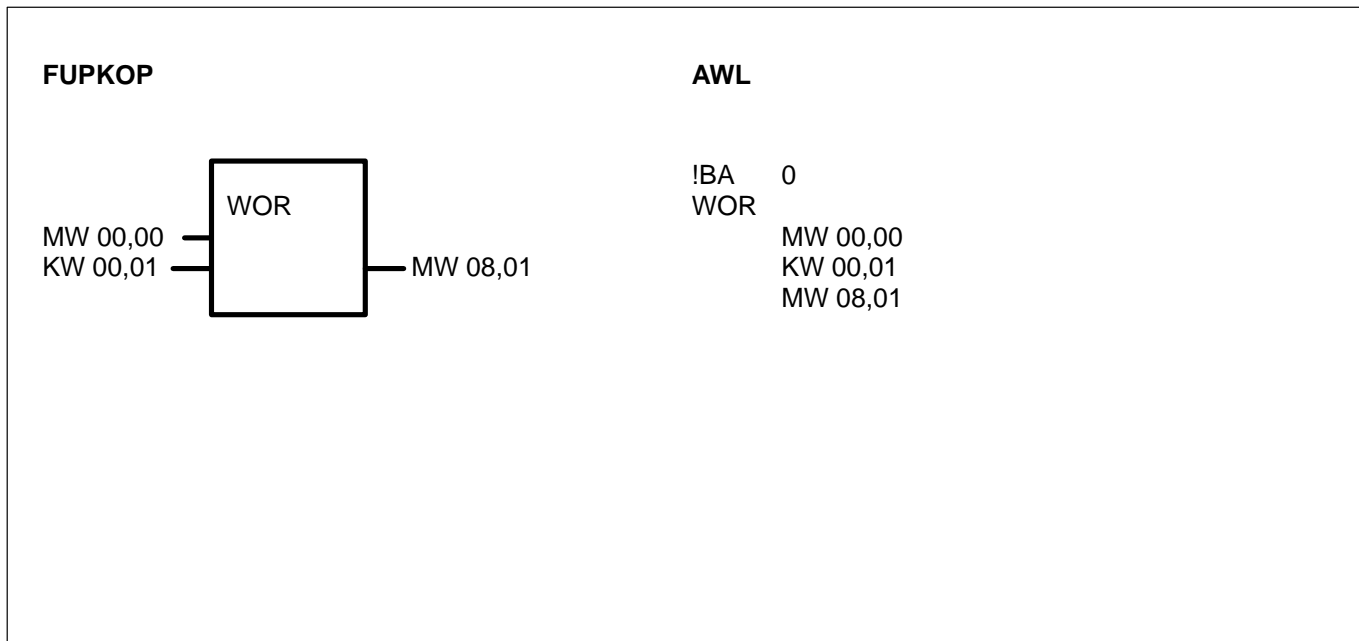
Der Funktionsbaustein bildet die bitweise ODER-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar noch invertierbar.

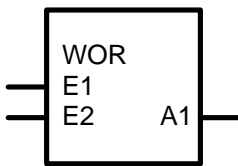
Beispiel

E1	0.0.0.0	0.0.1.1	0.0.1.0	0.1.1.0
E2	1.0.0.1	0.0.0.0	0.0.1.0	1.1.1.1
A1	1.0.0.1	0.0.1.1	0.0.1.0	1.1.1.1

Beispiel



VE-FUP-Definition

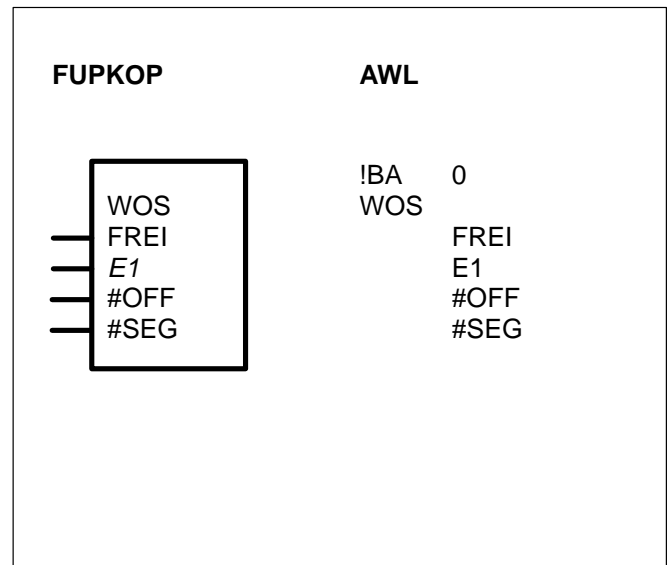


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
E2	E	W	N	P	N	0	0
A1	A	W	N	P	N	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	WOR			
00002		PP 0	E1	Eingang WORT
00003		PP 0	E2	Eingang WORT
00004		PP 0	A1	Ausgang WORT

1–Signal am Eingang FREI bewirkt, daß der Wert des Operanden am Eingang E1 gelesen und dann auf die angegebene physikalische Adresse geschrieben wird.



Parameter

FREI	BINÄR	E, M, A, S, K	Freigabe des Bausteins
E1	WORT	EW, MW, AW, KW	Eingang für den zu schreibenden Operanden
#OFF	DIREKTE KONSTANTE	#, #H	Offset–Adresse des Speicherplatzes, auf den der Wert von E1 zu schreiben ist
#SEG	DIREKTE KONSTANTE	#, #H	Segment–Adresse des Speicherplatzes, auf den der Wert von E1 zu schreiben ist

VE–DATEN

Laufzeit:	
Grundlaufzeit:	25 µs, kein Schreiben; 39 µs, Schreiben
Zusätzliche Laufzeit:	—
Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331

Beschreibung

1–Signal am Eingang FREI bewirkt, daß der Wert des Operanden am Eingang E1 gelesen und dann auf die angegebene physikalische Adresse geschrieben wird.

Bei 0–Signal am Eingang FREI erfolgt keine Bearbeitung des Bausteins.

Die physikalische Adresse besteht aus Segment und Offset. Der damit erreichbare Adreßraum beträgt 1 MByte.

Die Eingänge sind weder doppelbar noch negierbar.

FREI BINÄR
Mit dem Operanden am Eingang FREI wird die Bearbeitung des Bausteins freigegeben oder gesperrt.

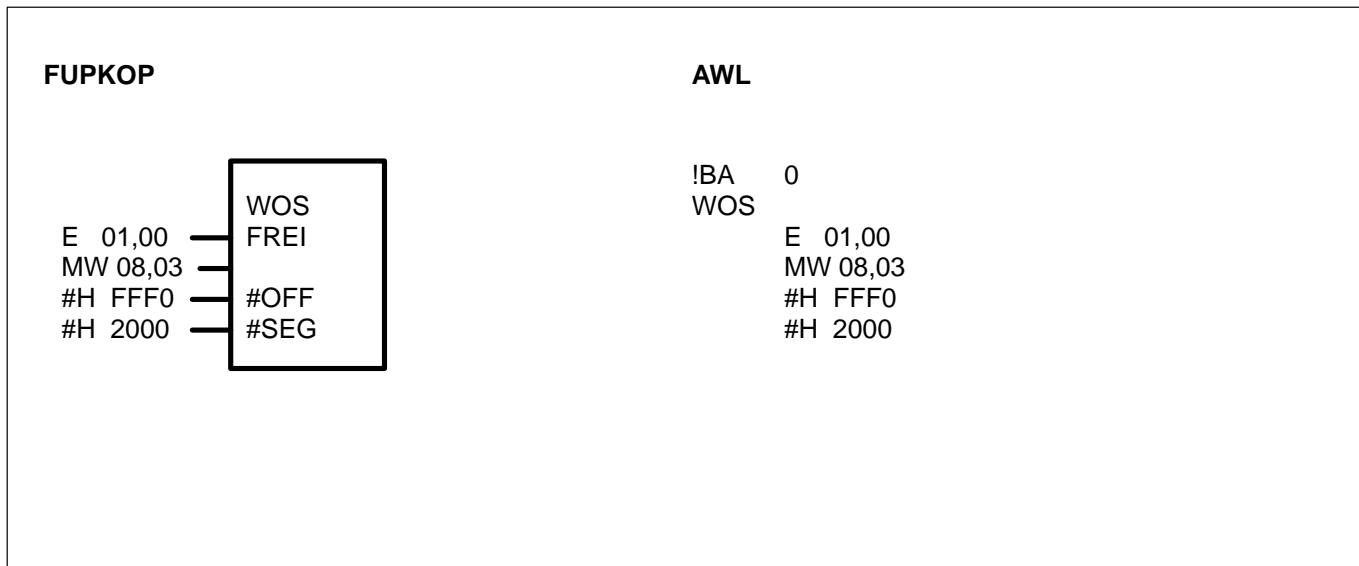
Es gilt:
FREI = 0 → Bearbeitung gesperrt
FREI = 1 → Bearbeitung freigegeben

E1 WORT
Der Operand am Eingang E1 wird gelesen und sein Wert auf die durch die Eingänge #OFF und #SEG definierte Adresse geschrieben.

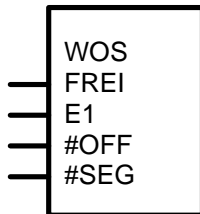
#OFF DIREKTE KONSTANTE (#, #H)
Am Eingang #OFF wird der Offset–Anteil der zu schreibenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

#SEG DIREKTE KONSTANTE (#, #H)
Am Eingang #SEG wird der Segment–Anteil der zu schreibenden Adresse angegeben. Die Angabe erfolgt als direkte Konstante.

Beispiel



VE-FUP-Definition

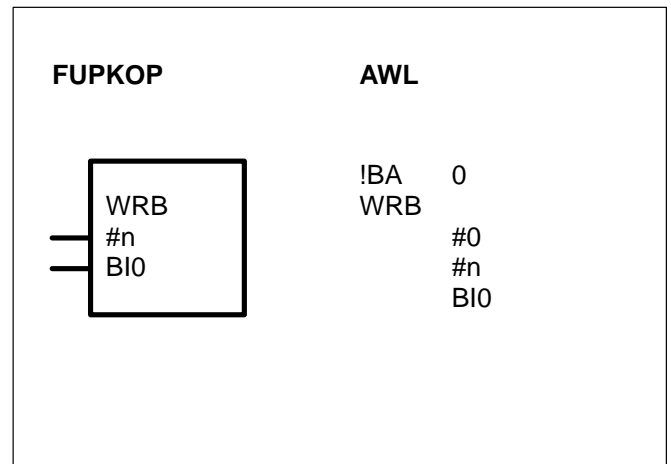


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
FREI	E	L	N	P	J	0	0
E1	E	W	N	P	N	0	0
#OFF	K	W	N	P	J	0	0
#SEG	K	W	N	P	J	0	0

VE-AWL-Definition

00000	IBA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	WOS			
00002		PP 0	FREI	Eingang BINÄR (Baustein-Freigabe)
00003		PP 0	E1	Eingang WORT (Wert)
00004		PP 0	#OFF	# KONSTANTE (Offset-Adresse)
00005		PP 0	#SEG	# KONSTANTE (Segment-Adresse)

Der Funktionsbaustein legt die Werte der Operanden an den Eingängen BI0 ... BI $n-1$ im Vergangenheitswertespeicher ab. Der zugehörige Funktionsbaustein RDB liest diese Werte wieder aus dem Vergangenheitswertespeicher aus. Die Funktionsbausteine WRB und RDB treten immer paarweise auf.



Parameter

#n	DIREKTE KONSTANTE	#, #H	Anzahl der Eingänge BI0 ... BI $n-1$
BI0	BINÄR	A, M	Eingang für die zu schreibenden Binärwerte, doppelbar (BI0 ... BI $n-1$)

VE–Daten

Laufzeit:		
Grundlaufzeit:		siehe Baustein RDB
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		entfällt
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein legt die Werte der Operanden an den Eingängen BI0 ... BI $n-1$ im Vergangenheitswertespeicher ab. Der zugehörige Funktionsbaustein RDB liest diese Werte wieder aus dem Vergangenheitswertespeicher aus. Die Funktionsbausteine WRB und RDB treten immer paarweise auf.

Um vorgefertigte Programmteile in einem Anwenderprogramm mehrfach verwenden zu können (z. B. 907 PC 31/32 Verknüpfungselemente), kann es erforderlich sein, innerhalb dieses Programmteils mit lokalen Variablen zu arbeiten. Diese lokalen Variablen verlieren außerhalb dieses Programmteils ihre Gültigkeit. Die Werte dieser lokalen Variablen werden am Ende des Programmteils vom Baustein WRB im Vergangenheitswertespeicher (VW–Speicher) des zugehörigen RDB–Bausteins abgelegt und am Anfang des Programmteils vom Baustein RDB den lokalen Variablen zugewiesen. Der

Baustein WRB legt dazu die Werte der Binäreingänge im Vergangenheitswertespeicher des zugehörigen RDB–Bausteins ab. Die Anzahl der Binäreingänge wird am Eingang #n als direkte Konstante vorgegeben. Diese Anzahl muß mit der Anzahl der Ausgänge des zugehörigen RDB–Bausteins übereinstimmen. Das Aufbereitungsprogramm "PA" trägt an dem dafür vorgesehenen Eingang in der Anweisungsliste des WRB–Bausteins den Zeiger auf den VW–Speicher des zugehörigen RDB–Bausteins ein.

#0 DIREKTE KONSTANTE

Im FUNKOP ist dieser Eingang *nicht* vorhanden. In der Anweisungsliste wird an dieser Stelle der Wert 0 als direkte Konstante vorgegeben. Die SPS trägt dann an dieser Stelle den Zeiger auf die Vergangenheitswerte des zugehörigen RDB–Bausteins ein.

#n DIREKTE KONSTANTE

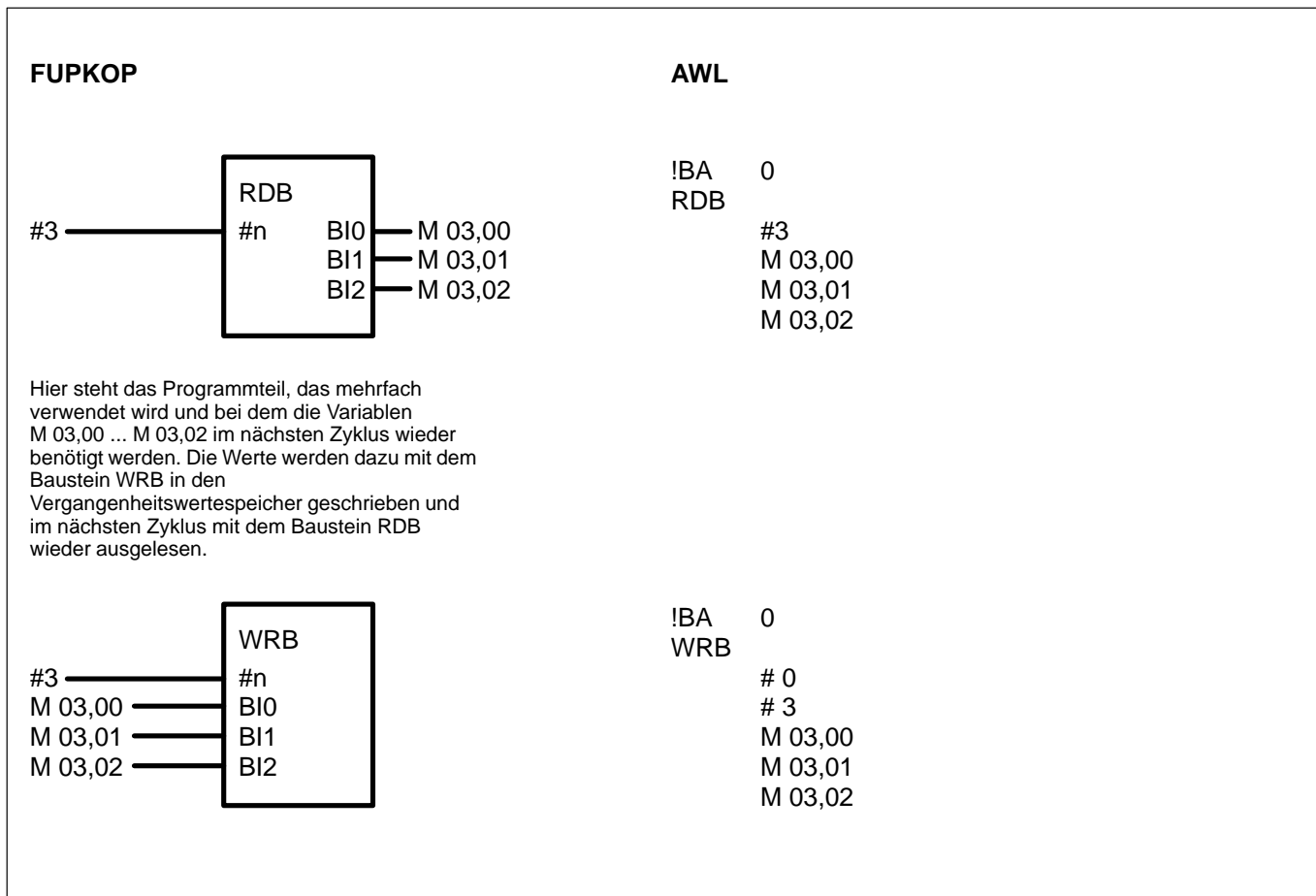
Am Eingang #n wird die Anzahl der Eingänge B10 ... BIn-1 angegeben. Die Angabe erfolgt als direkte Konstante.

Hinweis:

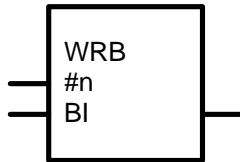
Der am Eingang #n angegebene Wert muß auch mit der Anzahl der Ausgänge des zugehörigen Bausteins RDB übereinstimmen.

B10 ... BIn-1 BINÄR

Der Eingang B10 ist doppelbar (B10 ... BIn-1). Die Werte, der an den Eingängen B10 ... BIn-1 angegebenen Operanden, werden in den Vergangenheitswertespeicher des zugehörigen Bausteins RDB geschrieben.

Beispiel

VE–FUP–Definition

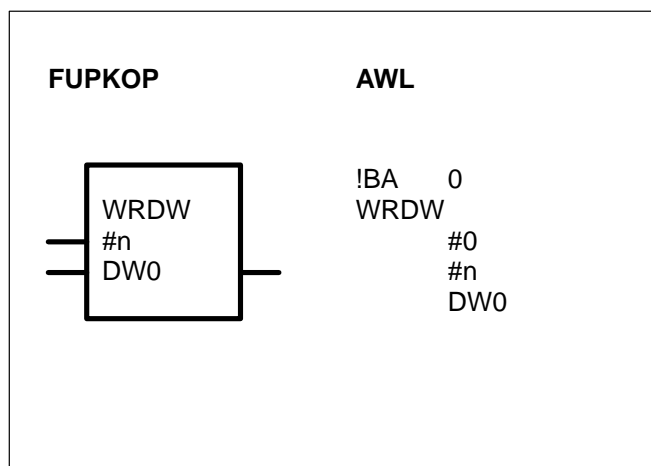


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
BI	E	L	N	P	J	1	0

VE–AWL–Definition

00000	!BA	0	Nr	Baustein–Nr (vorbelegt 0)
00001	WRB			
00002		# 0		
00003		PP 0	#n	DIREKTE KONSTANTE (Anzahl Bit)
	[1		
00004		PP 1	BI	Eingang BINÄR
]	1		

Der Funktionsbaustein legt die Werte der Operanden an den Eingängen DW0 ... DWn-1 im Vergangenheitswertespeicher ab. Der zugehörige Funktionsbaustein RDDW liest diese Werte wieder aus dem Vergangenheitswertespeicher aus. Die Funktionsbausteine WRDW und RDDW treten immer paarweise auf.



Parameter

#n	DIREKTE #, #H KONSTANTE	Anzahl der Eingänge DW0 ... DWn-1
DW0	DOPPELWORT MD	Eingang für die zu schreibenden Doppelwortwerte, doppelbar (DW0 ... DWn-1)

VE-Daten

Laufzeit:		
Grundlaufzeit:		siehe Baustein RDDW
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		entfällt
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein legt die Werte der Operanden an den Eingängen DW0 ... DWn-1 im Vergangenheitswertespeicher ab. Der zugehörige Funktionsbaustein RDDW liest diese Werte wieder aus dem Vergangenheitswertespeicher aus. Die Funktionsbausteine WRDW und RDDW treten immer paarweise auf.

Um vorgefertigte Programmteile in einem Anwenderprogramm mehrfach verwenden zu können (z. B. 907 PC 31/32 Verknüpfungselemente), kann es erforderlich sein, innerhalb dieses Programmteils mit lokalen Variablen zu arbeiten. Diese lokalen Variablen verlieren außerhalb dieses Programmteils ihre Gültigkeit. Die Werte dieser lokalen Variablen werden am Ende des Programmteils vom Baustein WRDW im Vergangenheitswertespeicher (VW-Speicher) des zugehörigen RDDW-Bausteins abgelegt und am Anfang des Programmteils

vom Baustein RDDW den lokalen Variablen zugewiesen. Der Baustein WRDW legt dazu die Werte der Doppelworteingänge im Vergangenheitswertespeicher des zugehörigen RDDW-Bausteins ab. Die Anzahl der Doppelworteingänge wird am Eingang #n als direkte Konstante vorgegeben. Diese Anzahl muß mit der Anzahl der Ausgänge des zugehörigen RDDW-Bausteins übereinstimmen. Das Aufbereitungsprogramm "PA" trägt an dem dafür vorgesehenen Eingang in der Anweisungsliste des WRDW-Bausteins den Zeiger auf den VW-Speicher des zugehörigen RDDW-Bausteins ein.

#0 DIREKTE KONSTANTE

Im FUPKOP ist dieser Eingang *nicht* vorhanden. In der Anweisungsliste wird an dieser Stelle der Wert 0 als direkte Konstante vorgegeben. Die SPS trägt dann an dieser Stelle den Zeiger auf die Vergangenheitswerte des zugehörigen RDDW-Bausteins ein.

#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der Eingänge DW0 ... DWn-1 angegeben. Die Angabe erfolgt als direkte Konstante.

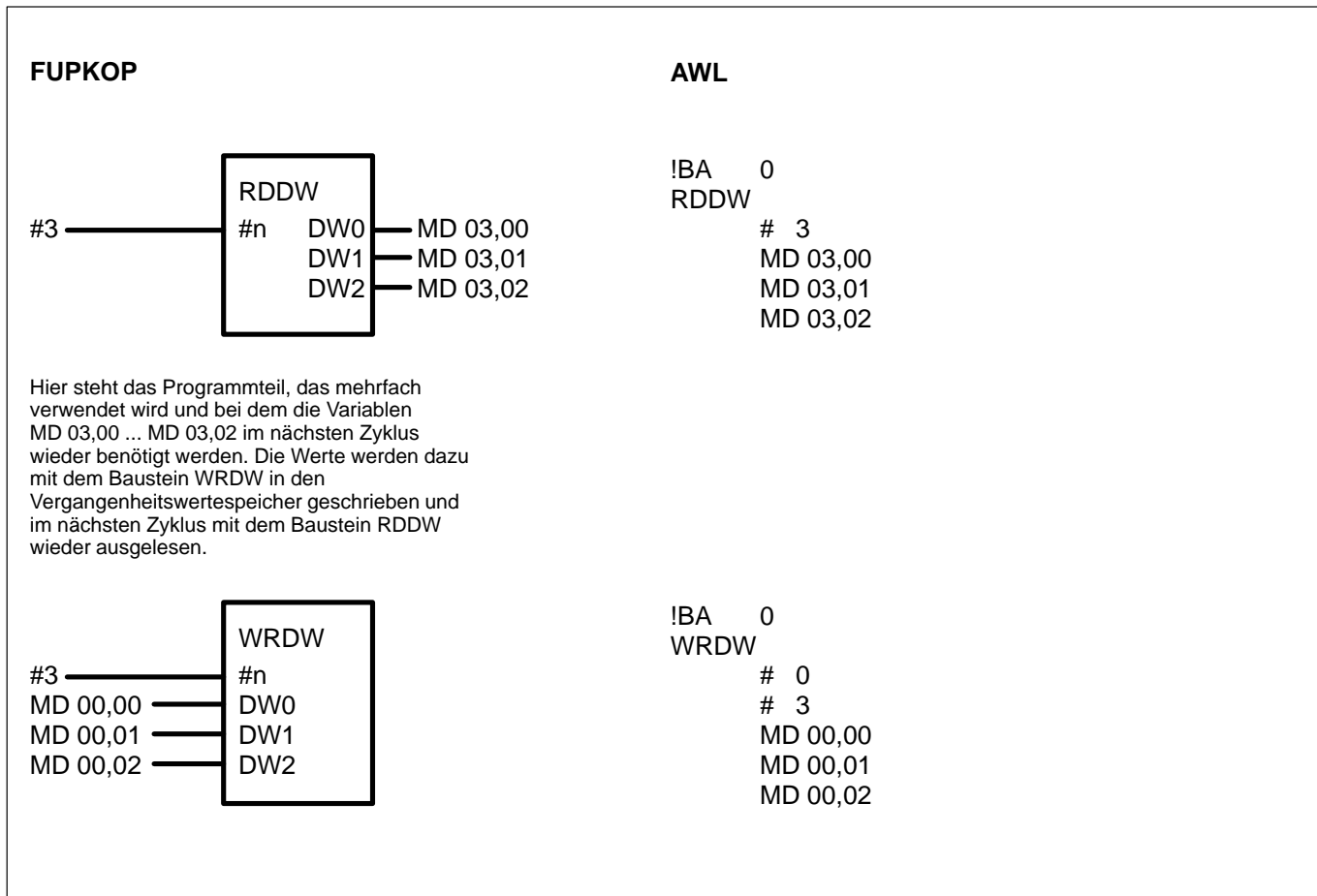
Hinweis:

Der am Eingang #n angegebene Wert muß auch mit der Anzahl der Ausgänge des zugehörigen Bausteins RDDW übereinstimmen.

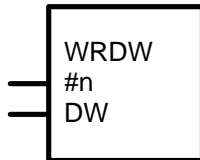
DW0 ... DWn-1 DOPPELWORT

Der Eingang DW0 ist doppelbar (DW0 ... DWn-1). Die Werte, die an den Eingängen DW0 ... DWn-1 angegebenen Operanden, werden in den Vergangenheitswertespeicher des zugehörigen Bausteins RDDW geschrieben.

Beispiel



VE-FUP-Definition



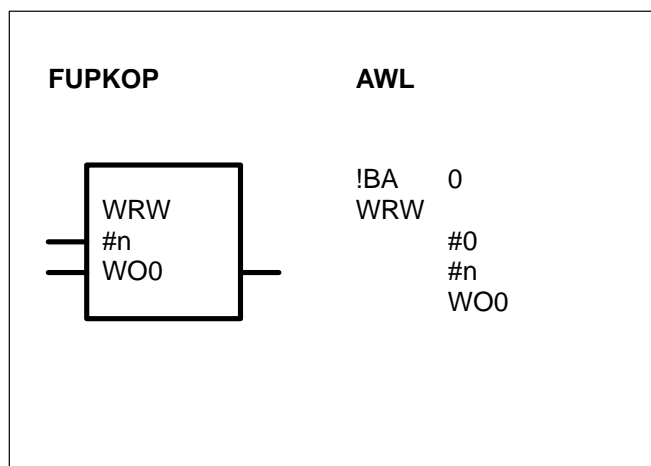
Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
DW	E	D	N	P	J	1	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	WRDW			
00002		#0		
00003		PP 0	#n	# KONSTANTE (Anzahl Doppelworte)
	[1		
00004		PP 1	DW	Eingang DOPPELWORT
]	1		

WRW

Der Funktionsbaustein legt die Werte der Operanden an den Eingängen WO0 ... WOn-1 im Vergangenheitswertespeicher ab. Der zugehörige Funktionsbaustein RDW liest diese Werte wieder aus dem Vergangenheitswertespeicher aus. Die Funktionsbausteine WRW und RDW treten immer paarweise auf.



Parameter

#n	DIREKTE KONSTANTE	#, #H	Anzahl der Eingänge WO0 ... WOn-1
WO0	WORT	AW, MW	Eingang für die zu schreibenden Wort-Werte, doppelbar (WO0 ... WOn-1)

VE-Daten

Laufzeit:		
Grundlaufzeit:		siehe Baustein RDW
Zusätzliche Laufzeit:		—
Aktualisierung der Ausgänge:		entfällt
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein legt die Werte der Operanden an den Eingängen WO0 ... WOn-1 im Vergangenheitswertespeicher ab. Der zugehörige Funktionsbaustein RDW liest diese Werte wieder aus dem Vergangenheitswertespeicher aus. Die Funktionsbausteine WRW und RDW treten immer paarweise auf.

Um vorgefertigte Programmteile in einem Anwenderprogramm mehrfach verwenden zu können (z.B. 907 PC 31/32 Verknüpfungselemente), kann es erforderlich sein, innerhalb dieses Programmteils mit lokalen Variablen zu arbeiten. Diese lokalen Variablen verlieren außerhalb dieses Programmteils ihre Gültigkeit. Die Wer-

te dieser lokalen Variablen werden am Ende des Programmteils vom Baustein WRW im Vergangenheitswertespeicher (VW-Speicher) des zugehörigen RDW-Bausteins abgelegt und am Anfang des Programmteils vom Baustein RDW den lokalen Variablen zugewiesen. Der Baustein WRW legt dazu die Werte der Worteingänge WO0 ... WOn-1 im Vergangenheitswertespeicher des zugehörigen RDW-Bausteins ab. Die Anzahl der Worteingänge wird am Eingang #n als direkte Konstante vorgegeben. Diese Anzahl muß mit der Anzahl der Ausgänge des zugehörigen RDW-Bausteins übereinstimmen. Das Aufbereitungsprogramm "PA" trägt an dem dafür vorgesehenen Eingang in der Anweisungsliste des WRW-Bausteins den Zeiger auf den VW-Speicher des zugehörigen RDW-Bausteins ein.

#0 DIREKTE KONSTANTE

Im FUPKOP ist dieser Eingang *nicht* vorhanden. In der Anweisungsliste wird an dieser Stelle der Wert 0 als direkte Konstante vorgegeben. Die SPS trägt dann an dieser Stelle den Zeiger auf die Vergangenheitswerte des zugehörigen RDW–Bausteins ein.

WO0 ... WOn–1 BINÄR

Der Eingang WO0 ist doppelbar (WO0 ... WOn–1). Die Werte, die an den Eingängen WO0 ... WOn–1 angegebenen Operanden, werden in den Vergangenheitswertespeicher des zugehörigen Bausteins RDW geschrieben.

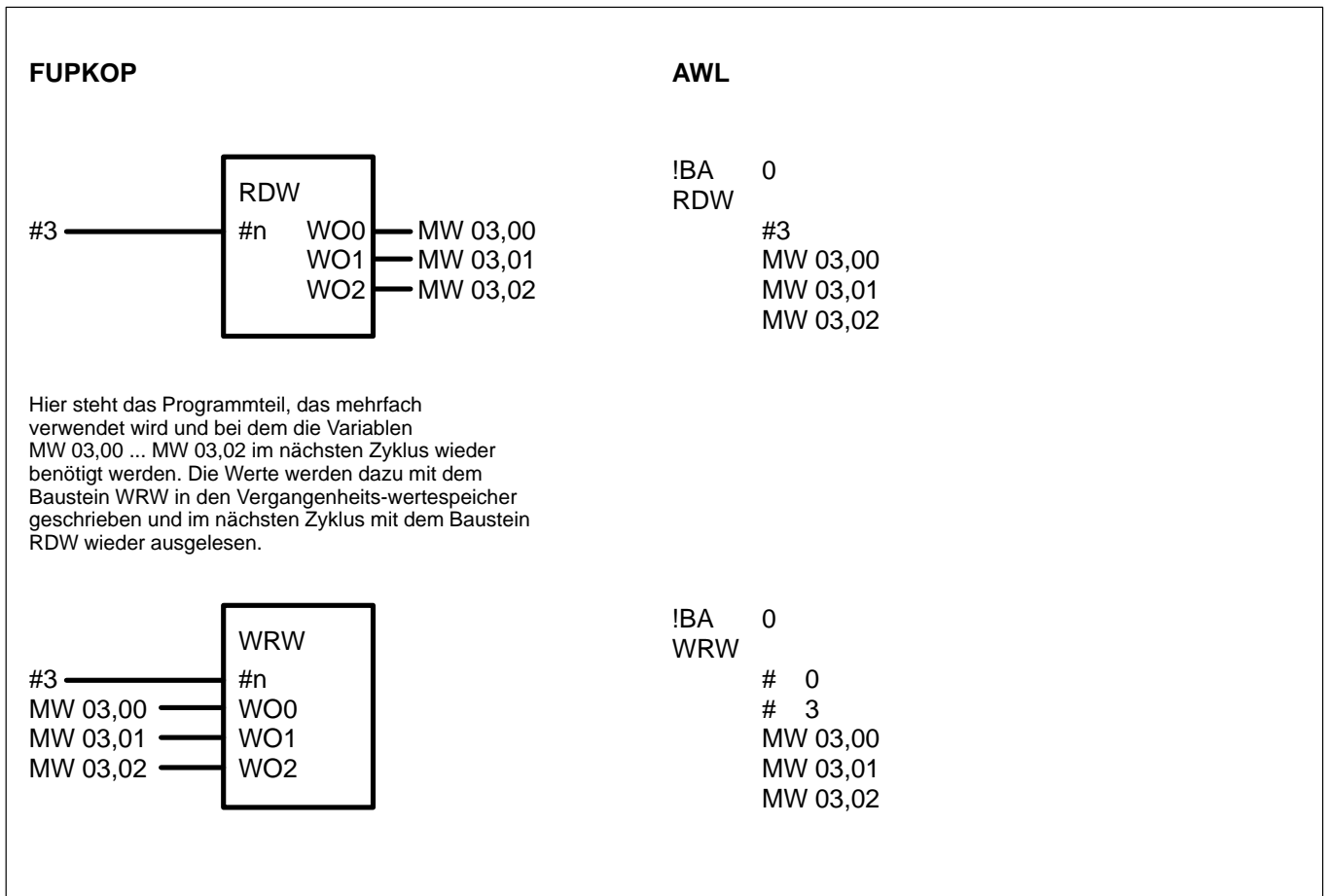
#n DIREKTE KONSTANTE

Am Eingang #n wird die Anzahl der Eingänge WO0 ... WOn–1 angegeben. Die Angabe erfolgt als direkte Konstante.

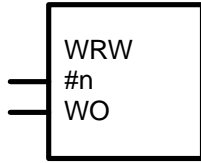
Hinweis:

Der am Eingang #n angegebene Wert muß auch mit der Anzahl der Ausgänge des zugehörigen Bausteins RDW übereinstimmen.

Beispiel



VE-FUP-Definition

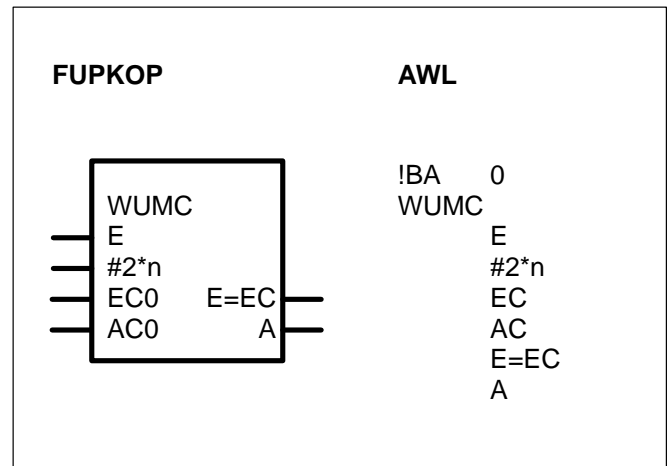


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
#n	K	W	N	P	J	0	0
WO	E	W	N	P	J	1	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	WRW			
00002		#0		
00003		PP 0	#n	#KONSTANTE (Anzahl Worte)
	[1		
00004		PP 1	WO	Eingang WORT
]	1		

Der Funktionsbaustein vergleicht den Wert des Operanden am Eingang E mit den Referenzwerten der Operanden an den Eingängen EC0 ... ECn-1. Stimmt der Eingang E mit mindestens einem der Referenzwerte EC überein, so wird der Ausgang E=EC auf 1 gesetzt. Der Ausgang A erhält den Wert des Ausgangscodes AC, der dem gefundenen Referenzwert EC zugeordnet ist.



Parameter

E	WORT	EW, AW, MW, KW	Eingang
#2*n	DIREKTE KONSTANTE	#, #H	Anzahl n der Referenzwerte (multipliziert mit 2)
EC0	WORT	EW, AW, MW, KW	Referenzwert; Eingang is doppelbar
AC0	WORT	EW, AW, MW, KW	Ausgangscodes; Eingang is doppelbar
E=EC	BINÄR	A, M	Koinzidenzanzeige
A	WORT	AW, MW	Ausgabe des Werts des Ausgangscodes

VE-Daten

Laufzeit:		
Grundlaufzeit:		40,0 µs
Zusätzliche Laufzeit:		7,5 µs, pro eingegebenem Vergleichswert
Aktualisierung der Ausgänge:		ja
Anzahl der Vergangenheitswerte:		keine
Verfügbar ab:		ABB Procontic CS31 / 907 PC 331

Beschreibung

Der Funktionsbaustein vergleicht den Wert des Operanden am Eingang E mit den Referenzwerten der Operanden an den Eingängen EC0 ... ECn-1. Stimmt der Eingang E mit mindestens einem der Referenzwerte EC überein, so wird der Ausgang E=EC auf 1 gesetzt. Der Ausgang A erhält den Wert des Ausgangscodes AC, der dem gefundenen Referenzwert EC zugeordnet ist.

Jedem Referenzwert an den Eingängen EC_i ist ein Operand für den Ausgangscodes AC_i zugeordnet. Die Zuordnung von EC zu AC ist am Index i zu erkennen. Der Index beginnt mit 0 und wird bei Dopplung automatisch erzeugt.

Die Anzahl der Eingänge EC und AC ist am Eingang #2*n als direkte Konstante anzugeben.

Die Eingänge und Ausgänge sind nicht negierbar/invertierbar.

E WORT
Am Eingang E wird der Operand angegeben, dessen Wert mit den Werten der n Referenzwerte (EC0 ... ECn-1) verglichen werden soll.

#2*n DIREKTE KONSTANTE (#, #H)
Am Eingang #2*n wird die Gesamtzahl (2*n) der Referenzwerte (EC0 ... ECn-1) und Ausgangscodes (AC0 ... ACn-1) angegeben. Die Angabe erfolgt als indirekte Konstante.

EC0 ... ECn-1 WORT
Der Eingang EC0 ist entsprechend der benötigten Anzahl von Referenzwerten zu doppeln. An den Eingängen EC0 ... ECn-1 werden die Operanden für die Referenzwerte angegeben. Der Wert des Operanden am Eingang E1 wird mit den Referenzwerten verglichen.

AC0 ... ACn-1 WORT

Durch die Dopplung des Eingangs EC0 wird der Eingang AC0 automatisch mitgedoppelt.

An den Eingängen AC0 ... ACn-1 werden die Ausgangs-codes angegeben. Bei Übereinstimmung des Eingangs E mit einem der Referenzwerte ECi wird der Ausgangscode ACi am Ausgang A ausgegeben.

Zuordnung der Referenzwerte und Ausgangscodes:

EC0 <-> AC0

EC1 <-> AC1

·
·

ECn-1 <-> ACn-1

E=EC BINÄR

Am Ausgang E=EC wird die Übereinstimmung des Operandenwertes des Eingangs E mit einem der Referenzwerte signalisiert.

Es gilt: E=EC = 0 -> keine Übereinstimmung

E=EC = 1 -> Übereinstimmung

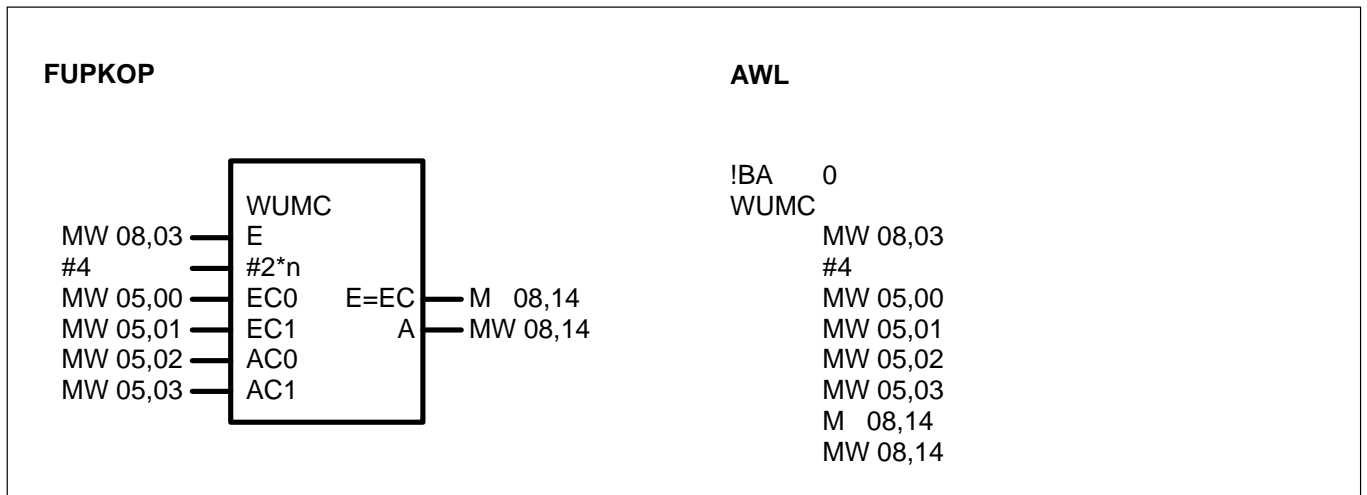
A WORT

Bei Übereinstimmung des Eingangs E mit einem der Referenzwerte ECi wird der Ausgangscode ACi am Ausgang A ausgegeben.

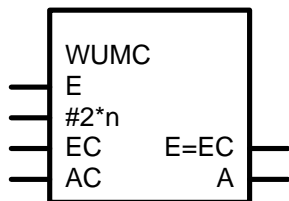
Es gilt: A = 0 -> keine Übereinstimmung

A = ACi -> Übereinstimmung

Beispiel



VE-FUP-Definition

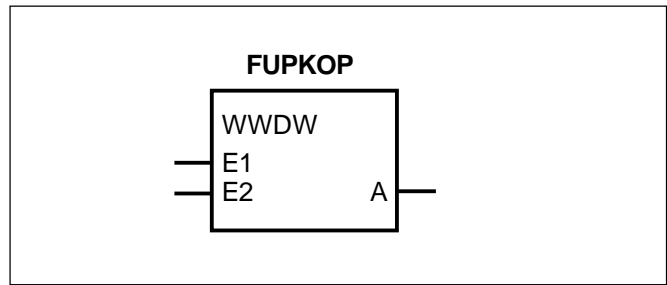


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E	E	W	N	P	J	0	0
#2*n	K	W	N	P	J	0	0
EC	E	W	N	P	J	1	0
AC	E	W	N	P	J	1	0
E=EC	A	L	N	P	J	0	0
A	A	W	N	P	J	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	WUMC			
00002		PP 0	E	Eingang WORT (Eingangswert)
00003		PP 0	#2*n	# KONSTANTE (Anzahl EC und AC)
	[1		
00004		PP 1	EC	Eingang WORT (Referenzwert)
]	1		
	[1		
00005		PP 1	AC	Eingang WORT (Ausgangswert)
]	1		
00006		PP 0	E=EC	Ausgang BINÄR (Eing.wert=Ecode)
00007		PP 0	A	Ausgang WORT (Ausgangswert)

Der Funktionsbaustein packt 2 16-Bit-Worte in ein 32-Bit-Doppelwort.



Parameter

E1	WORT	EW, MW	Eingang E1 wird zum Low Word des Ausgangs A
E2	WORT	EW, MW	Eingang E2 wird zum High Word des Ausgangs A
A	DOPPELWORT	MD	Doppelwortausgang

VE-Daten

Aktualisierung der Ausgänge:	ja
Anzahl der Vergangenheitswerte:	keine
Verfügbar ab:	07 KR 91 R2 Index c, 07 KT 92 R2 Index c, 07 KT 93 R2 Index f, 07 KT 94

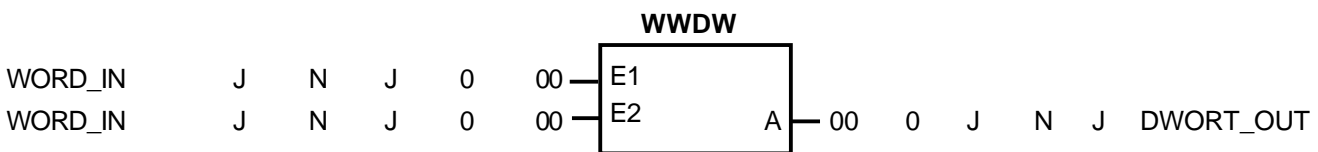
Beschreibung

Der Baustein packt 2 16-Bit-Worte in ein 32-Bit-Doppelwort. Der Wert von E1 wird zum Low Word des Ausgangs A und der Wert von E2 wird zum High Word des Ausgangs A (Doppelwort).

- E1 WORT**
Der Wert von E1 wird zum Low Word (Bit 0 bis Bit 15) des Ausgangs A.
- E2 WORT**
Der Wert von E2 wird zum High Word (Bit 16 bis Bit 31) des Ausgangs A.
- A DOPPELWORT**
Erzeugtes Doppelwort aus E1 und E2

Grafik

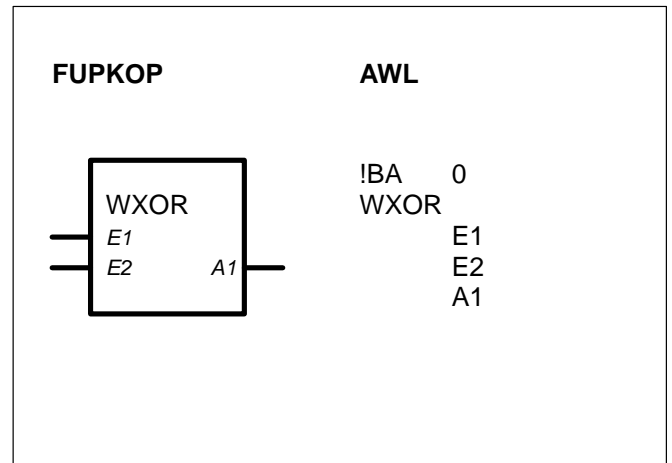
Plaus-gruppe	Pf1	Inv	Anz	Art	Nr		Nr	Art	Anz	Inv	Pf1	Plaus-gruppe
--------------	-----	-----	-----	-----	----	--	----	-----	-----	-----	-----	--------------



Source

SN	ZN	Befehl / Variable	Symbol	Langtext
00000	00000	WWDW		
	00001	PP 00	E1	
	00002	PP 00	E2	
	00003	PP 00	A	
	00004			

Der Funktionsbaustein bildet die bitweise XOR-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.



Parameter

E1	WORT	EW, MW, AW, KW	Operand 1
E2	WORT	EW, MW, AW, KW	Operand 2
A1	WORT	MW, AW	Ergebnis der XOR-Verknüpfung

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	28 µs	46 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / 907 PC 331	907 PC 331 / 07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

Beschreibung

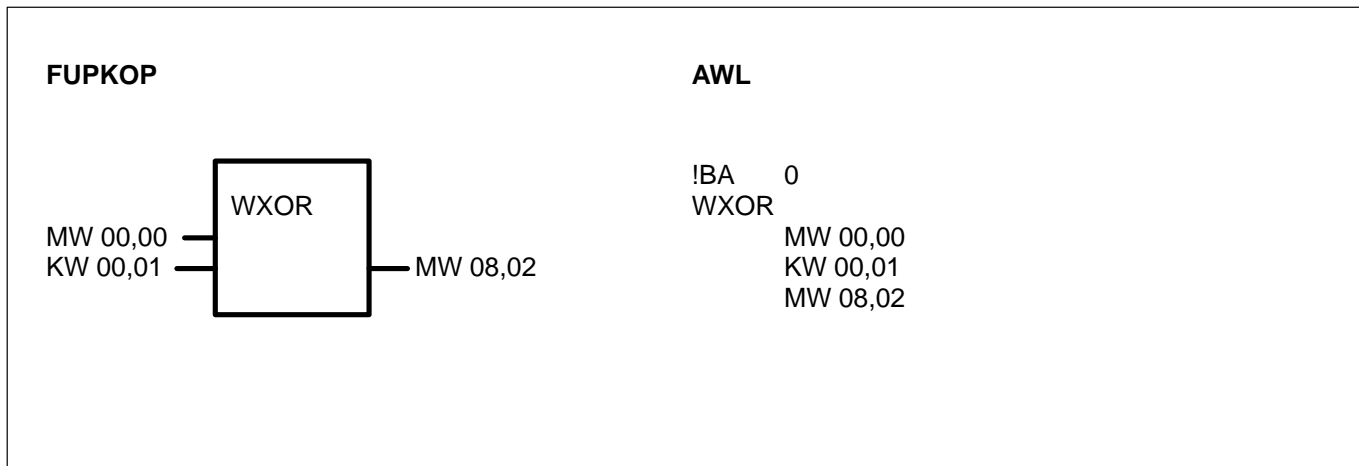
Der Funktionsbaustein bildet die bitweise XOR-Verknüpfung der an den Eingängen E1 und E2 anliegenden Operanden. Das Ergebnis wird dem Operanden am Ausgang A1 zugewiesen.

Die Eingänge und der Ausgang sind weder doppelbar noch negierbar noch invertierbar.

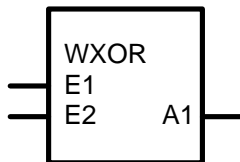
Beispiel

E1	0.0.0.0	0.0.1.1	0.0.1.0	0.1.1.0
E2	1.0.0.1	0.0.0.0	0.0.1.0	1.1.1.1
A1	1.0.0.1	0.0.1.1	0.0.0.0	1.0.0.1

Beispiel



VE-FUP-Definition

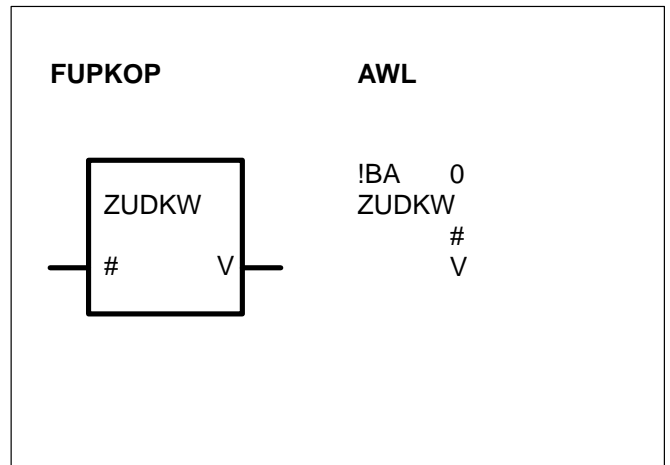


Param.	Param. Gruppe	Param. Typ	Inv.	Belegung	Darst. Schirm	Param. Block	Doppl. Art
E1	E	W	N	P	N	0	0
E2	E	W	N	P	N	0	0
A1	A	W	N	P	N	0	0

VE-AWL-Definition

00000	!BA	0	Nr	Baustein-Nr. (vorbelegt 0)
00001	WXOR			
00002		PP 0	E1	Eingang WORT
00003		PP 0	E2	Eingang WORT
00004		PP 0	A1	Ausgang WORT

Der Funktionsbaustein dient zur Zuweisung eines Zahlenwertes an eine Wort-Variable.
Der Zahlenwert wird als direkte Konstante vorgegeben.



Parameter

#	DIREKTE KONSTANTE	#, #H	Zahlenwert, der der Wort-Variablen am Ausgang V zuzuweisen ist
V	WORT	EW, MW, AW	Wort-Variable, der der Zahlenwert vom Eingang # zuzuweisen ist

VE-Daten

Serie 90

Serie 30, 40, 50

Laufzeit:		
Grundlaufzeit:	10 µs	20 µs
Zusätzliche Laufzeit:	—	—
Aktualisierung der Ausgänge:	ja	ja
Anzahl der Vergangenheitswerte:	keine	keine
Verfügbar ab:	ABB Procontic CS31 / Kommunikationsmodul 07 KP 62	07 KR 31 V 1.0, 07 CR 41, 07 CT 41, 07 KR 51, 07 KT 51

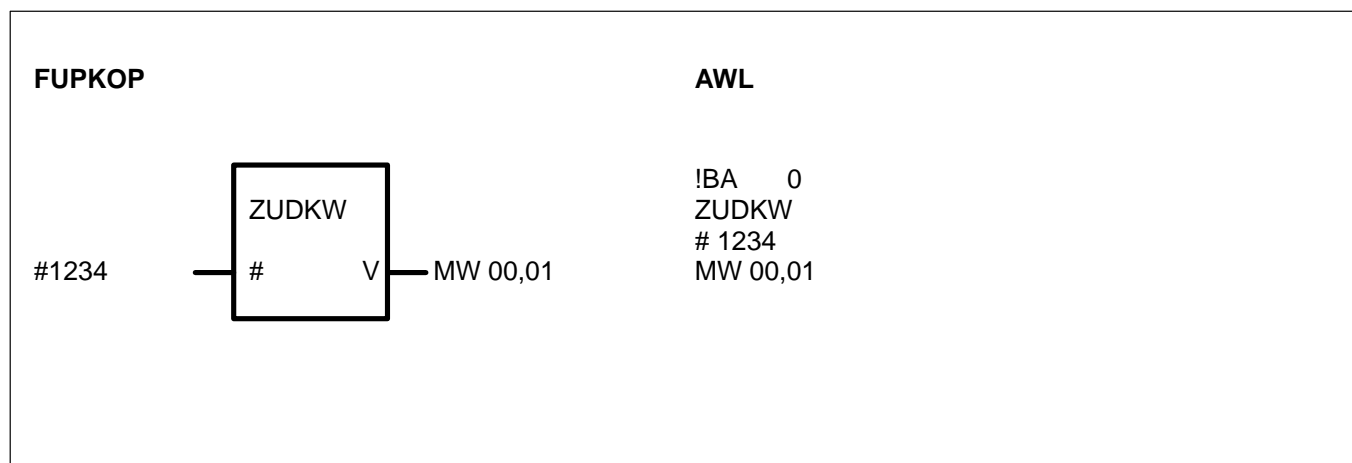
Beschreibung

Der Funktionsbaustein dient zur Zuweisung eines Zahlenwertes an eine Wort-Variable.
Der Zahlenwert wird als direkte Konstante vorgegeben.

DIREKTE KONSTANTE (#, #H)
Dieser Zahlenwert wird der Wort-Variablen am Ausgang V zugewiesen.

V WORT
Wort-Variable, der der Zahlenwert vom Eingang # zugewiesen wird.

Beispiel





Druck auf chlorfrei gebleichtem Papier

ABB Schalt- und Steuerungstechnik GmbH
Eppelheimer Straße 82 Postfach 10 50 09
D-69123 Heidelberg D-69040 Heidelberg

Telefon (06221) 777-0
Telefax (06221) 777-111
E-Mail desst.helpline@de.abb.com
Internet <http://www.abb-sst.de>

Printed in the Federal Republic of Germany (08.99)