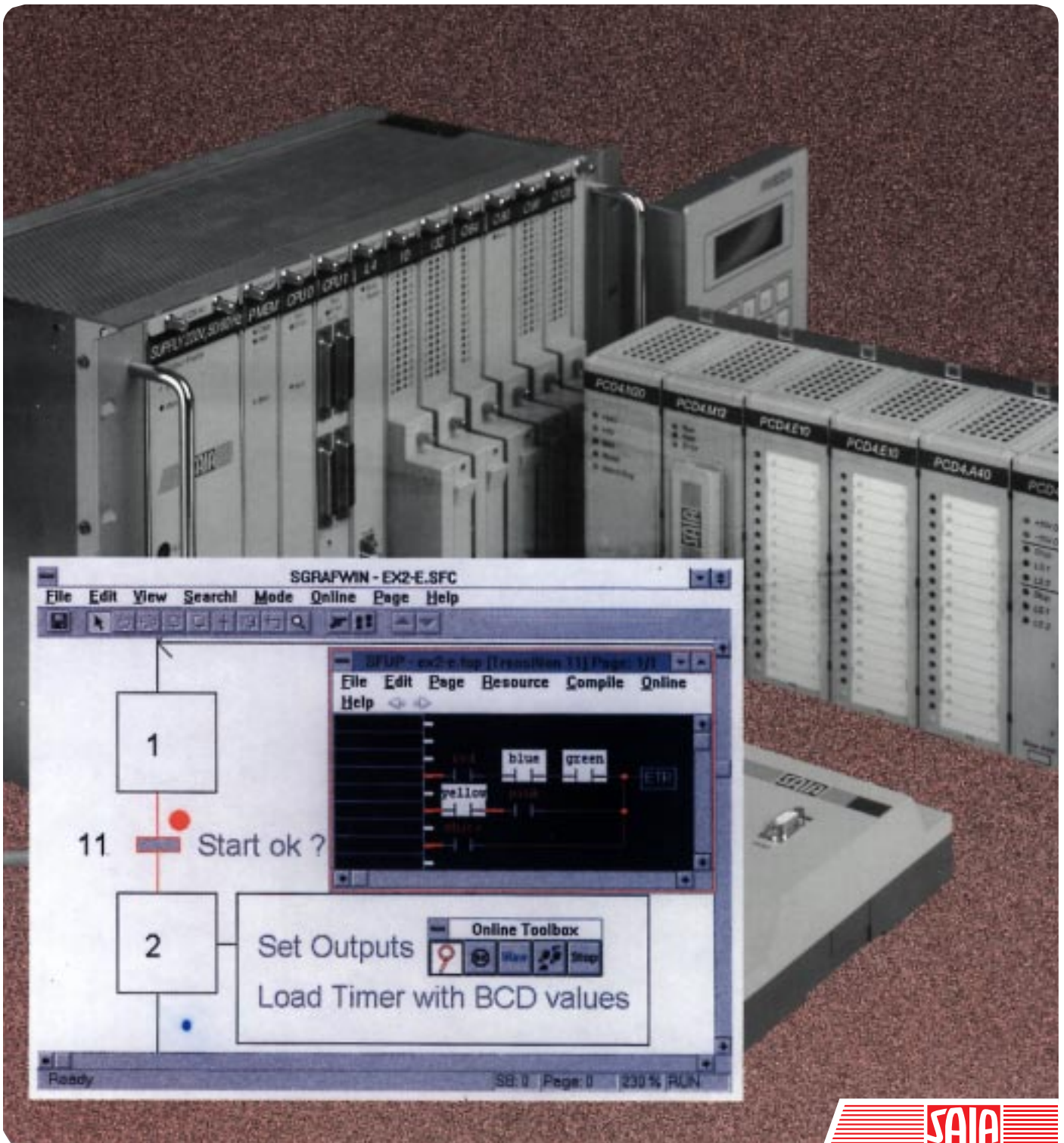


SAIA®PCD
Process Control Devices

**Die Funktionsfamilien des
FUPLA und des KOPLA**



SAIA-Burgess Gesellschaften

| | | | |
|--------------------|---|--------------------|---|
| Schweiz | SAIA-Burgess Electronics AG Freiburgstrasse 33 CH-3280 Murten ☎ 026 672 77 77, Fax 026 670 19 83 | Frankreich | SAIA-Burgess Electronics Sàrl. 10, Bld. Louise Michel F-92230 Gennevilliers ☎ 01 46 88 07 70, Fax 01 46 88 07 99 |
| Deutschland | SAIA-Burgess Electronics GmbH Daimlerstrasse 1k D-63303 Dreieich ☎ 06103 89 060, Fax 06103 89 06 66 | Niederlande | SAIA-Burgess Electronics B.V. Hanzeweg 12c NL-2803 MC Gouda ☎ 0182 54 31 54, Fax 0182 54 31 51 |
| Österreich | SAIA-Burgess Electronics Ges.m.b.H. Schallmooser Hauptstrasse 38 A-5020 Salzburg ☎ 0662 88 49 10, Fax 0662 88 49 10 11 | Belgien | SAIA-Burgess Electronics Belgium Avenue Roi Albert 1er, 50 B-1780 Wemmel ☎ 02 456 06 20, Fax 02 460 50 44 |
| Italien | SAIA-Burgess Electronics S.r.l. Via Cadamosto 3 I-20094 Corsico MI ☎ 02 48 69 21, Fax 02 48 60 06 92 | Ungarn | SAIA-Burgess Electronics Automation Kft. Liget utca 1. H-2040 Budaörs ☎ 23 501 170, Fax 23 501 180 |

Vertretungen

| | | | |
|----------------------------|---|------------------------------|---|
| Gross-britannien | Canham Controls Ltd. 25 Fenlake Business Centre, Fengate Peterborough PE1 5BQ UK ☎ 01733 89 44 89, Fax 01733 89 44 88 | Portugal | INFOCONTROL Electronica e Automatismo LDA. Praceta Cesário Verde, No 10 s/cv, Massamá P-2745 Queluz ☎ 21 430 08 24, Fax 21 430 08 04 |
| Dänemark | Malthe Winje Automation AS Håndværkerbyen 57 B DK-2670 Greve ☎ 70 20 52 01, Fax 70 20 52 02 | Spanien | Tecnosistemas Medioambientales, S.L. Poligono Industrial El Cabril, 9 E-28864 Ajalvir, Madrid ☎ 91 884 47 93, Fax 91 884 40 72 |
| Norwegen | Malthe Winje Automasjon AS Haukelivn 48 N-1415 Oppegård ☎ 66 99 61 00, Fax 66 99 61 01 | Tschechische Republik | ICS Industrie Control Service, s.r.o. Modranská 43 CZ-14700 Praha 4 ☎ 2 44 06 22 79, Fax 2 44 46 08 57 |
| Schweden | Malthe Winje Automation AB Truckvägen 14A S-194 52 Upplands Väsby ☎ 08 795 59 10, Fax 08 795 59 20 | Polen | SABUR Ltd. ul. Druzynowa 3A PL-02-590 Warszawa ☎ 22 844 63 70, Fax 22 844 75 20 |
| Suomi/ Finnland | ENERGEL OY Atomitie 1 FIN-00370 Helsinki ☎ 09 586 2066, Fax 09 586 2046 | | |
| Australien | Siemens Building Technologies Pty. Ltd. Landis & Staefa Division 411 Ferntree Gully Road AUS-Mount Waverley, 3149 Victoria ☎ 3 9544 2322, Fax 3 9543 8106 | Argentinien | MURTEN S.r.l. Av. del Libertador 184, 4° "A" RA-1001 Buenos Aires ☎ 054 11 4312 0172, Fax 054 11 4312 0172 |

Kundendienst

| | |
|------------|---|
| USA | SAIA-Burgess Electronics Inc. 1335 Barclay Boulevard Buffalo Grove, IL 60089, USA ☎ 847 215 96 00, Fax 847 215 96 06 |
|------------|---|

SAIA® Process Control Devices

Programmier-Werkzeuge für MS WINDOWS

Die Funktionsfamilien des FUPLA und des KOPLA

PG4 - Version 1.3

SAIA-Burgess Electronics AG 1996. Alle Rechte vorbehalten
Ausgabe 26/749 D1 - 02.1996

Technische Änderungen vorbehalten

Anpassungen

Handbuch:

Die Funktionsfamilien des FUPLA und des KOPLA - PG4 Version 1.3 - Ausgabe D1

| Datum | Abschnitt | Seite | Beschreibung |
|--------------|------------------|--------------|---|
| 27.10.2000 | --- | --- | Kleine Anpassungen für die Support Homepage |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Inhalt

Das vorliegende Handbuch beschreibt sämtliche Funktionen des Standard-FUPLA und des KOPLA des Programmier-Werkzeuges PG4.

Dieses Handbuch ist das Kapitel 4.4 und 4.5 des PG4-Handbuches (26/748 D).

Die Beschreibungen der einzelnen Funktionen sind weitgehend identisch mit den "Infos" zu den einzelnen FBoxen auf dem Bildschirm.

| Uebersicht über die Funktionsfamilien: | | Seite |
|---|--|--------------|
| 4.4 | Die Funktionsfamilien des FUPLA | 4-5 |
| 4.4.1 | Binäre - Funktionen | 4-15 |
| 4.4.2 | Flip-Flops (Kippstufen) | 4-27 |
| 4.4.3 | Zähler (Counter) | 4-37 |
| 4.4.4 | Zeitfunktionen (Timer) | 4-43 |
| 4.4.5 | Blinker | 4-53 |
| 4.4.6 | Arithmetik im Ganzzahlformat | 4-55 |
| 4.4.7 | Arithmetik im Fließpunktformat | 4-71 |
| 4.4.8 | Wandler (Binär - Ganzzahl - Fließpunkt) | 4-85 |
| 4.4.9 | Indirekte Adressierung | 4-99 |
| 4.4.10 | Move Data | 4-107 |
| 4.4.11 | Anzeigen (Displays) | 4-117 |
| 4.4.12 | GRAFTEC - Funktionen | 4-121 |
| 4.4.13 | Spezialfunktionen (Watch-Dog) | 4-129 |
| 4.4.14 | Analog Module | 4-131 |
| 4.4.15 | Regelung (Einfache PID-Regelung) | 4-141 |
| 4.4.16 | Benutzerdefinierte Funktionen | 4-143 |
| 4.4.17 | Serielle Kommunikation (Datenaustausch) | 4-149 |
| 4.5 | Die Funktionen des KOPLA (Ladder Diagram) | 4-165 |

Notizen:



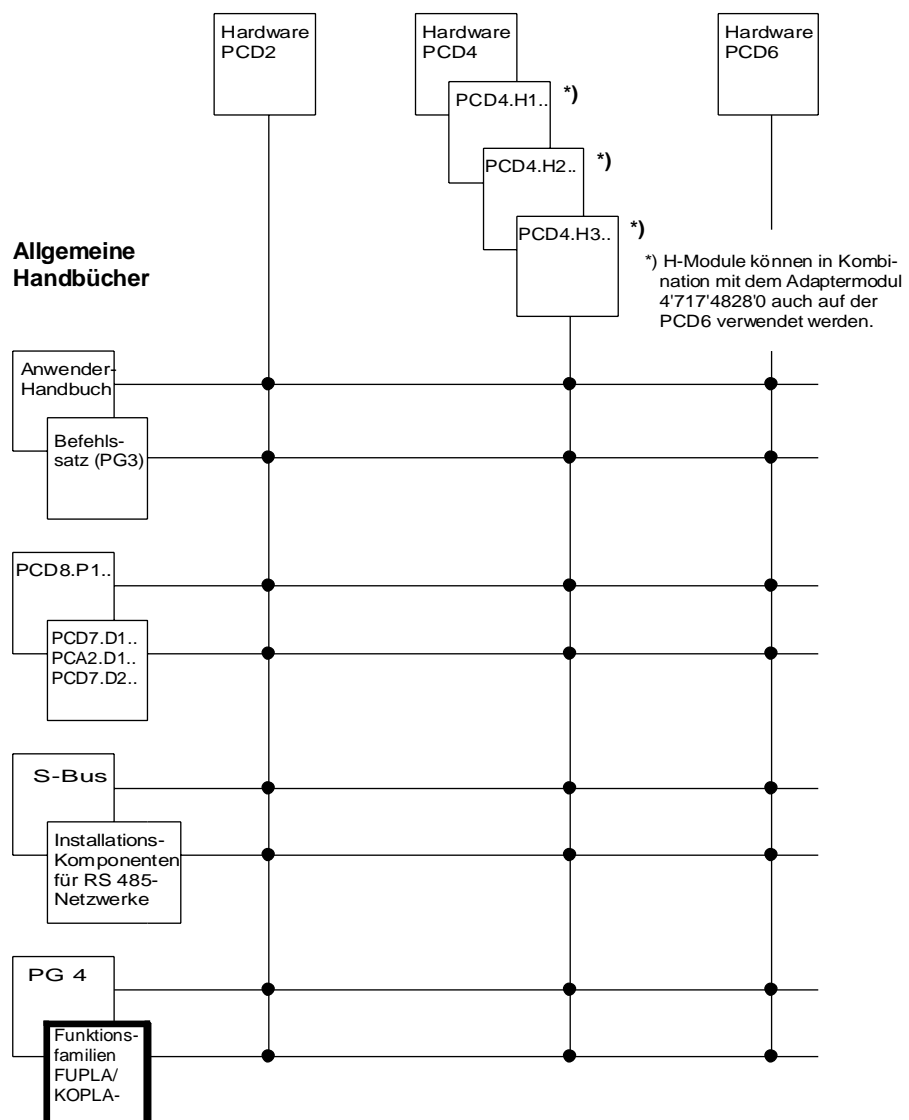
Wichtiger Hinweis:

Um den einwandfreien Betrieb von SAIA® PCD sicherstellen zu können, wurde eine Vielzahl detaillierter Handbücher geschaffen. Diese wenden sich an technisch qualifiziertes Personal, das nach Möglichkeit auch unsere Workshops erfolgreich absolviert hat.

Die vielfältigen Leistungen der SAIA® PCD treten nur dann optimal in Erscheinung, wenn alle in diesen Handbüchern aufgeführten Angaben und Richtlinien bezüglich Montage, Verkabelung, Programmierung und Inbetriebnahme genau befolgt werden.

Damit allerdings werden Sie zum grossen Kreis der begeisterten SAIA® PCD Anwendern gehören.

Übersicht



Zuverlässigkeit und Sicherheit elektronischer Steuerungen

Die Firma SAIA AG konzipiert, entwickelt und stellt ihre Produkte mit aller Sorgfalt her:

- Neuster Stand der Technik
- Einhaltung der Normen
- Zertifiziert nach ISO 9001
- Internationale Approbationen: z.B. Germanischer Lloyd, Det Norske Veritas, CE-Zeichen ...
- Auswahl qualitativ hochwertiger Bauelemente
- Kontrollen in verschiedenen Stufen der Fertigung
- In-Circuit-Tests
- Run-in (Wärmelauf bei 85°C während 48h)

Die daraus resultierende hochstehende Qualität zeigt trotz aller Sorgfalt Grenzen. So ist z.B. mit natürlichen Ausfällen von Bauelementen zu rechnen. Für diese gibt die Firma SAIA AG Garantie gemäss den "Allgemeinen Lieferbedingungen".

Der Anlagebauer seinerseits muss auch seinen Teil für das zuverlässige Arbeiten einer Anlage beitragen. So ist er dafür verantwortlich, dass die Steuerung datenkonform eingesetzt wird und keine Überbeanspruchungen, z.B. auf Temperaturbereiche, Überspannungen und Störfelder oder mechanischen Beanspruchungen auftreten.

Darüber hinaus ist der Anlagebauer auch dafür verantwortlich, dass ein fehlerhaftes Produkt in keinem Fall zu Verletzungen oder gar zum Tod von Personen bzw. zur Beschädigung oder Zerstörung von Sachen führen kann. Die einschlägigen Sicherheitsvorschriften sind in jedem Fall einzuhalten. Gefährliche Fehler müssen durch zusätzliche Massnahmen erkannt und hinsichtlich ihrer Auswirkung blockiert werden. So sind z.B. für die Sicherheit wichtige Ausgänge auf Eingänge zurückzuführen und softwaremässig zu überwachen. Es sind die Diagnoseelemente der PCD wie Watch-Dog, Ausnahme-Organisations-Blocks (XOB) sowie Test- und Diagnose-Befehle konsequent anzuwenden.

Werden alle diese Punkte berücksichtigt, verfügen Sie mit der SAIA® PCD über eine moderne und sichere programmierbare Steuerung, die Ihre Anlage über viele Jahre zuverlässig steuern, regeln und überwachen wird.

4.4 Die Funktionsfamilien des FUPLA

Übersicht über die Funktionen der einzelnen Funktionsfamilien

(nach Funktionen und Gebräuchlichkeit gegliedert *)

4.4.1 Binäre Funktionen

- 4.4.1.1 Und-Verknüpfung, 2-10 Eingänge
- 4.4.1.2 Oder-Verknüpfung 2-10 Eingänge
- 4.4.1.3 Exklusiv-Oder Verknüpfung, 2-10 Eingänge
- 4.4.1.4 Move - Bringt den Eingangsstatus zum Ausgang
- 4.4.1.5 Dynamische Ansteuerung, Flankentriggerung

- 4.4.1.6 Setze "H" (High)
- 4.4.1.7 Setze "L" (Low)
- 4.4.1.8 Abschluss offener binärer Ausgänge

- 4.4.1.9 Multiplexer mit binärer Vorwahl
- 4.4.1.10 Multiplexer mit dezimaler Vorwahl
- 4.4.1.11 Demultiplexer mit binärer Vorwahl
- 4.4.1.12 Demultiplexer mit dezimaler Vorwahl

- 4.4.1.13 Status eines Eingangs/Ausgangs
- 4.4.1.14 Status eines Flags

- 4.4.1.15 Gerade, 2-10 Eingänge
- 4.4.1.16 Ungerade, 2-10 Eingänge

4.4.2 Flip-Flop - Flip-Flops (Kippstufen)

- 4.4.2.1 Schrittschalter
- 4.4.2.2 "D"-Typ Flip-Flop

- 4.4.2.3 Dynamisches "RS"-Flip-Flop mit Priorität auf "R"
- 4.4.2.4 Dynamisches "RS"-Flip-Flop mit Priorität auf "S"

- 4.4.2.5 Getaktetes "JK"-Flip-Flop

- 4.4.2.6 Getaktetes "RS"-FF mit Priorität auf "R"
- 4.4.2.7 Getaktetes "RS"-FF mit Priorität auf "S"

- 4.4.2.8 "RS"-Flip-Flop mit Priorität auf "R"
- 4.4.2.9 "RS"-Flip-Flop mit Priorität auf "S"

*) Im FUPLA werden alle Funktionen automatisch alphabetisch geordnet

4.4.3 Zähler (Counter)

- 4.4.3.1 Aufwärtszähler mit Vorwahl
- 4.4.3.2 Abwärtszähler mit Vorwahl
- 4.4.3.3 Aufwärtszähler mit Nullstellung
- 4.4.3.4 Auf-/Abwärtszähler mit Vorwahl
- 4.4.3.5 Auf-/Abwärtszähler mit Vorwahl und Nullstellung

4.4.4 Zeitfunktionen (Timer)

- 4.4.4.1 Anzugverzögerung
- 4.4.4.2 Anzugverzögerung mit Selbsthaltung
- 4.4.4.3 Anzugwisch-Funktion

- 4.4.4.4 Abfallverzögerung
- 4.4.4.5 Anzug- und Abfallverzögerung
- 4.4.4.6 Abfallverzögerung mit Rückstellung
- 4.4.4.7 Impulsfunktion

- 4.4.4.8 Stoppuhr (Chrono)
- 4.4.4.9 Hardwareuhr der PCD

- 4.4.4.10 Startverzögerung

4.4.5 Blinker

- 4.4.5.1 Blinker symmetrisch
- 4.4.5.2 Blinker asymmetrisch

- 4.4.5.3 Abtastsignal

4.4.6 Arithmetik im Ganzzahlformat

- 4.4.6.1 Addition
- 4.4.6.2 Subtraktion
- 4.4.6.3 Multiplikation
- 4.4.6.4 Division
- 4.4.6.5 Quadratwurzel
- 4.4.6.6 Durchschnitt

- 4.4.6.7 Konstante
- 4.4.6.8 Absolutwert

- 4.4.6.9 Und-Verknüpfung von Registern
- 4.4.6.10 Oder-Verknüpfung von Registern
- 4.4.6.11 Exklusiv-Oder-Verknüpfung von Registern
- 4.4.6.12 Komplementieren eines Register

- 4.4.6.13 Vergleich auf 2 gleiche Werte
- 4.4.6.14 Vergleich auf grösseren oder gleichen Wert
- 4.4.6.15 Vergleich auf grösseren Wert
- 4.4.6.16 Vergleich auf kleineren oder gleichen Wert
- 4.4.6.17 Vergleich auf kleineren Wert

- 4.4.6.18 Wert = Null
- 4.4.6.19 Wert zwischen Grenzen
- 4.4.6.20 Grösster Wert wird ausgegeben
- 4.4.6.21 Kleinster Wert wird ausgegeben

- 4.4.6.22 Move - Bringt den Eingangswert zum Ausgang
- 4.4.6.23 Move mit Enable
- 4.4.6.24 Move mit Speicher
- 4.4.6.25 Durchschaltung vordefinierter Werte

- 4.4.6.26 Multiplexer mit binärer Vorwahl
- 4.4.6.27 Multiplexer mit dezimaler Vorwahl
- 4.4.6.28 Demultiplexer mit binärer Vorwahl
- 4.4.6.29 Demultiplexer mit dezimaler Vorwahl

- 4.4.6.30 Schiebe Registerinhalt nach links
- 4.4.6.31 Schiebe Registerinhalt nach rechts
- 4.4.6.32 Rotiere Registerinhalt nach links
- 4.4.6.33 Rotiere Registerinhalt nach rechts

- 4.4.6.34 Register Indikation
- 4.4.6.35 Timer/Counter Indikation

- 4.4.6.36 Nicht angeschlossen

4.4.7 Arithmetik im Fließpunktformat

- 4.4.7.1 Addition
- 4.4.7.2 Subtraktion
- 4.4.7.3 Multiplikation
- 4.4.7.4 Division
- 4.4.7.5 Quadratwurzel
- 4.4.7.6 Durchschnitt

- 4.4.7.7 Konstante
- 4.4.7.8 Absolutwert

- 4.4.7.9 Sinus
- 4.4.7.10 Cosinus
- 4.4.7.11 Arcus Tangens

- 4.4.7.12 Natürlicher Exponent - Exponentialfunktion
- 4.4.7.13 Natürlicher Logarithmus

- 4.4.7.14 Vergleich auf 2 gleiche Werte
- 4.4.7.15 Vergleich auf grösseren oder gleichen Wert
- 4.4.7.16 Vergleich auf grösseren Wert
- 4.4.7.17 Vergleich auf kleineren oder gleichen Wert
- 4.4.7.18 Vergleich auf kleineren Wert

- 4.4.7.19 Wert gleich Null
- 4.4.7.20 Lomit - Wert zwischen Grenzen
- 4.4.7.21 Maximum - Grösster Wert wird ausgegeben
- 4.4.7.22 Minimum - Kleinster wird ausgegeben

- 4.4.7.23 Move - Bringt den Eingangswert zum Ausgang
- 4.4.7.24 Move mit Enable
- 4.4.7.25 Move mit Speicher
- 4.4.7.26 Schalter - Durchschaltung vordefinierter Werte

- 4.4.7.27 Multiplexer mit binärer Vorwahl
- 4.4.7.28 Multiplexer mit dezimaler Vorwahl
- 4.4.7.29 Demultiplexer mit binärer Vorwahl
- 4.4.7.30 Demultiplexer mit dezimaler Vorwahl

- 4.4.7.31 Nicht angeschlossen - Abschluss offener Ausgänge

4.4.8 Wandler (Binär - Dezimal - Fließpunkt)

- 4.4.8.1 Binär zu dezimal allgemein, ab 1-8 Einzelementen
- 4.4.8.2 Binär zu dezimal allgemein, ab 1-24 Einzelementen
- 4.4.8.3 Binär zu dezimal, ab Elementreihe (PCD-Format)
- 4.4.8.4 Binär zu dezimal invers, ab Elementreihe (PCA-Format)

- 4.4.8.5 Dezimal zu binär allgemein, auf 1-8 Einzelemente
- 4.4.8.6 Dezimal zu binär allgemein, auf 1-24 Einzelemente
- 4.4.8.7 Dezimal zu binär, auf Elementreihe (PCD-Format)
- 4.4.8.8 Dezimal zu binär invers, auf Elementreihe (PCA-Format)

- 4.4.8.9 BCD zu dezimal allgemein, ab Einzelementen
- 4.4.8.10 BCD zu dezimal, ab Elementreihe (PCD-Format)
- 4.4.8.11 BCD zu dezimal invers, ab Elementreihe (PCA-Format)

- 4.4.8.12 Dezimal zu BCD allgemein, auf Einzelemente
- 4.4.8.13 Dezimal zu BCD, auf Elementreihe (PCD-Format)
- 4.4.8.14 Dezimal zu BCD invers, auf Elementreihe (PCA-Format)

- 4.4.8.15 1 Bit zu dezimal
- 4.4.8.16 1 Bit zu dezimal LSB

- 4.4.8.17 Dezimal zu 1 Bit
- 4.4.8.18 Dezimal LSB zu 1 Bit

- 4.4.8.19 Fließpunkt- zu Ganzzahlformat
- 4.4.8.20 Ganzzahl- zu Fließpunktformat

4.4.9 Indirekte Adressierung

- 4.4.9.1 Kopieren auf Ausgänge
- 4.4.9.2 Lesen von Eingängen
- 4.4.9.3 Kopieren auf Flags
- 4.4.9.4 Lesen von Flags
- 4.4.9.5 Kopieren von Registern numerisch
- 4.4.9.6 Lesen von Registern numerisch
- 4.4.9.7 Kopieren von Registern fließpunkt
- 4.4.9.8 Lesen von Registern fließpunkt
- 4.4.9.9 Kopieren auf Timer/Counter
- 4.4.9.10 Lesen von Timer/Counter
- 4.4.9.11 Timer
- 4.4.9.12 Counter
- 4.4.9.13 Lesen der logischen Zustände von Timer/Counter

4.4.10 Move Data

- 4.4.10.1 Move-In Bit
- 4.4.10.2 Move-Out Bit
- 4.4.10.3 Move-In Nibble (4 Bit binär)
- 4.4.10.4 Move-Out Nibble (4 Bit binär)
- 4.4.10.5 Move-In Digit (4 Bit BCD)
- 4.4.10.6 Move-Out Digit (4 Bit BCD)
- 4.4.10.7 Move-In Byte (8 Bit)
- 4.4.10.8 Move-Out Byte (8 Bit)
- 4.4.10.9 Move-In Word (16 Bit)
- 4.4.10.10 Move-Out Word (16 Bit)

4.4.11 Anzeigen (Displays)

- 4.4.11.1 Displaymodul PCA2.D12
- 4.4.11.2 Displaymodul PCA2.D14
- 4.4.11.3 PCD2.F510 für numerische Anzeigen
- 4.4.11.4 PCD2.F510 für Text-Anzeigen

4.4.12 GRAFTEC-Funktionen

- 4.4.12.1 Lade Timer ohne Bedingung
- 4.4.12.2 Lade Timer bedingt
- 4.4.12.3 Lade Zähler ohne Bedingung
- 4.4.12.4 Lade Zähler bedingt
- 4.4.12.5 Inkrementiere Zähler
- 4.4.12.6 Dekrementiere Zähler
- 4.4.12.7 Ist Timer abgelaufen ?
- 4.4.12.8 Ist Zähler = 0 ?
- 4.4.12.9 Ende Transition
- 4.4.12.10 Timer setzen und Zeit abwarten
- 4.4.12.11 Counter setzen und warten bis Counter = 0

4.4.13 Spezialfunktionen (Watch-Dog)

- 4.4.13.1 Watch Dog (dauernd aktiv)
- 4.4.13.2 Watch Dog ein-/ausschaltbar (Enable)

4.4.14 Analog Module

- 4.4.14.1 PCD2.W1 Analoges Eingangsmodul PCD2.W1 (12 Bit)
- 4.4.14.2 PCD2.W2 Analoges Eingangsmodul PCD2.W2 (10 Bit)
- 4.4.14.3 PCD2.W4 Analoges Ausgangsmodul PCD2.W4 (8 Bit)
- 4.4.14.4 PCD2.W5 Analoges Ein-/Ausgangsm PCD2.W5 (12 Bit)

- 4.4.14.5 PCD4.W1 Analoges Ein-/Ausgangsm. PCD4.W1 (12 Bit)
- 4.4.14.6 PCD4.W3 Analoges Eingangsmodul PCD4.W3 (12 Bit + Vz)
- 4.4.14.7 PCD4.W4 Analoges Ausgangsmodul PCD4.W4 (8 Bit)

- 4.4.14.8 PCD6.W1 Analoges Ein-/Ausgangsm. PCD6.W1 (12 Bit)
- 4.4.14.9 PCD6.W3 Analoges Eingangsmodul PCD6.W3 (12 Bit + Vz)
- 4.4.14.10 PCD6.W4 Analoges Ausgangsm. PCD6.W4 (8 Bit)

4.4.15 Regelung

- 4.4.15.1 PID-Regelung

4.4.16 Benutzerdefinierte Funktionen

- 4.4.16.1 Aufruf PB
- 4.4.16.2 Aufruf FB
- 4.4.16.3 Aufruf SB (GRAFTEC)

- 4.4.16.4 Durch Anwender definierten Block Nr. 1
- 4.4.16.5 Durch Anwender definierten Block Nr. 2
- 4.4.16.6 Durch Anwender definierten Block Nr. 3
- 4.4.16.7 Durch Anwender definierten Block Nr. 4
- 4.4.16.8 Durch Anwender definierten Block Nr. 5

4.4.17 Serielle Kommunikation (Datenaustausch)

- 4.4.17.1 Schnittstellen-Parameter
- 4.4.17.2 Schnittstellen-Parameter für externe Dateien

- 4.4.17.3 Empfangen von 1 - 20 I/O/F
- 4.4.17.4 Empfangen von 1 - 20 R/T/C/Uhr

- 4.4.17.5 Senden von 1 - 20 I/O/F
- 4.4.17.6 Senden von 1 - 20 R/T/C/Uhr

- 4.4.17.7 Empfang I/O/F von mehreren Stationen
- 4.4.17.8 Empfang R/T/C von mehreren Stationen

Erläuterung zur Schreibweise und den verwendeten Symbolen bei den Beschreibungen der einzelnen Funktionen:

Name der Funktion

[_fupxyz]



Name der Funktion

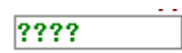
Name der Funktion aus dem FUPLA-Menü. (aus der 'FBox Selection...')

Links an der FBox sind die Eingänge, z.B:

| | | | | | |
|-----------|-------|---|--------|-----|--------------------------|
| Eingänge: | - En | → | Enable | "-" | binärer Eingang (H/L) |
| | > Set | → | Setze | ">" | dynamischer bin. Eingang |
| | = Val | → | Wert | "=" | numerischer Eing. (Wert) |

Rechts an der FBox sind die Ausgänge, z.B:

| | | | | | |
|-----------|------|---|--------------|-----|--------------------------|
| Ausgänge: | - Q | → | Ausgang | "-" | binärer Ausgang (H/L) |
| | - /Q | → | Ausgang inv. | "-" | binärer Ausgang (H/L) |
| | = R | → | Resultat | "=" | numerischer Ausg. (Wert) |

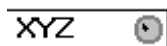
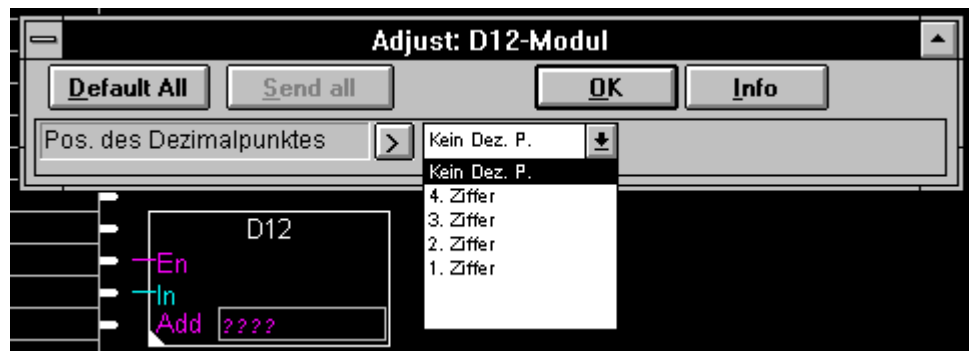


Feld für die Basisadresse der Funktion. Es ist normalerweise ein "O" oder "o" (Output) einzutragen, z.B. "o 16".



Funktionen welche ein Einstellfenster haben, sind links unten mit einem ausgefüllten Dreieck bezeichnet. Das Einstellfenster wird durch einen Doppelklick auf die FBox geöffnet. Das Pfeilsymbol muss dazu gewählt sein. Nach dem Plazieren einer FBox kann das Pfeilsymbol durch das Betätigen der rechten Maustaste direkt gewählt werden.

Beispiel für ein Einstellfenster: 'Anzeigen' - 'D12-Modul'



Komplexere Funktionen, wie z.B. die Kommunikation, enthalten in der FBox eine LED. Diese ist beim Programmieren grau. In RUN (online), falls alles ok ist, wird die LED grün. Im Fehlerfall leuchtet die LED rot und ein "A" erscheint. (Dies damit der Status auch auf einem monochromen Bildschirm erkannt werden kann).

[fupabc]

(in der Titelzeile jeder Funktions-Familie) bezeichnet die Dateien dieser Funktions-Familie, z.B. [sfuptime] die Familie der Zeitfunktionen: "sfuptime.def", "sfuptime.hlp", "sfuptime.idx", "sfuptime.lib".

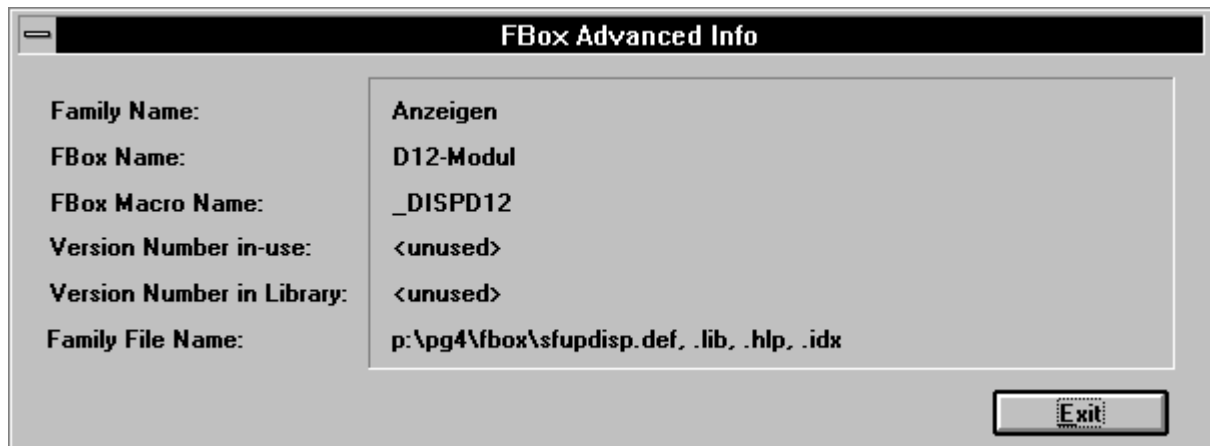
[_fupxyz]:

(FBox Macro Name) Interne Referenz in den einzelnen Dateien, z.B. [_ondel] für die Anzugverzögerung. Die Bezeichnungen sind für alle Sprachversionen identisch und dienen so quasi als Querverweis zwischen den Bezeichnungen der eigentlichen Funktion und den FBoxen in den verschiedenen Sprachen.

Für den Anwender haben diese Namen im Normalfall keine Bedeutung. Bei einem "Update", oder einer nachträglichen Korrektur, ist diese Referenz ein unerlässliches Hilfsmittel, da bei der Vielzahl von ähnlichen Funktionen die Übersicht sehr rasch verloren geht.

Der aktuelle interne Dateiname jeder FBox kann im Info-Fenster durch anklicken von 'Advanced Info' eingesehen werden.

Beispiel für die "Advanced Info" der FBox "D-12 Modul"



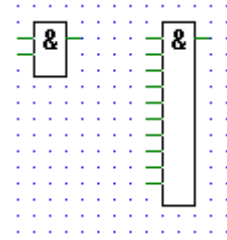
4.4.1 Binäre - Funktionen

[sfupbina]

4.4.1.1 Und-Verknüpfung, 2-10 Eingänge



[_band]



Alle Ein- und Ausgänge: – → Binärformat

Am Ausgang erscheint die binäre UND-Verknüpfung der Eingänge. Der Ausgang wird = H, wenn alle Eingänge = H sind.

Ausziehbar von 2 auf 10 Eingänge.

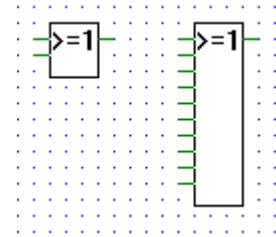
Beispiel für 2 Eingänge In1 und In2:

| In1 | In2 | Out |
|-----|-----|-----|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

4.4.1.2 Oder-Verknüpfung, 2-10 Eingänge



[_bor]



Alle Ein- und Ausgänge: – → Binärformat

Am Ausgang erscheint die binäre ODER-Verknüpfung der Eingänge. Der Ausgang wird = H, wenn zumindest 1 Eingang = H ist.

Ausziehbar von 2 auf 10 Eingänge.

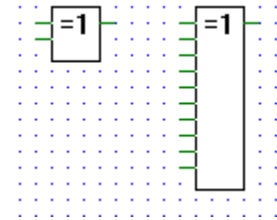
Beispiel für 2 Eingänge In1 und In2:

| In1 | In2 | Out |
|-----|-----|-----|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

4.4.1.3 Exklusiv-Oder Verknüpfung, 2-10 Eingänge



[_bxor]



Alle Ein- und Ausgänge: – → Binärformat

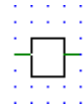
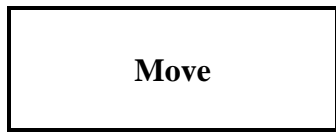
Am Ausgang erscheint die binäre XOR-Verknüpfung der Eingänge. Der Ausgang wird = H, wenn nur 1 Eingang = H ist.

Ausziehbar von 2 auf 10 Eingänge.

Beispiel für 2 Eingänge In1 und In2:

| In1 | In2 | Output |
|-----|-----|--------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

4.4.1.4 Bringt den Eingangsstatus zum Ausgang



[_bmove]

Alle Ein- und Ausgänge: – → Binärformat

Am Ausgang erscheint der logische Status des Eingangs.

Dient zum direkten Durchschalten eines PCD-Eingangs zu einem PCD-Ausgang.

4.4.1.5 Dynamische Ansteuerung, Flankentriggerung



[_bdyn2]

Alle Ein- und Ausgänge: – → Binärformat

Der Ausgang wird nur = H, wenn der Eingang von L auf H geht.

Im Einstellfenster können 3 Optionen für das Verhalten der Funktion beim Einschalten der Steuerung bzw. beim Programmstart gewählt werden:



"Immer": Ein H am Eingang wird immer als ansteigende Flanke erkannt.

"Niemals": Ein H am Eingang wird nie als ansteigende Flanke erkannt.

"Ein=0/Aus=1": Eine ansteigende Flanke wird nur dann erkannt, wenn der Eingang beim Ausschalten der Steuerung = L und beim Wiedereinschalten = H ist.

Achtung: Die letztgenannte Option kann undefinierte Reaktionen zeigen, wenn nach einer Programmänderung die Flags nicht zurückgesetzt wurden!

4.4.1.6 Setze "H" (high)

[_bhigh]

Ausgang:– → Binärformat.

Der Ausgang der Funktion ist immer = H.

Dient z.B. zum dauernden Anlegen eines H-Signals an einen Eingang.**4.4.1.7 Setze "L" (Low)**

[_blow]

Ausgang:– → Binärformat.

Der Ausgang der Funktion ist immer = L.

Dient z.B. zum dauernden Anlegen eines L-Signals an einen Eingang.**4.4.1.8 Abschluss offener binärer Ausgänge**

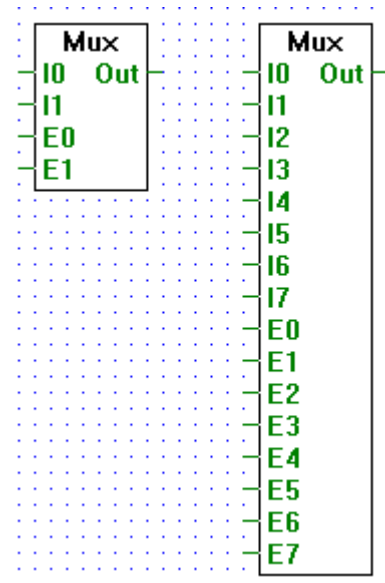
[_bnotcn]

Die "Nicht angeschlossen"-Funktion dient zum Abschluss nicht verwendeter binärer Ausgänge.

4.4.1.9 Multiplexer mit binärer Vorwahl



[_bmux]



Alle Ein- und Ausgänge: – → Binärformat

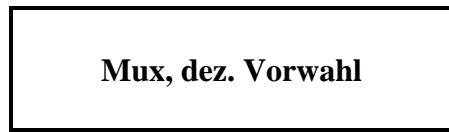
Leitet den binären Status am Eingang "Ix" zum Ausgang "OUT" wenn der entsprechende binäre Enable-Eingang "Ex" = H ist. (x = 0 - 7)

Ausziehbar von 2 auf 8 Eingänge.

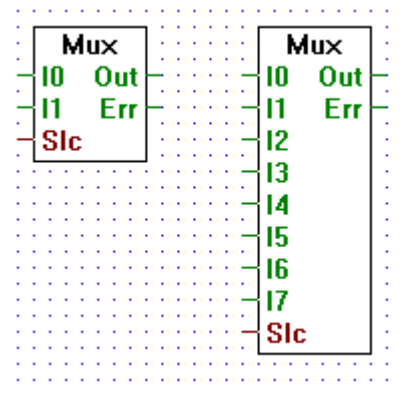
Sind alle Enable-Eingänge "Ex" = L, wird der Ausgang "OUT" = L.

Der Enable-Eingang "E0" hat die höchste, "E7" die niedrigste Priorität. Sind also 2 oder mehr Enable-Eingänge gleichzeitig = H, ist der Eingang mit der höchsten Priorität bestimmend.

4.4.1.10 Multiplexer mit dezimaler Vorwahl



[_bmux2]



- Eingänge:
- I0:
 - | Binärformat
 - I7:
 - = Slc Ganzzahlformat (Slc = Select)
- Ausgänge:
- Out Binärformat
 - Err Binärformat

Leitet den binären Status des Eingang "Ix" zum Ausgang "OUT". Die Wahl des Eingangs erfolgt durch Anlegen eines ganzzahligen Wertes an den Eingang "Slc". Dieser Wert entspricht der Nummer des zu wählenden Eingangs. Wird z.B. 2 an den Eingang "Slc" angelegt, wird der binäre Eingang "I2" zum Ausgang "OUT" geleitet.

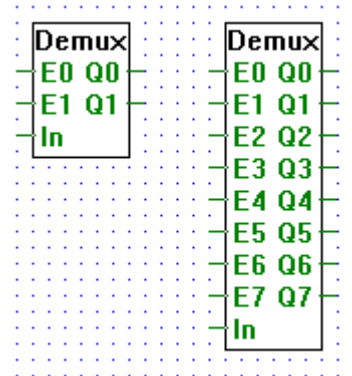
Ausziehbar von 2 auf 8 Eingänge.

Der Ausgang "Err" (Error) wird = H, wenn ein unmöglicher Wert am Eingang "Slc" angelegt wird.

4.4.1.11 Demultiplexer mit binärer Vorwahl



[_bdemux]



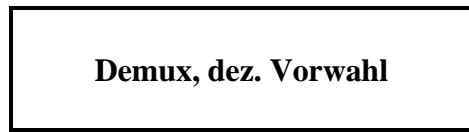
Alle Ein- und Ausgänge: – → Binärformat

Überträgt den binären Eingang "In" zu einem Ausgang "Qx" wenn der entsprechende Eingangskanal "Ex" = H ist. (x = 0 - 7)

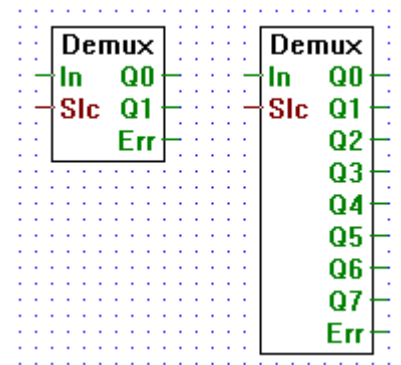
Ausziehbar von 2 auf 8 Eingänge.

Ist ein Eingangskanal "Ex" = L, wird der entsprechende Ausgang "Qx" = L gesetzt.

4.4.1.12 Demultiplexer mit dezimaler Vorwahl



[_bdemux2]



Eingänge: – In: Binärformat
 = Slc: Ganzzahlformat (Slc = Select)

Ausgänge: – Q0
 | Binärformat
 – Q7
 – Err: Binärformat

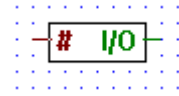
Überträgt den binären Eingang "In" zu einem Ausgang "Qx". Die Ausgänge werden mit dem Anlegen eines ganzzahligen Wertes an den Eingang "Slc" gewählt. Dieser Wert entspricht der Nummer des zu wählenden Ausganges. Wird z.B. 2 an den Eingang "Slc" angelegt, wird der binäre Eingang "In" an den Ausgang "Q2" geleitet.

Alle nicht gewählten Ausgänge sind = L.

Ausziehbar von 2 auf 8 Eingänge.

Der Ausgang "Err" (Error) wird = H, wenn ein unmöglicher Wert am Eingang "Slc" angelegt wird.

4.4.1.13 Status eines Eingangs/Ausgangs



[_bioind]

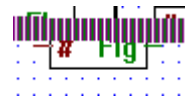
Eingang: = I/O Adresse: Ganzzahlformat

Ausgang: – I/O state: Binärformat

Am Ausgang erscheint der Status des Eingangs oder Ausgangs Nr. #.

Beispiel: Ist # = 5, erscheint am Ausgang der Status des E/A Nr. 5

4.4.1.14 Status eines Flags



[_bflgin]

Eingang: = Flag Adresse: Ganzzahlformat

Ausgang: – Flag state: Binärformat

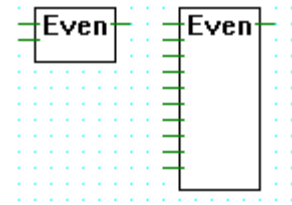
Am Ausgang erscheint der Status des Flags Nr. #.

Beispiel: Ist # = 5, erscheint am Ausgang der Status des Flags Nr. 5

4.4.1.15

Gerade, 2-10 Eingänge

[_beven]



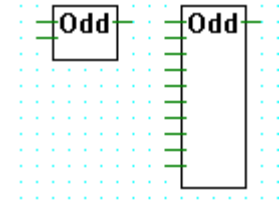
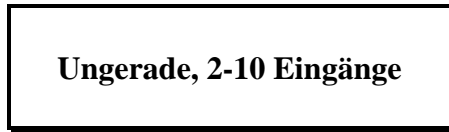
Eingänge: – Binärformat
 Ausgang: – Binärformat

Der Ausgang ist = H, wenn die Anzahl Eingänge welche = H sind gerade ist. Andernfalls wird der Ausgang = L.

z.B. für 3 Eingänge:

| In1 | In2 | In3 | Out |
|-----|-----|-----|-----|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

4.4.1.16 Ungerade, 2-10 Eingänge



[_bodd]

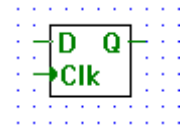
Eingänge: – Binärformat
 Ausgang: – Binärformat

Der Ausgang ist = H, wenn die Anzahl Eingänge welche = H sind ungerade ist. Andernfalls wird der Ausgang = L.

z.B. für 3 Eingänge:

| In1 | In2 | In3 | Out |
|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

4.4.2.2 "D"-Typ Flip-Flop



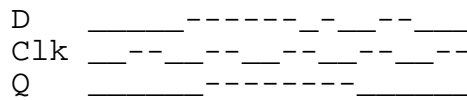
[_flid]

Alle Ein- und Ausgänge: – → Binärformat.

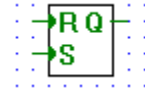
"D"-Flip-Flop mit Flankentriggerung am "Clk"-Eingang (Clock).

Der Status des "D"-Eingangs wird beim Uebergang von L zu H am Eingang "Clk" gespeichert.

Der logische Zustand des Ausgangs "Q" ist immer der zuletzt gespeicherte Status von "D".



4.4.2.3 **Dynamisches "RS"-FF mit Priorität auf "R"**



[_flirs]

Alle Ein- und Ausgänge: – → Binärformat.

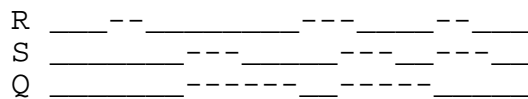
"RS"-Flip-Flop mit Priorität auf dem "R"-Eingang. Beide Eingänge "R" und "S" sind flankengetriggert.

Geht Eingang "R" von L zu H, wird Ausgang Q = L gesetzt.

Geht Eingang "S" von L zu H, wird Ausgang Q = H gesetzt.

Gehen beide Eingänge "R" und "S" zur gleichen Zeit von L zu H, wird der Ausgang "Q" = L gesetzt.

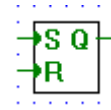
In allen anderen Fällen bleibt der Ausgang unverändert.



| R | S | Q |
|-----|-----|-------------|
| X | X | Unverändert |
| X | _/- | 1 |
| _/- | X | 0 |
| _/- | _/- | 0 |

_/- = ansteigende Flanke
X = beliebiger Status

4.4.2.4 **Dynamisches "RS"-FF mit Priorität auf "S"**



[_flisr]

Alle Ein- und Ausgänge: – → Binärformat.

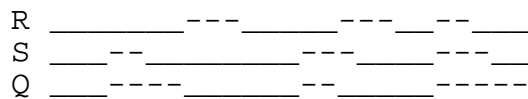
"RS"-Flip-Flop mit Priorität auf dem "S"-Eingang. Beide Eingänge "R" und "S" sind flankengetriggert.

Geht Eingang "R" von L zu H, wird Ausgang Q = L gesetzt.

Geht Eingang "S" von L zu H, wird Ausgang Q = H gesetzt.

Gehen beide Eingänge "R" und "S" zur gleichen Zeit von L zu H, wird der Ausgang "Q" = H gesetzt.

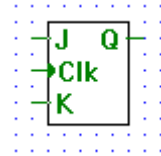
In allen anderen Fällen bleibt der Ausgang unverändert



| R | S | Q |
|-----|-----|-------------|
| X | X | Unverändert |
| X | _/- | 1 |
| _/- | X | 0 |
| _/- | _/- | 1 |

_/- = ansteigende Flanke
X = beliebiger Status

4.4.2.5 Getaktetes "JK"-Flip-Flop



[_fijk]

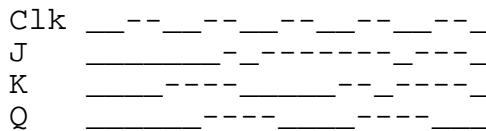
Alle Ein- und Ausgänge: – → Binärformat.

Getaktetes "JK"-Flip-Flop.

Die Eingänge "J" und "K" werden nur gelesen, wenn am Eingang "Clk" eine ansteigende Flanke erkannt wird. Wird keine Flanke erkannt, bleibt der Ausgang "Q" unverändert.

Bei einer ansteigenden Flanke am Eingang "Clk":

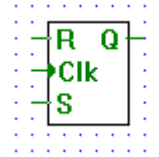
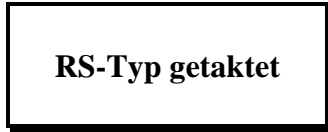
- und "J" ist = H, wird "Q" = L,
- und "K" ist = H, wird "Q" = H,
- und "J" und "K" sind = H, wird "Q" invertiert (toggle),
- und "J" und "K" sind = L, bleibt "Q" unverändert.



| Clk | J | K | Q(t) |
|-----|---|---|------------------|
| X | X | X | Q(t-1) |
| _/- | 0 | 0 | Q(t-1) |
| _/- | 1 | 0 | 0 |
| _/- | 1 | 1 | /Q(t-1) (toggle) |

_/- = ansteigende Flanke
 X = beliebiger Status

4.4.2.6 Getaktetes "RS"-FF mit Priorität auf "R"



[_flirclk]

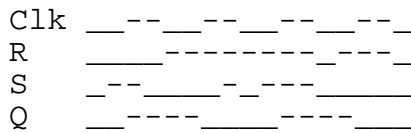
Alle Ein- und Ausgänge: – → Binärformat.

Getakteter "RS"-Flip-Flop mit Priorität auf dem "S"-Eingang.

Die Eingänge "R" und "S" werden bei jeder ansteigenden Flanke am Eingang "Clk" gelesen. Wird keine ansteigende Flanke erkannt, bleibt der Ausgang "Q" unverändert.

Bei einer ansteigenden Flanke am Eingang "Clk":

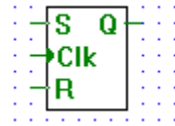
- und "R" ist = H, wird "Q" = L,
- und "S" ist = H, wird "Q" = H,
- und "R" und "S" sind = H, wird "Q" = H,
- und "R" und "S" sind = L, bleibt "Q" unverändert.



| Clk | R | S | Q |
|-----|---|---|-------------|
| X | X | X | Unverändert |
| _/- | 0 | 0 | Unverändert |
| _/- | 0 | 1 | 1 |
| _/- | 1 | 0 | 0 |
| _/- | 1 | 1 | 1 |

_/ - = ansteigende Flanke
 X = beliebiger Status

4.4.2.7 Getaktetes "RS"-FF mit Priorität auf "S"



[_flisrclk]

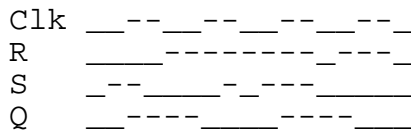
Alle Ein- und Ausgänge: – → Binärformat.

Getakteter "RS"-Flip-Flop mit Priorität auf dem "S"-Eingang.

Die Eingänge "R" und "S" werden bei jeder ansteigenden Flanke am Eingang "Clk" gelesen. Wird keine ansteigende Flanke erkannt, bleibt der Ausgang "Q" unverändert.

Bei einer ansteigenden Flanke am Eingang "Clk":

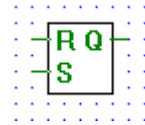
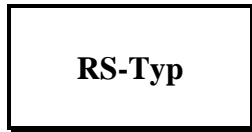
- und "R" ist = H, wird "Q" = L,
- und "S" ist = H, wird "Q" = H,
- und "R" und "S" sind = H, wird "Q" = L,
- und "R" und "S" sind = L, bleibt "Q" unverändert.



| Clk | R | S | Q |
|------|---|---|-------------|
| X | X | X | Unverändert |
| _-/- | 0 | 0 | Unverändert |
| _-/- | 0 | 1 | 1 |
| _-/- | 1 | 0 | 0 |
| _-/- | 1 | 1 | 1 |

_-/- = ansteigende Flanke
 X = beliebiger Status

4.4.2.8 "RS"-Flip-Flop mit Priorität auf "R"



[_flirsndyn]

Alle Ein- und Ausgänge: – → Binärformat.

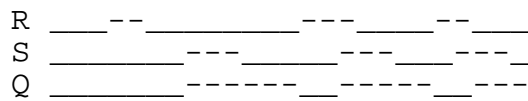
RS Flip Flop mit Priorität des "R"-Eingangs.

Wenn Eingang "R" H und Eingang "S" L ist, dann wird der Ausgang "Q" auf L gesetzt.

Wenn der Eingang "S" H ist, wird der Ausgang "Q" auf H gesetzt.

Wenn beide Eingänge "R" und "S" H sind, wird der Ausgang "Q" auf L gesetzt.

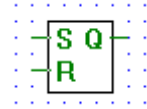
In allen anderen Fällen wird der Ausgang nicht verändert.



| R | S | Q |
|---|---|-------------|
| 0 | 0 | Unverändert |
| 1 | X | 0 |
| 0 | 1 | 1 |

X = beliebiger Status.

4.4.2.9 "RS"-Flip-Flop mit Priorität auf "S"



[_flisrndyn]

Alle Ein- und Ausgänge: – → Binärformat.

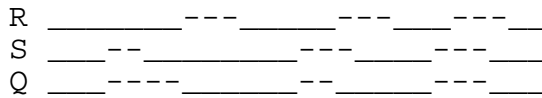
RS Flip Flop mit Priorität des "S"-Eingangs.

Wenn Eingang "R" H und Eingang "S" L ist, dann wird der Ausgang "Q" auf L gesetzt.

Wenn der Eingang "S" H ist, wird der Ausgang "Q" auf H gesetzt.

Wenn beide Eingänge "R" und "S" H sind, wird der Ausgang "Q" auf H gesetzt.

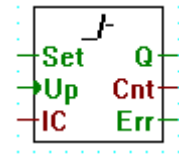
In allen anderen Fällen wird der Ausgang nicht verändert.



| R | S | Q |
|---|---|-------------|
| 0 | 0 | Unverändert |
| X | 1 | 1 |
| 1 | 0 | 0 |

X = beliebiger Status

Notizen:

4.4.3 Zähler (Counter)**[sfupcoun]****4.4.3.1 Aufwärtszähler mit Vorwahl**

[_uppr2]

Eingänge: – Set: Setzen Binärformat
 > Up: +1 Binärformat
 = IC: Ladewert Ganzzahlformat

Ausgänge: – Q: pos/neg Binärformat
 = Cnt: Zählwert Ganzzahlformat
 – Err: Error Binärformat

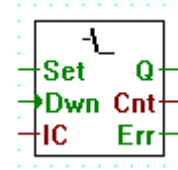
Inkrementiert einen Zähler bei jeder positiven Flanke am Eingang "Up". Der Zählerstand wird am Ausgang "Cnt" ausgegeben.

Wird der Eingang "Set" = H, wird der Zähler mit dem Wert am Eingang "IC" (Initial Count) geladen.

Ist der Zählerinhalt > 0 , wird der Ausgang "Q" = H. Ist der Zählerwert $<$ oder $= 0$, wird der Ausgang "Q" = L.

Bei einem Ueberlauf des Zählers wird der Ausgang "Err" = H (Error).

4.4.3.2 Abwärtszähler mit Vorwahl



[_dwnpr2]

Eingänge: – Set: Setzen Binärformat
 > Dwn: -1 Binärformat
 = IC: Ladewert Ganzzahlformat

Ausgänge: – Q: pos/neg Binärformat
 = Cnt: Zählwert Ganzzahlformat
 – Err: Error Binärformat

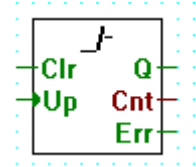
Dekrementiert einen Zähler bei jeder positiven Flanke am Eingang "Dwn". Der Zählerstand wird am Ausgang "Cnt" ausgegeben.

Wird der Eingang "Set" = H, wird der Zähler mit dem Wert am Eingang "IC" (Initial Count) geladen.

Ist der Zählerinhalt > 0 , wird der Ausgang "Q" = H. Ist der Zählerwert $<$ oder $= 0$, wird der Ausgang "Q" = L.

Bei einem Ueberlauf des Zählers wird der Ausgang "Err" = H (Error).

4.4.3.3 Aufwärtszähler mit Nullstellung



[_up2]

| | | |
|-----------|-----------------|----------------|
| Eingänge: | – Clr: Clear | Binärformat |
| | > Up: +1 | Binärformat |
| Ausgänge: | – Q: pos/neg | Binärformat |
| | = Cnt: Zählwert | Ganzzahlformat |
| | – Err: Error | Binärformat |

Inkrementiert einen Zähler bei jeder positiven Flanke am Eingang "Up". Der Zählerstand wird am Ausgang "Cnt" ausgegeben.

Ein H-Signal am Eingang "Clr" (Clear) lädt den Zähler mit Null.

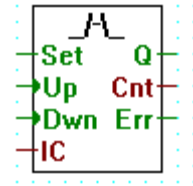
Ist der Zählerinhalt > 0 , wird der Ausgang "Q" = H. Ist der Zählerwert $<$ oder $= 0$, wird der Ausgang "Q" = L.

Bei einem Ueberlauf des Zählers wird der Ausgang "Err" = H (Error).

4.4.3.4 Auf-/Abwärtszähler mit Vorwahl



[_updnpr2]



| | | |
|-----------|--|---|
| Eingänge: | <ul style="list-style-type: none"> – Set: Setzen > Up: +1 > Dwn: -1 = IC: Ladewert | <ul style="list-style-type: none"> Binärformat Binärformat Binärformat Ganzzahlformat |
| Ausgänge: | <ul style="list-style-type: none"> – Q: pos/neg = Cnt: Zählwert – Err: Error | <ul style="list-style-type: none"> Binärformat Ganzzahlformat Binärformat |

Inkrementiert einen Zähler bei jeder positiven Flanke am Eingang "Up" und dekrementiert den Zähler bei jeder positiven Flanke am Eingang "Dwn". Der Zählerstand wird am Ausgang "Cnt" ausgegeben.

Wird der Eingang "Set" = H, wird der Zähler mit dem Wert am Eingang "IC" (Initial Count) geladen.

Ist der Zählerinhalt > 0 , wird der Ausgang "Q" = H. Ist der Zählerwert $<$ oder $= 0$, wird der Ausgang "Q" = L.

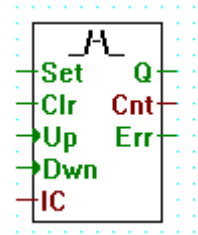
Bei einem Ueberlauf des Zählers, im positiven und im negativen Bereich, wird der Ausgang "Err" = H (Error).

4.4.3.5

Auf-/Abwärtszähler mit Vorwahl und Nullstellung



[_updncs]



| | | | |
|-----------|--------|----------|----------------|
| Eingänge: | – Set: | Setzen | Binärformat |
| | – Clr: | Clear | Binärformat |
| | > Up: | +1 | Binärformat |
| | > Dwn: | -1 | Binärformat |
| | = IC: | Ladewert | Ganzzahlformat |
| Ausgänge: | – Q: | pos/neg | Binärformat |
| | = Cnt: | Zählwert | Ganzzahlformat |
| | – Err: | Error | Binärformat |

Inkrementiert einen Zähler bei jeder positiven Flanke am Eingang "Up" und dekrementiert den Zähler bei jeder positiven Flanke am Eingang "Dwn". Der Zählerstand wird am Ausgang "Cnt" ausgegeben.

Wird der Eingang "Set" = H, wird der Zähler mit dem Wert am Eingang "IC" (Initial Count) geladen.

Ein H-Signal am Eingang "Clr" (Clear) lädt den Zähler mit Null.

Ist der Zählerinhalt > 0 , wird der Ausgang "Q" = H. Ist der Zählerwert $<$ oder $= 0$, wird der Ausgang "Q" = L.

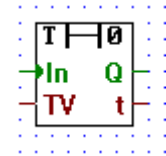
Bei einem Ueberlauf des Zählers, im positiven und im negativen Bereich, wird der Ausgang "Err" = H (Error).

Notizen:

4.4.4 Zeitfunktionen (Timer)

[sfuptime]

4.4.4.1 Anzugverzögerung



[_ondel]

Eingänge: > In: Start Binärformat
 = TV: Zeitwert Ganzzahlformat

Ausgänge: - Q: Ausgang Binärformat
 = t: Timerstand Ganzzahlformat

Anzugverzögerung:

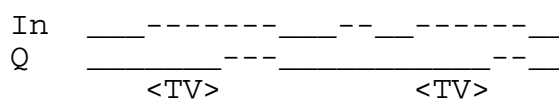
Nach dem H Setzen des Eingangs "In" wird der Ausgang "Q" nach der abgelaufenen Verzögerungszeit, die am Eingang "TV" (Timer Value) als Ganzzahlwert anliegt, = H gesetzt.

Wird der Eingang "In" = L, wird der Ausgang auch = L.

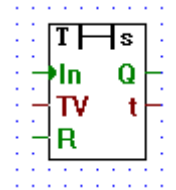
Wird der Eingang "In" für eine kürzere Zeit als "TV" = H, bleibt der Ausgang "Q" unberührt.

Am Ausgang "t" kann der Stand des internen Timers abgelesen werden.

Funktionsdiagramm:



4.4.4.2 Anzugverzögerung mit Selbsthaltung



[_stode!]

Eingänge: > In: Start Binärformat
 = TV: Zeitwert Ganzzahlformat
 – R: Reset Ganzzahlformat

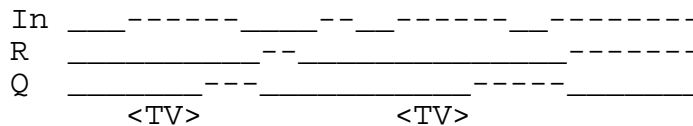
Ausgänge: – Q: Ausgang Binärformat
 = t: Timerstand Ganzzahlformat

Wird der Eingang "In" für eine Zeit die länger als der Wert am Eingang "TV" (Timer Value) ist = H, wird der Ausgang "Q" nach dem Ablauf der Verzögerungszeit "TV" = H und bleibt = H, bis am binären Eingang "R" (Reset) ein H-Signal angelegt wird.

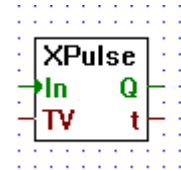
Ist der Eingang "R" = H, hat der Eingang "In" keinen Einfluss mehr.

Am Ausgang "t" kann der Stand des internen Timers abgelesen werden.

Funktionsdiagramm:



4.4.4.3 Anzugwisch-Funktion



[_xpulse]

Eingänge: > In: Start Binärformat
 = TV: Zeitwert Ganzzahlformat

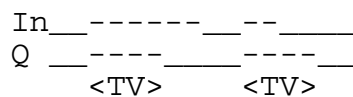
Ausgänge: - Q: Ausgang Binärformat
 = t: Timerstand Ganzzahlformat

Wird am binären Eingang "In" eine ansteigende Flanke erkannt, geht der Ausgang "Q" für die am Eingang "TV" (Timer Value) definierte Zeit in den H-Zustand.

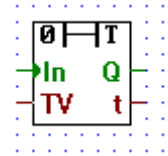
Wird während der laufenden Zeit eine neue Flanke erkannt, beginnt die Ablaufzeit von neuem.

Am Ausgang "t" kann der Stand des internen Timers abgelesen werden.

Funktionsdiagramm:



4.4.4.4 Abfallverzögerung



[_offdel]

Eingänge: > In: Start Binärformat
 = TV: Zeitwert Ganzzahlformat

Ausgänge: - Q: Ausgang Binärformat
 = t: Timerstand Ganzzahlformat

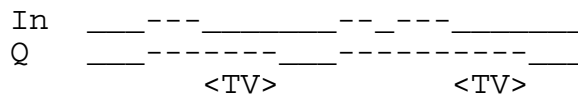
Wird der binäre Eingang "In" = H, wird der binäre Ausgang "Q" auch = H.

Nachdem der Eingang "In" = L wird, erfolgt das Ausschalten des Ausganges "Q" nach der am Eingang "TV" (Timer Value) angelegten Verzögerungszeit.

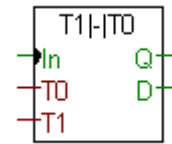
Die Verzögerungszeit "TV" wird in Zeiteinheiten der PCD eingegeben, in 1/10 sec, wenn nicht anders definiert.

Am Ausgang "t" kann der Stand des internen Timers abgelesen werden.

Funktionsdiagramm:



4.4.4.5 Anzug- und Abfallverzögerung



[_onoffd2]

Eingänge: > In: Start Binärformat
 = T0: Zeitwert 0 Ganzzahlformat
 = T1: Zeitwert 1 Ganzzahlformat

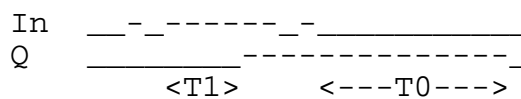
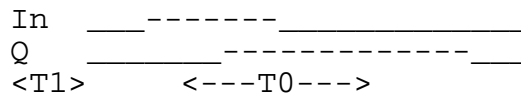
Ausgänge: – Q: Ausgang Binärformat
 – D: Verzögerung aktiv/inaktiv Binärformat

Der Ausgang "Q" wird um "T1" verzögert eingeschaltet, wenn der Eingang "In" = H wird.

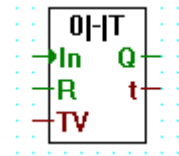
Der Ausgang "Q" wird um "T0" verzögert ausgeschaltet, wenn der Eingang "In" wieder = L wird.

Ausgang "D" = H: Verzögerung aktiv,
 Ausgang "D" = L: Verzögerung inaktiv.

Funktionsdiagramm:



4.4.4.6 Abfallverzögerung mit Rückstellung



[_offdelr]

Eingänge: > In: Start Binärformat
 - R: Reset Binärformat
 = TV: Zeitwert Ganzzahlformat

Ausgänge: - Q: Ausgang Binärformat
 = t: Timerstand Ganzzahlformat

Abfallverzögerung mit Reset.

Wird der Eingang "In" = H, wird der Ausgang "Q" sofort = H.

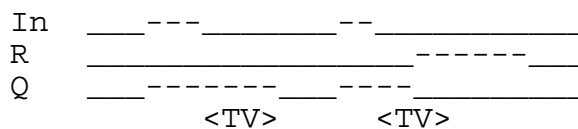
Der Ausgang "Q" wird um den am Eingang "TV" (Time Value) angelegten Wert (Zeiteinheiten) verzögert, nachdem der Eingang "In" = L wurde.

Wird der Eingang "R" (Reset) = H, wird das Zeitglied zurückgestellt und der Ausgang "Q" = L.

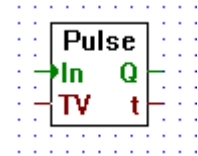
Geht der Eingang "In" für eine kürzere Dauer als an "TV" angelegt "H" - "L" - "H" wird der Ausgang "Q" nicht beeinflusst.

Am Ausgang "t" kann der aktuelle Stand des Zeitgliedes gelesen werden.

z.B.:



4.4.4.7 Impulsfunktion



[_pulse]

Eingänge: > In: Start Binärformat
 = TV: Zeitwert Ganzzahlformat

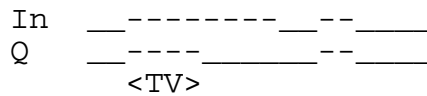
Ausgänge: - Q: Ausgang Binärformat
 = t: Timerstand Ganzzahlformat

Wird der Eingang "In" = H, wird der Ausgang "Q" sofort für die am Eingang "TV" (Timer Value) angelegte Zeit = H.

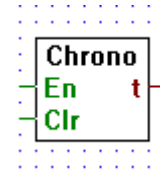
Ist die Zeit "TV" noch nicht erreicht wenn am Eingang "In" das H-Signal wieder weggenommen wird, fällt der Ausgang "Q" sofort zurück.

Am Ausgang "t" kann der Stand des internen Timers abgelesen werden.

Funktionsdiagramm:



4.4.4.8 Stoppuhr (Chrono)



[_chrono2]

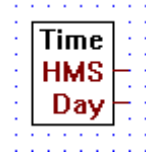
Eingänge: – En: Enable Binärformat
 – Clr: Clear Binärformat

Ausgang: = t: Timerstand Ganzzahlformat

Die Funktion dient zur Messung von Zeitintervallen. Ein H-Signal am binären Eingang "Clr" (Clear) stellt die Stoppuhr zurück. Wird an den binären Eingang "En" (Enable) ein H-Signal angelegt, läuft die Uhr, wird ein L-Signal angelegt stoppt die Uhr. Am Ausgang "t" kann der Stand der Stoppuhr gelesen werden.

Die Inkrementierung erfolgt jede Sekunde.

4.4.4.9 Hardwareuhr der PCD



[_time]

Ausgänge: = HMS: Hours, Minutes, Seconds Ganzz.form.
 = Day: Day and Date Ganzz.form.

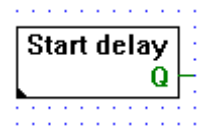
An 2 Ausgängen liegen die Informationen der PCD-internen Hardwareuhr im Ganzzahlformat vor.

Die Bedeutungen sind die folgenden:

| HMS | | Day | |
|---------------|------------------|---------------|------------------|
| <u>Inhalt</u> | <u>Digit Nr.</u> | <u>Inhalt</u> | <u>Digit Nr.</u> |
| 0 | 9-6 | 0 | 9 |
| Stunden | 5-4 | Woche | 8-7 |
| Minuten | 3-2 | Wochentag | 6 |
| Sekunden | 1-0 | Jahr | 5-4 |
| | | Monat | 3-2 |
| | | Tag | 1-0 |

Siehe auch die Beschreibung des Befehls RTIME im PCD-Befehlssatz.

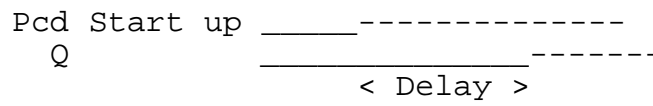
4.4.4.10 Startverzögerung



[_stdeI]

Ausgang: – Q: Binärformat

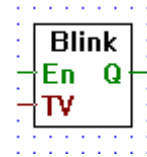
Beim Einschalten der PCD vergeht die im Einstellfenster definierte Zeit, bis der Ausgang "Q" = H wird.



4.4.5 Blinker

[sfupblin]

4.4.5.1 Blinker symmetrisch



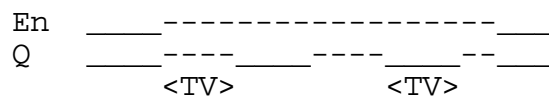
[_blink1]

Eingänge: – En: Enable Binärformat
 = TV Time Value (Zeit Wert) Ganzzahlformat

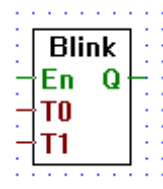
Ausgang: – Blinker Ausgang Binärformat

Ausgang "Q" blinkt, wenn der Eingang "En" (Enable) = H ist. Die Impuls- bzw. die Pausenzeit des Blinkers erfolgt durch Anlegen eines ganzzahligen Wertes an den Eingang "TV" (Time Value).

Ist der Eingang "En" = L, wird der Ausgang "Q" sofort = L.



4.4.5.2 Blinker asymmetrisch



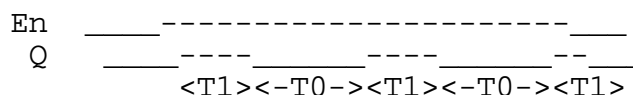
[_blink2]

Eingänge: – En: Enable Binärformat
 = T0 Time Value 0 (Zeit Wert 1) Ganzzahlformat
 = T1 Time Value 1 (Zeit Wert 2) Ganzzahlformat

Ausgang: – Blinker Ausgang Binärformat

Ausgang "Q" blinkt, wenn der Eingang "En" (Enable) = H ist. Ein ganzzahliger Wert am Eingang "T1" ergibt die Impulszeit. Ein ganzzahliger Wert am Eingang "T0" ergibt die Pausenzeit.

Ist der Eingang "En" = L, wird der Ausgang "Q" sofort = L.



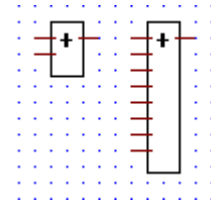
4.4.6 Arithmetik im Ganzzahlformat

[sfupinte]

4.4.6.1 Addition



[_iadd]

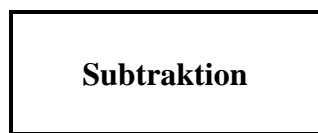


Alle Ein- und Ausgänge: – → Ganzzahlformat

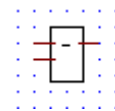
Addiert die Werte an den Eingängen und leitet diese an den Ausgang.

Ausziehbar von 2 auf 8 Eingänge.

4.4.6.2 Subtraktion



[_isub]



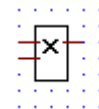
Alle Ein- und Ausgänge: – → Ganzzahlformat

Es wird der Wert am unteren Eingang vom Wert am oberen Eingang subtrahiert und am Ausgang ausgegeben.

4.4.6.3 Multiplikation



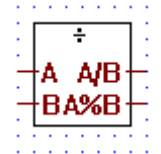
[_imul]



Alle Ein- und Ausgänge: – → Ganzzahlformat

Am Ausgang erscheint das Produkt der beiden Eingänge.

4.4.6.4 Division



[_idiv]

Alle Ein- und Ausgänge: – → Ganzzahlformat

Das Resultat der Division des Wertes am Eingang "A" dividiert durch den Wert am Eingang "B" erscheint am Ausgang "A/B". Der Rest wird am Ausgang "A%B" ausgegeben.

4.4.6.5 Quadratwurzel

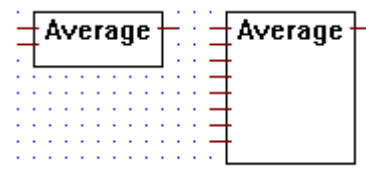


[_isqr]

Alle Ein- und Ausgänge: – → Ganzzahlformat

Die Quadratwurzel des Eingangswertes wird am Ausgang ausgegeben. Das Resultat wird abgerundet. Es entsteht kein Rest.

4.4.6.6 Durchschnitt



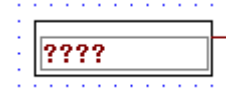
[_iaverage]

Alle Ein- und Ausgänge: – → Ganzzahlformat

Es wird der Durchschnitt der Werte an den Eingängen am Ausgang ausgegeben

Ausziehbar von 2 auf 8 Eingänge.

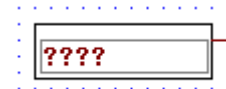
4.4.6.7

Konstante

[_iconst]

Die im Eintragsfeld stehende Ganzzahl-Konstante wird an den Ausgang ausgegeben.

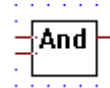
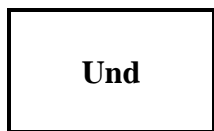
4.4.6.8

Absolutwert

[_iabs]

Am Ausgang erscheint der Absolutwert des Wertes am Eingang. Liegt am Eingang z.B. -12345, erscheint am Ausgang 12345.

4.4.6.9

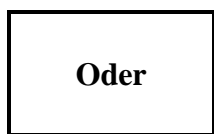
Und-Verknüpfung von Registern

[_iand]

Alle Ein- und Ausgänge: – → Ganzzahlformat

Am Ausgang erscheint die logische UND-Verknüpfung der 32-Bit-Information welche an den beiden Eingängen anliegt.

4.4.6.10

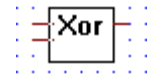
Oder-Verknüpfung von Registern

[_ior]

Alle Ein- und Ausgänge: – → Ganzzahlformat

Am Ausgang erscheint die logische ODER-Verknüpfung der 32-Bit-Information welche an den beiden Eingängen anliegt.

4.4.6.11 Exklusiv-Oder-Verknüpfung von Registern

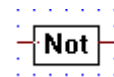


[_ixor]

Alle Ein- und Ausgänge: – → Ganzzahlformat

Am Ausgang erscheint die logische XOR-Verknüpfung der 32-Bit-Information welche an den beiden Eingängen anliegt.

4.4.6.12 Komplementieren eines Registers

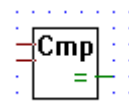


[_inot]

Alle Ein- und Ausgänge: – → Ganzzahlformat

Gibt am Ausgang den komplementierten 32-Bit Wert des Ganzzahlwertes, der am Eingang anliegt, aus. (1er-Komplement)

4.4.6.13 Vergleich auf 2 gleiche Werte

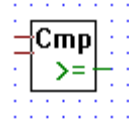


[_icmpeq]

Eingang: = Ganzzahlformat

Ausgang: – Binärformat

Haben beide Eingänge den gleicher Wert, wird der binäre Ausgang = H.

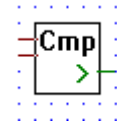
4.4.6.14 Vergleich auf grösseren oder gleichen Wert

[_icmpge]

Eingang: = Ganzzahlformat

Ausgang: – Binärformat

Ist der obere Eingangswert gleich oder grösser als der untere, wird der binäre Ausgang = H.

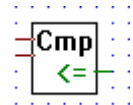
4.4.6.15 Vergleich auf grösseren Wert

[_icmpgt]

Eingang: = Ganzzahlformat

Ausgang: – Binärformat

Ist der obere Eingangswert grösser als der untere, wird der binäre Ausgang = H.

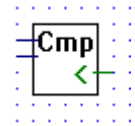
4.4.6.16 Vergleich auf kleineren oder gleichen Wert

[_icmpse]

Eingang: = Ganzzahlformat

Ausgang: – Binärformat

Ist der obere Eingangswert gleich oder kleiner als der untere Eingangswert, wird der binäre Ausgang = H.

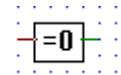
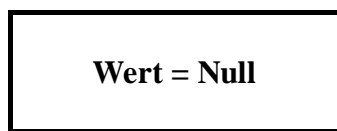
4.4.6.17 Vergleich auf kleineren Wert

[_icmpst]

Eingang: = Ganzzahlformat

Ausgang: – Binärformat

Ist der obere Eingangswert kleiner als der untere, wird der binäre Ausgang = H.

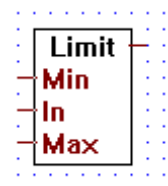
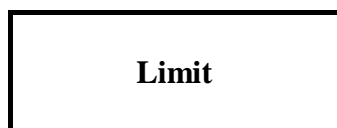
4.4.6.18 Wert = Null

[_iiszero]

Eingang: = Ganzzahlformat

Ausgang: – Binärformat

Der Ausgang wird = H, wenn der Ganzzahl-Eingangswert = 0 ist.

4.4.6.19 Wert zwischen Grenzen

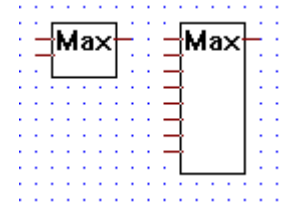
[_ilimit]

Alle Ein- und Ausgänge: – → Ganzzahlformat

Am Ausgang erscheint der Wert des Eingangs "In", wobei der Ausgangswert durch die Werte an den Eingängen "Min" und "Max" begrenzt sind.

4.4.6.20 Grösster Wert wird ausgegeben

[_imax]

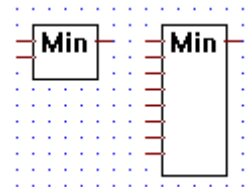


Alle Ein- und Ausgänge: – → Ganzzahlformat

Der grösste Wert der an den Eingängen liegt, wird am Ausgang ausgegeben.

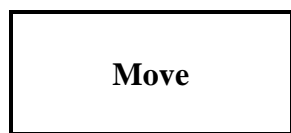
Ausziehbar von 2 auf 8 Eingänge.**4.4.6.21 Kleinster Wert wird ausgegeben**

[_imin]



Alle Ein- und Ausgänge: – → Ganzzahlformat

Der kleinste Wert der an den Eingängen liegt, wird am Ausgang ausgegeben.

Ausziehbar von 2 auf 8 Eingänge.**4.4.6.22 Bringt den Eingangswert zum Ausgang**

[_imove]

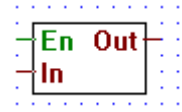


Alle Ein- und Ausgänge: – → Ganzzahlformat

Am Ausgang erscheint der Ganzzahlwert der am Eingang angelegt ist.

Dient zum direkten Durchschalten eines Eingangswertes zu einem Ausgangswert.

4.4.6.23 Move mit Enable



[_imovee]

Eingang: – En: Binärformat
 – In: Ganzzahlformat

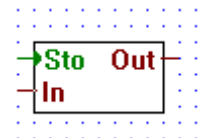
Ausgang: – Out: Fließpunktformat

Dient zum direkten Durchschalten eines Eingangswertes zu einem Ausgangswert. Die Durchschaltung erfolgt aber nur, wenn am Eingang "En"

(Enable) ein H-Signal anliegt.

Ist "En" = L, erscheint am Ausgang der Wert Null.

4.4.6.24 Move mit Speicher



[_imoves]

Eingang: > Sto: Binärformat
 – In: Ganzzahlformat

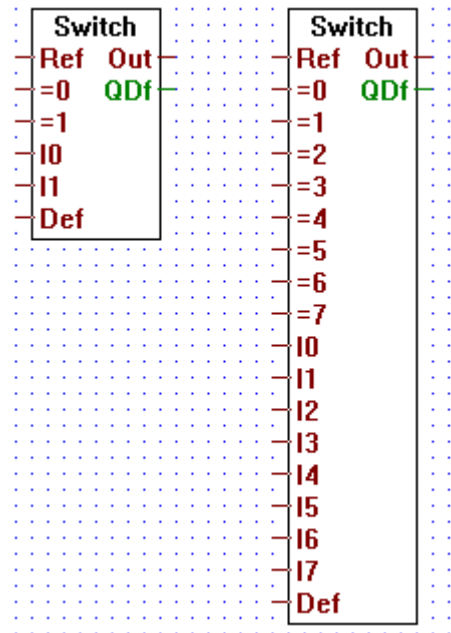
Ausgang: – Out: Ganzzahlformat

Der Wert am Eingang "In" wird in ein internes Register abgelegt, wenn der Eingang "Sto" (Store) von L nach H geht (ansteigende Flanke). Am Ausgang erscheint immer der Wert des internen Registers.

4.4.6.25 Durchschaltung vordefinierter Werte



[_iswitch]



Eingänge: – alle → Ganzzahlformat

Ausgänge: – Out → Ganzzahlformat
 – QDf → Binärformat

Stimmt der Wert des Eingang "Ref" mit einem Wert der Eingänge "=x" überein, so wird der Wert des korrespondierenden Eingangs "Ix" an den Ausgang "Out" ausgegeben. Stimmt keiner der Werte mit "Ref" überein, wird der Wert am Eingang "Def" an den Ausgang "Out" übertragen und der binäre Ausgang "QDf" wird = H.

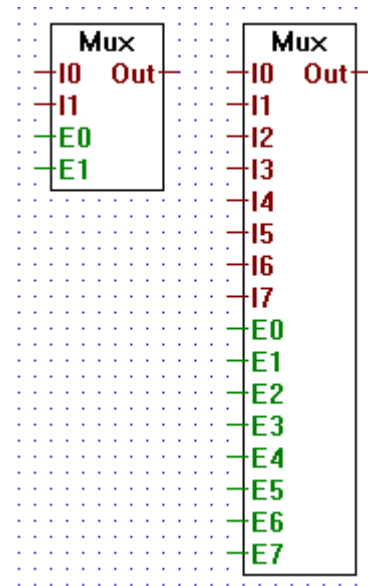
Ausziehbar von 2 auf 8.

Der Eingang "I0" hat die höchste, "I7" die niedrigste Priorität. Stimmen also 2 oder mehr "=x"-Eingänge mit dem Wert von "Ref" überein, so wird nur der Wert mit der höchsten Priorität zum Ausgang übertragen.

4.4.6.26 Multiplexer mit binärer Vorwahl



[_imux]



Eingänge:

- I0:
- | Ganzzahlformat
- I7:
- E0:
- | Binärformat
- E7:

Ausgang: – Out: Ganzzahlformat

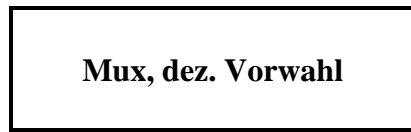
Leitet den Ganzzahlwert am Eingang "Ix" zum Ausgang "OUT" wenn der entsprechende binäre Enable-Eingang "Ex" = H ist. (x = 0 - 7)

Ausziehbar von 2 auf 8 Eingänge.

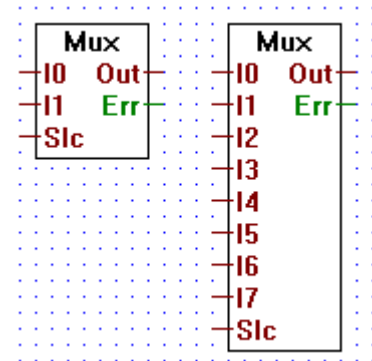
Sind alle Enable-Eingänge "Ex" = L, wird der Ausgang "OUT" = 0.

Der Enable-Eingang "E0" hat die höchste, "E7" die niedrigste Priorität. Sind also 2 oder mehr Enable-Eingänge gleichzeitig = H, ist der Eingang mit der höchsten Priorität bestimmend.

4.4.6.27 Multiplexer mit dezimaler Vorwahl



[_Imux2]



Eingänge:

- I0: |
- | Ganzzahlformat
- I7: |
- Slc Ganzzahlformat (Slc = Select)

Ausgänge:

- Out Ganzzahlformat
- Err Binärformat

Leitet den Ganzzahlwert am Eingang "Ix" zum Ausgang "Out". Die Wahl des Einganges erfolgt durch Anlegen eines ganzzahligen Wertes an den Eingang "Slc". Dieser Wert entspricht der Nummer des zu wählenden Einganges. Wird z.B. 2 an den Eingang "Slc" angelegt, wird der binäre Eingang "I2" zum Ausgang "Out" geleitet.

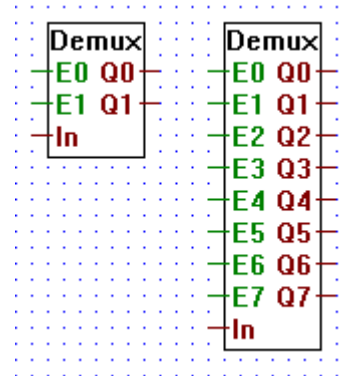
Ausziehbar von 2 auf 8 Eingänge.

Der Ausgang "Err" (Error) wird = H, wenn ein unmöglicher Wert am Eingang "Slc" angelegt wird.

4.4.6.28 Demultiplexer mit binärer Vorwahl



[_idemux]



Eingänge: – E0:
 | Binärformat
 – E7:
 = In Ganzzahlformat

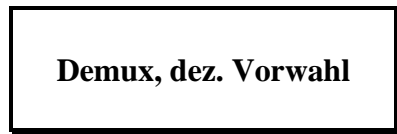
Ausgänge: – Q0
 | Ganzzahlformat
 – Q7

Leitet den Ganzzahlwert am Eingang "In" zum Ausgang "Qx", wenn der entsprechende binäre Enable-Eingang "Ex" = H ist. (x = 0 - 7).

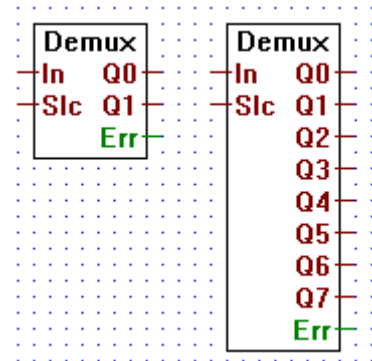
Ausziehbar von 2 auf 8 Ausgänge.

Ist ein Eingangskanal "Ex" = L, wird der entsprechende Ausgang "Qx" = 0.

4.4.6.29 Demultiplexer mit dezimaler Vorwahl



[_idemux2]



Eingänge: – In: Ganzzahlformat
 = Slc: Ganzzahlformat (Slc = Select)

Ausgänge: – Q0
 | Ganzzahlformat
 – Q7
 – Err: Binärformat

Überträgt den Ganzzahlwert am Eingang "In" zu einem Ausgang "Qx". Die Ausgänge werden mit dem Anlegen eines ganzzahligen Wertes an den Eingang "Slc" gewählt. Dieser Wert entspricht der Nummer des zu wählenden Ausganges. Wird z.B. 2 an den Eingang "Slc" angelegt, wird der Wert des Eingang "In" an den Ausgang "Q2" geleitet.

Alle nicht gewählten Ausgänge sind = 0.

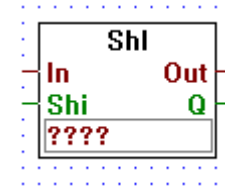
Ausziehbar von 2 auf 8 Ausgänge.

Der Ausgang "Err" (Error) wird = H, wenn ein unmöglicher Wert am Eingang "Slc" angelegt wird.

4.4.6.30 Schiebe Registerinhalt nach links



[_ishftl2]



Eingänge: – In: (Ganzzahlformat)
– Shi: Shift (Binärformat)

Ausgänge: – Out (Ganzzahlformat)
– Q: – Out (Binärformat)

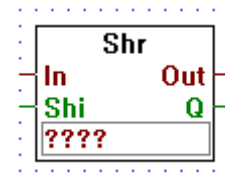
Am Ausgang "Out" erscheint der Ganzzahlwert der am Eingang "In" anliegt und zwar um so viele Bit nach links geschoben wie im Eintragungsfeld angegeben sind. Der Ausgang "Q" zeigt den Status des zuletzt geschobenen Bits. Am Eingang "Shi" wird der Wert des Bits, das ins Bit 0 geschoben wird, angegeben (0/1).

(Dem Befehl liegt das interne 32-Bit Format zugrunde).

4.4.6.31 Schiebe Registerinhalt nach rechts



[_ishfr2]



Eingänge: – In: (Ganzzahlformat)
– Shi: Shift (Binärformat)

Ausgänge: – Out (Ganzzahlformat)
– Q: – Out (Binärformat)

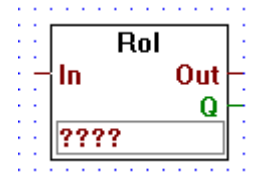
Am Ausgang "Out" erscheint der Ganzzahlwert der am Eingang "In" anliegt und zwar um so viele Bit nach rechts geschoben wie im Eintragungsfeld angegeben sind. Der Ausgang "Q" zeigt den Status des zuletzt geschobenen Bits. Am Eingang "Shi" wird der Wert des Bits, das ins Bit 0 geschoben wird, angegeben (0/1)

(Dem Befehl liegt das interne 32-Bit Format zugrunde).

4.4.6.32 Rotiere Registerinhalt nach links



[_irotlf2]



Eingänge: – In: (Ganzzahlformat)

Ausgänge: – Out (Ganzzahlformat)

– Q: – Out (Binärformat)

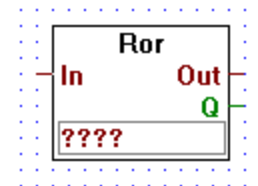
Am Ausgang "Out" erscheint der Ganzzahlwert der am Eingang "In" anliegt und zwar um so viele Bit nach links rotiert wie im Eintragsfeld angegeben sind. Der Ausgang "Q" zeigt den Status des zuletzt rotierten Bits.

(Dem Befehl liegt das interne 32-Bit Format zugrunde).

4.4.6.33 Rotiere Registerinhalt nach rechts



[_irotri2]



Eingänge: – In: (Ganzzahlformat)

Ausgänge: – Out (Ganzzahlformat)

– Q: – Out (Binärformat)

Am Ausgang "Out" erscheint der Ganzzahlwert der am Eingang "In" anliegt und zwar um so viele Bit nach rechts rotiert wie im Eintragsfeld angegeben sind. Der Ausgang "Q" zeigt den Status des zuletzt rotierten Bits.

(Dem Befehl liegt das interne 32-Bit Format zugrunde).

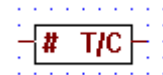
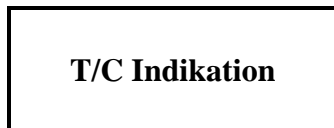
4.4.6.34 Register Indikation

[_iregin]

Alle Ein- und Ausgänge: – → Ganzzahlformat

Am Ausgang erscheint der Ganzzahlwert des am Eingang bezeichneten Registers "#"

Ist "#" z.B. 5, wird der Inhalt des Registers 5 ausgegeben.

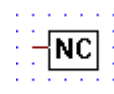
4.4.6.35 Timer/Counter Indikation

[_itcind]

Alle Ein- und Ausgänge: – → Ganzzahlformat

Am Ausgang erscheint der Ganzzahlwert des am Eingang bezeichneten Timers oder Counters "#".

Ist "#" z.B. 5, wird der Inhalt des T/C 5 ausgegeben.

4.4.6.36 Nicht angeschlossen

[_inotcn]

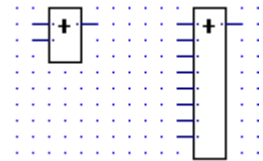
Die "Nicht angeschlossen"-Funktion dient zum Abschluss nicht verwendeter Ausgänge im Ganzzahlformat.

4.4.7 Arithmetik im Fließpunktformat

[sfupfloa]

4.4.7.1

Addition



[_fadd2]

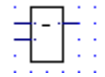
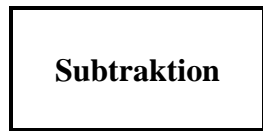
Alle Ein- und Ausgänge: – → Fließpunktformat

Addiert die Werte an den Eingängen und leitet diese an den Ausgang.

Ausziehbar von 2 auf 8 Eingänge.

4.4.7.2

Subtraktion



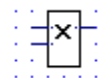
[_fsub]

Alle Ein- und Ausgänge: – → Fließpunktformat

Es wird der Wert am unteren Eingang vom Wert am oberen Eingang subtrahiert und am Ausgang ausgegeben.

4.4.7.3

Multiplikation



[_fmul]

Alle Ein- und Ausgänge: – → Fließpunktformat

Am Ausgang erscheint das Produkt der beiden Eingänge.

4.4.7.4

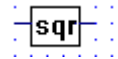
Division

[_fdiv]

Alle Ein- und Ausgänge: – → Fließpunktformat

Das Resultat der Division des Wertes des oberen durch den Wert des unteren Einganges erscheint am Ausgang.

4.4.7.5

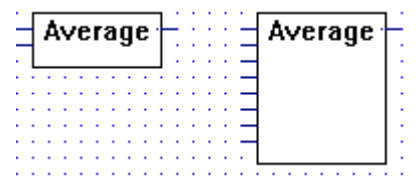
Quadratwurzel

[_fsqr]

Alle Ein- und Ausgänge: – → Fließpunktformat

Die Quadratwurzel des Eingangswertes wird am Ausgang ausgegeben

4.4.7.6

Durchschnitt

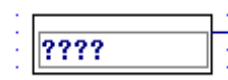
[_faverage]

Alle Ein- und Ausgänge: – → Fließpunktformat

Es wird der Durchschnitt der Werte an den Eingängen am Ausgang ausgegeben

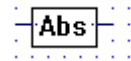
Ausziehbar von 2 auf 8 Eingänge.

4.4.7.7

Konstante

[_fconst]

Die im Eintragungsfeld stehende Fließpunkt-Konstante wird an den Ausgang ausgegeben.

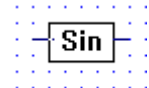
4.4.7.8 Absolutwert

[_fabs]

Alle Ein- und Ausgänge: – → Fließpunktformat

Am Ausgang erscheint der Absolutwert des Wertes am Eingang.

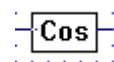
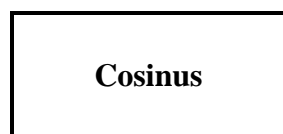
Liegt am Eingang z.B. -1.2345, erscheint am Ausgang 1.2345.

4.4.7.9 Sinus

[_fsin]

Alle Ein- und Ausgänge: – → Fließpunktformat

Der Sinus des Eingangswertes wird im Bogenmass am Ausgang ausgegeben.

4.4.7.10 Cosinus

[_fcos]

Alle Ein- und Ausgänge: – → Fließpunktformat

Der Cosinus des Eingangswertes wird im Bogenmass am Ausgang ausgegeben.

4.4.7.11 Arcus Tangens**Arcus Tangens****Atan**

[_fatan]

Alle Ein- und Ausgänge: – → Fließpunktformat

Am Ausgang erscheint der Arcus Tangens des Wertes am Eingang.

Der Wert am Ausgang ist im Bogenmass.**4.4.7.12 Natürlicher Exponent - Exponentialfunktion****Nat. Exponent****Exp**

[_fexp]

Alle Ein- und Ausgänge: – → Fließpunktformat

Am Ausgang erscheint e^{\wedge} Eingangswert**4.4.7.13 Natürlicher Logarithmus****Nat. Logarithmus****Ln**

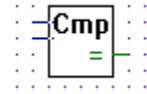
[_fln]

Alle Ein- und Ausgänge: – → Fließpunktformat

Am Ausgang erscheint der natürliche Logarithmus des Eingangswertes.

4.4.7.14 Vergleich auf 2 gleiche Werte

Werte sind gleich



[_fcmpeq]

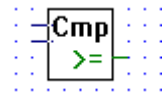
Alle Eingänge: – Fließpunktformat

Ausgang: – Binärformat

Haben beide Eingänge den gleichen Wert, wird der binäre Ausgang = H.

4.4.7.15 Vergleich auf grösseren oder gleichen Wert

Wert grösser/ gleich



[_fcmpge]

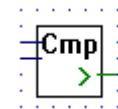
Alle Eingänge: – Fließpunktformat

Ausgang: – Binärformat

Ist der obere Eingangswert gleich oder grösser als der untere, wird der binäre Ausgang = H.

4.4.7.16 Vergleich auf grösseren Wert

Wert grösser

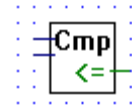


[_fcmpgt]

Alle Eingänge: – Fließpunktformat

Ausgang: – Binärformat

Ist der obere Eingangswert grösser als der untere, wird der binäre Ausgang = H.

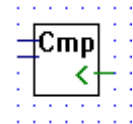
4.4.7.17 Vergleich auf kleineren oder gleichen Wert

[_fcmpse]

Alle Eingänge: – Fließpunktformat

Ausgang: – Binärformat

Ist der obere Eingangswert gleich oder kleiner als der untere Eingangswert, wird der binäre Ausgang = H

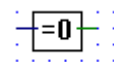
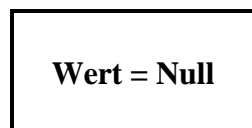
4.4.7.18 Vergleich auf kleineren Wert

[_fcmpst]

Alle Eingänge: – Fließpunktformat

Ausgang: – Binärformat

Ist der obere Eingangswert kleiner als der untere, wird der binäre Ausgang = H.

4.4.7.19 Wert gleich Null

[_fiszero]

Eingang: – Fließpunktformat

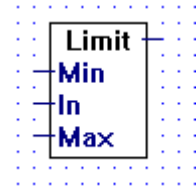
Ausgang: – Binärformat

Der Ausgang wird = H, wenn der Fließpunkt-Eingangswert = 0 ist.

4.4.7.20 Limit - Wert zwischen Grenzen



[_flimit]



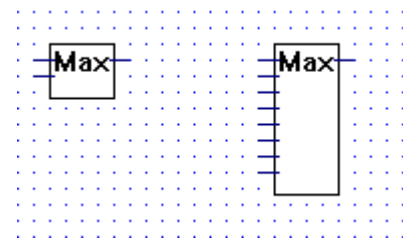
Alle Ein- und Ausgänge: – → Fließpunktformat

Am Ausgang erscheint der Wert des Eingangs "In", wobei der Ausgangswert durch die Werte an den Eingängen "Min" und "Max" begrenzt sind.

4.4.7.21 Maximum - Grösster Wert wird ausgegeben



[_fmax2]



Alle Ein- und Ausgänge: – → Fließpunktformat

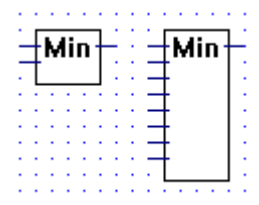
Der grösste Wert der an den Eingängen liegt, wird am Ausgang ausgegeben.

Ausziehbar von 2 auf 8 Eingänge.

4.4.7.22 Minimum - Kleinster Wert wird ausgegeben



[_fmin2]

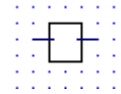


Alle Ein- und Ausgänge: – → Fließpunktformat

Der kleinste Wert der an den Eingängen liegt, wird am Ausgang ausgegeben.

Ausziehbar von 2 auf 8 Eingänge.

4.4.7.23 Move - Bringt den Eingangswert zum Ausgang



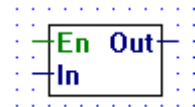
[_fmove]

Alle Ein- und Ausgänge: – → Fliesspunktformat

Am Ausgang erscheint der Fliesspunktwert der am Eingang angelegt ist.

Dient zum direkten Durchschalten eines Eingangswertes zu einem Ausgangswert.

4.4.7.24 Move mit Enable



[_fmovee]

Eingang: – En: Binärformat
– In: Fliesspunktformat

Ausgang: – Out: Fliesspunktformat

Dient zum direkten Durchschalten eines Eingangswertes zu einem Ausgangswert. Die Durchschaltung erfolgt aber nur, wenn am Eingang "En" (Enable) ein H-Signal anliegt. Ist der Eingang "En" = L, wird der Ausgang "Out" = 0.

4.4.7.25

Move mit Speicher

[_fmoves]

Eingang: > Sto: Binärformat
 – In: Fliesspunktformat

Ausgang: – Out: Fliesspunktformat

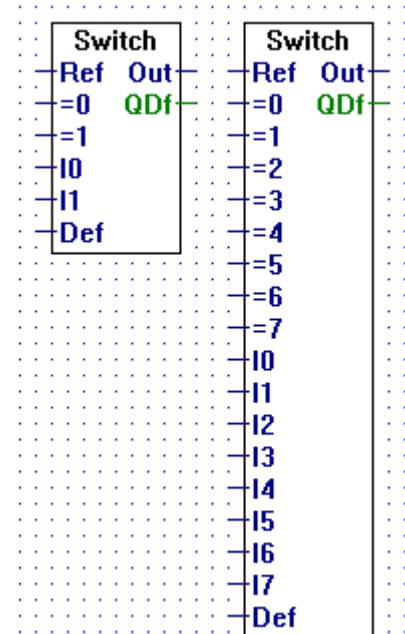
Der Wert am Eingang "In" wird in ein internes Register abgelegt, wenn der Eingang "Sto" (Store) von L nach H geht (ansteigende Flanke). Am Ausgang erscheint immer der Wert des internen Registers.

4.4.7.26

Schalter - Durchschaltung vordefinierter Werte



[_fswitch]



Eingänge: – alle → Fließpunktformat

Ausgänge: – Out → Fließpunktformat
– QDf → Binärformat

Stimmt der Wert des Eingangs "Ref" mit einem Wert der Eingänge "=x" überein, so wird der Wert des korrespondierenden Eingangs "Ix" an den Ausgang "Out" ausgegeben. Stimmt keiner der Werte mit "Ref" überein, wird der Wert am Eingang "Def" an den Ausgang "Out" übertragen und der binäre Ausgang "QDf" wird = H.

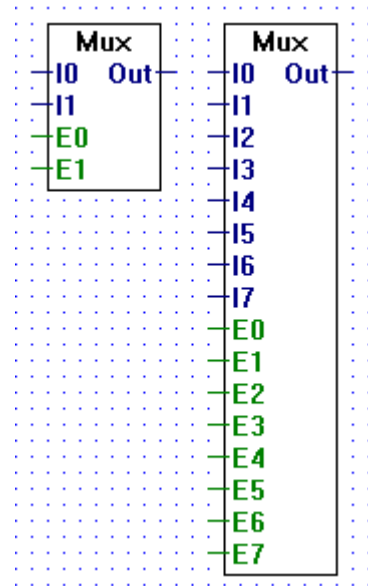
Ausziehbar von 2 auf 8.

Der Eingang "I0" hat die höchste, "I7" die niedrigste Priorität. Stimmen also 2 oder mehr "=x"-Eingänge mit dem Wert von "Ref" überein, so wird nur der Wert mit der höchsten Priorität zum Ausgang übertragen.

4.4.7.27 Multiplexer mit binärer Vorwahl



[_fmux]



Eingänge:

- I0:
- | Fließpunktformat
- I7:
- E0:
- | Binärformat
- E7:

Ausgang: – Out: Fließpunktformat

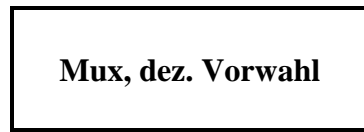
Leitet den Fließpunktwert am Eingang "Ix" zum Ausgang "OUT" wenn der entsprechende binäre Enable-Eingang "Ex" = H ist. (x = 0 - 7)

Ausziehbar von 2 auf 8 Eingänge.

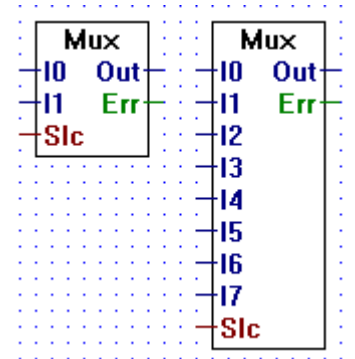
Sind alle Enable-Eingänge "Ex" = L, wird der Ausgang "OUT" = 0.0.

Der Enable-Eingang "E0" hat die höchste, "E7" die niedrigste Priorität. Sind also 2 oder mehr Enable-Eingänge gleichzeitig = H, ist der Eingang mit der höchsten Priorität bestimmend.

4.4.7.28 Multiplexer mit dezimaler Vorwahl



[_fmux2]



Eingänge:

- I0: | Fließpunktformat
- I1: | Fließpunktformat
- I7: | Fließpunktformat
- = Slc Ganzzahlformat

Ausgänge:

- Out Fließpunktformat
- Err Binärformat

Leitet den Fließpunktwert des Eingangs "Ix" zum Ausgang "Out". Die Wahl des Eingangs erfolgt durch Anlegen eines ganzzahligen Wertes an den Eingang "Slc". Dieser Wert entspricht der Nummer des zu wählenden Eingangs. Wird z.B. 2 an den Eingang "Slc" angelegt, wird der binäre Eingang "I2" zum Ausgang "Out" geleitet.

Ausziehbar von 2 auf 8 Eingänge.

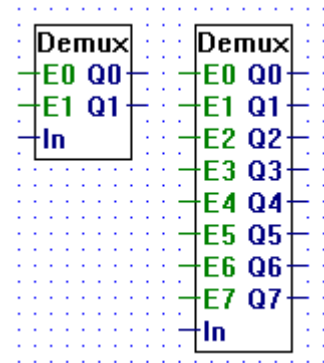
Der Ausgang "Err" (Error) wird = H, wenn ein unmöglicher Wert am Eingang "Slc" angelegt wird.

4.4.7.29

Demultiplexer mit binärer Vorwahl



[_fdemux]



Eingänge: – E0:
 | Binärformat
 – E7:
 = In Fließpunktformat

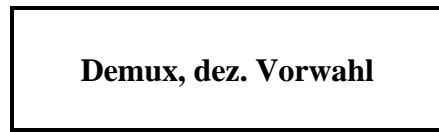
Ausgänge: – Q0
 | Fließpunktformat
 – Q7

Leitet den Fließpunktwert am Eingang "In" zum Ausgang "Qx", wenn der entsprechende binäre Enable-Eingang "Ex" = H ist. (x = 0 - 7)

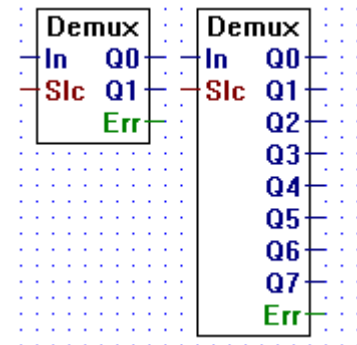
Ausziehbar von 2 auf 8 Eingänge.

Ist ein Eingangskanal "Ex" = L, wird der entsprechende Ausgang "Qx" = 0.0.

4.4.7.30 Demultiplexer mit dezimaler Vorwahl



[_fdemux2]



Eingänge: – In: Fließpunktformat
= Slc: Ganzzahlformat (Slc = Select)

Ausgänge: – Q0
| Fließpunktformat
– Q7
– Err: Binärformat

Überträgt den Fließpunktwert am Eingang "In" zu einem Ausgang "Qx". Die Ausgänge werden mit dem Anlegen eines ganzzahligen Wertes an den Eingang "Slc" gewählt. Dieser Wert entspricht der Nummer des zu wählenden Ausganges. Wird z.B. 2 an den Eingang "Slc" angelegt, wird der Wert des Eingang "In" an den Ausgang "Q2" geleitet.

Alle nicht gewählten Ausgänge sind = 0.0.

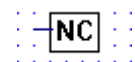
Ausziehbar von 2 auf 8 Ausgänge.

Der Ausgang "Err" (Error) wird = H, wenn ein unmöglicher Wert am Eingang "Slc" angelegt wird.

4.4.7.31 Nicht angeschlossen - Abschluss offener Ausgänge



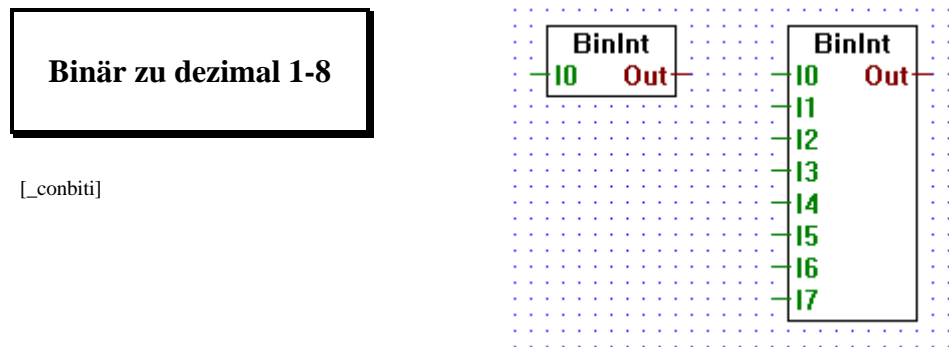
[_Fnotcn]



Die "Nicht angeschlossen" Funktion dient zum Abschluss nicht verwendeter Ausgänge im Fließpunktformat.

4.4.8 Wandler (Binär - Dezimal - Fließpunkt) [sfupconv]

4.4.8.1 Binär zu dezimal allgemein, ab 1-8 Einzelementen



Eingänge: – I0
 | Binärformat
 – I7

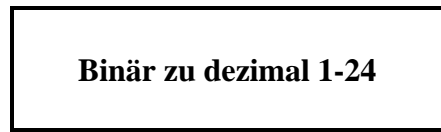
Ausgang: = Out Ganzzahlformat

Überträgt die Binärinformation an den Eingängen "I0" - "I7" als Ganzzahlwert an den Ausgang "Out". "I0" hat die kleinste Wertigkeit.

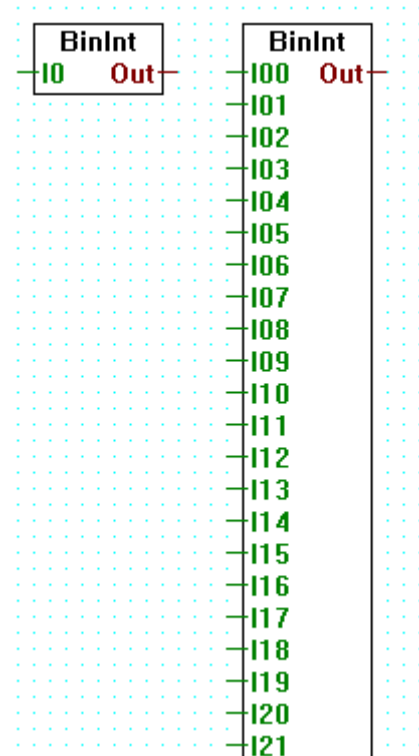
Alle am Ausgang nicht verwendeten Bit werden auf 0 gesetzt.

Ausziehbar von 1 auf 8 Eingänge.

4.4.8.2 Binär zu dezimal allgemein, ab 1-24 Einzelementen



[_conbitim]



Eingänge: – I0:
 | Binärformat
 – I23

Ausgang: = Out Ganzzahlformat

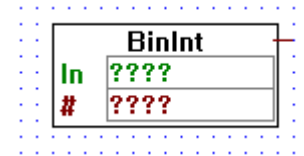
Überträgt die Binärinformation an den Eingängen "I0" - "I23" als Ganzzahlwert an den Ausgang "Out". "I0" hat die kleinste Wertigkeit.

Alle am Ausgang nicht verwendeten Bit werden auf 0 gesetzt.

Ausziehbar von 1 auf 24 Eingänge.

Für mehr als 24 Bit verwenden Sie bitte die FBox "Binär zu dezimal"

4.4.8.3 Binär zu dezimal, ab Elementreihe (PCD-Format)



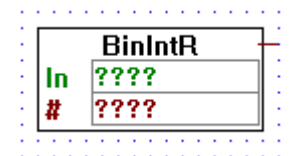
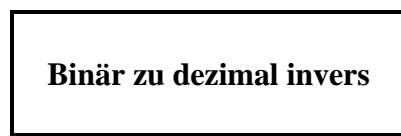
[_conbitiq]

Ausgang: = Ganzzahlformat

Ueberträgt eine Binärinformation ab dem Eingang "In" als Ganzzahlwert an den Ausgang. Im Eintragsfeld "#" wird die Anzahl Bit (I, O ,F) und im Eintragsfeld "In" wird die tiefste Adresse der zusammenhängenden Elementreihe angegeben.

Die tiefste Elementadresse hat die kleinste Wertigkeit (PCD-Format).

4.4.8.4 Binär zu dezimal invers, ab Elementreihe (PCA-Format)



[_conbitiqr]

Ausgang: = Ganzzahlformat

Ueberträgt eine Binärinformation ab dem Eingang "In" als Ganzzahlwert an den Ausgang. Im Eintragsfeld "#" wird die Anzahl Bit (I, O ,F) und im Eintragsfeld "In" wird die tiefste Adresse der zusammenhängenden Elementreihe angegeben.

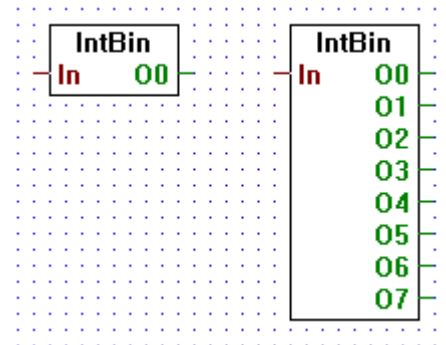
Die tiefste Elementadresse hat die grösste Wertigkeit (PCA-Format).

4.4.8.5

Dezimal zu binär allgemein, auf 1-8 Einzelelemente



[_conbito]



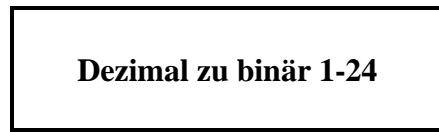
Eingang: = In: Ganzzahlformat

Ausgänge: – O0:
 | Binärformat
 – O7

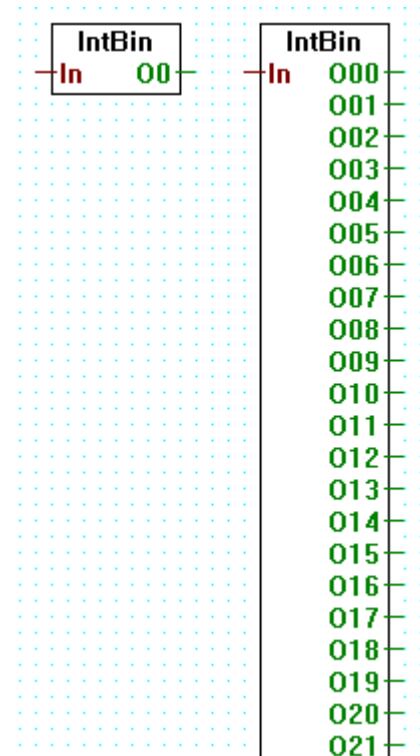
Überträgt den ganzzahligen Wert der an "In" anliegt auf die binären Ausgänge "O0" - "O7". "O0" ist das Bit mit der niedrigsten Wertigkeit.

Ausziehbar von 1 auf 8 Ausgänge.

4.4.8.6 Dezimal zu binär allgemein, auf 1-24 Einzelemente



[_conbitom]



Eingang: = In: Ganzzahlformat

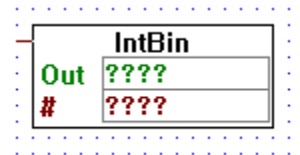
Ausgänge: – O0
Binärformat
– O23

Überträgt den ganzzahligen Wert der an "In" anliegt auf die binären Ausgänge "O0" - "O23". "O0" ist das Bit mit der niedrigsten Wertigkeit.

Ausziehbar von 1 auf 24 Ausgänge.

Für mehr als 24 Bit verwenden Sie bitte die FBox "Dezimal zu binär"

4.4.8.7 Dezimal zu binär, auf Elementreihe (PCD-Format)



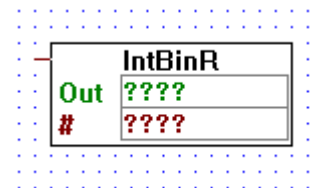
[_conbitoq]

Eingang: = Ganzzahlformat

Ueberträgt einen Ganzzahlwert der am Eingang anliegt an Elemente die bei "Out" definiert sind. Im Eintragsfeld "#" wird die Anzahl Bit (O, F) und im Eintragsfeld "Out" die tiefste Adresse der zusammenhängenden Elementreihe angegeben.

Die tiefste Elementadresse hat die kleinste Wertigkeit (PCD-Format).

4.4.8.8 Dezimal zu binär invers, auf Elementreihe (PCA-Format)



[_conbitoqr]

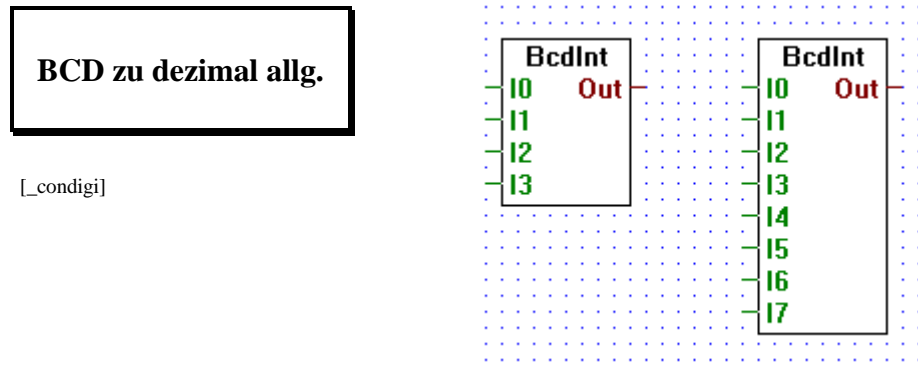
Eingang: = Ganzzahlformat

Ueberträgt die am Eingang anliegende BCD-Information an Elemente (O, F) die bei "Out" bezeichnet sind.

Im Eintragsfeld "Out" wird die tiefste Adresse der zusammenhängenden Elemente, ab welchen die BCD-Information abgelegt werden soll, angegeben. Im Eintragsfeld "#" wird die Anzahl Digits angegeben (1-9).

Die niedrigste Elementadresse ist das Bit mit der grössten Wertigkeit des Digits mit der grössten Wertigkeit. (PCA-Format)

4.4.8.9 BCD zu dezimal allgemein, ab Einzelelementen



Eingänge: – I0
 | Binärformat
 – I7

Ausgang: = Out Ganzzahlformat

Liest 1 oder 2 4-Bit BCD-Digits ab den Eingängen "Ix" und gibt diese am Ausgang "Out" als Ganzzahlwert aus.

Ausziehbar, 4 oder 8 Eingänge

Eingang "I0" ist das Bit mit der kleinsten Wertigkeit.

4.4.8.10 BCD zu dezimal, ab Elementreihe (PCD-Format)



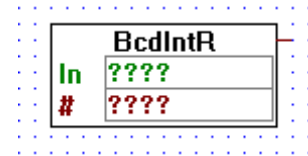
Ausgang: = Ganzzahlformat

Liest Binär Codierte Dezimalwerte (BCD) ab Elementen (I, O, F) und bringt diese als Ganzzahlwert an den Ausgang.

Im Eintragsfeld "In" wird die tiefste Adresse der zusammenhängenden Elemente, ab welchen die BCD-Information eingelesen werden soll, angegeben. Im Eintragsfeld "#" wird die Anzahl Digits angegeben (1-9).

Die niedrigste Elementadresse ist das Bit mit der kleinsten Wertigkeit des Digits mit der kleinsten Wertigkeit. (PCD-Format)

4.4.8.11 BCD zu dezimal invers, ab Elementreihe (PCA-Format)



[_condigiqr]

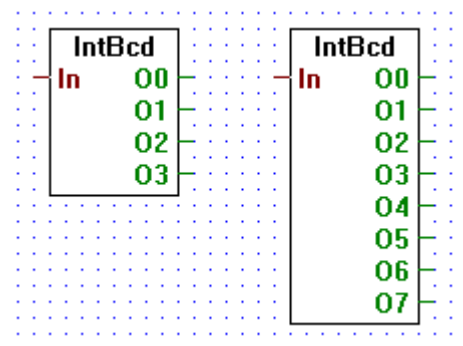
Ausgang: = Ganzzahlformat

Liest Binär Codierte Dezimalwerte (BCD) ab Elementen (I, O, F) und bringt diese als Ganzzahlwert an den Ausgang.

Im Eintragungsfeld "In" wird die tiefste Adresse der zusammenhängenden Elemente, ab welchen die BCD-Information eingelesen werden soll, angegeben. Im Eintragungsfeld "#" wird die Anzahl Digits angegeben (1-9).

Die niedrigste Elementadresse ist das Bit mit der grössten Wertigkeit des Digits mit der grössten Wertigkeit. (PCA-Format)

4.4.8.12 Dezimal zu BCD allgemein, auf Einzelelemente



[_condigo]

Eingang: = In: Ganzzahlformat

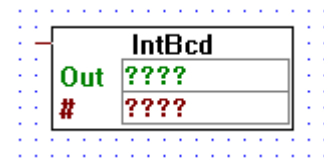
Ausgänge: – 00:
 | Binärformat
 – 07:

Wandelt 1- oder 2-stellige Ganzzahlwerte die an "In" anliegen und gibt diese als 1 oder 2 4-Bit BCD-Digits an den Ausgängen "Ox" aus.

Ausziehbar, 4 oder 8 Ausgänge

Ausgang "00" ist das Bit mit der kleinsten Wertigkeit.

4.4.8.13 Dezimal zu BCD, auf Elementreihe (PCD-Format)



[_condigoq]

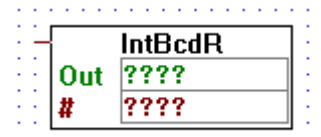
Eingang: = Ganzzahlformat

Überträgt die am Eingang anliegende BCD-Information an Elemente (O, F) die bei "Out" bezeichnet sind.

Im Eintragsfeld "Out" wird die tiefste Adresse der zusammenhängenden Elemente, ab welchen die BCD-Information abgelegt werden soll, angegeben. Im Eintragsfeld "#" wird die Anzahl Digits angegeben (1-9).

Die niedrigste Elementadresse ist das Bit mit der kleinsten Wertigkeit des Digits mit der kleinsten Wertigkeit. (PCD-Format)

4.4.8.14 Dezimal zu BCD invers, auf Elementreihe (PCA-Format)



[_condigoqr]

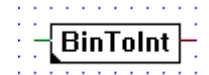
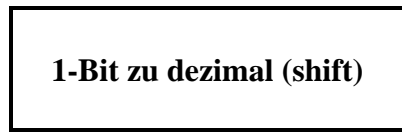
Eingang: = Ganzzahlformat

Überträgt die am Eingang anliegende BCD-Information an Elemente (O, F) die bei "Out" bezeichnet sind.

Im Eintragsfeld "Out" wird die tiefste Adresse der zusammenhängenden Elemente, ab welchen die BCD-Information abgelegt werden soll, angegeben. Im Eintragsfeld "#" wird die Anzahl Digits angegeben (1-9).

Die niedrigste Elementadresse ist das Bit mit der grössten Wertigkeit des Digits mit der grössten Wertigkeit. (PCA-Format)

4.4.8.15 1-Bit zu dezimal



[_conbintoi]

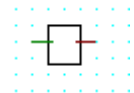
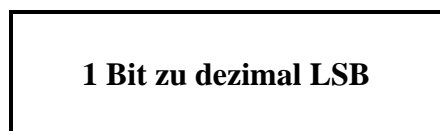
Eingang: – Binärformat
Ausgang: = Ganzzahlformat

Schreibt den binären Wert am Eingang im Dezimalwert am Ausgang an die im Einstllfenster eingetragene Position.

Bit Nummer: Der binäre Eingangswert wird zum angegebenen Bit des dezimalen Ausgangswertes kopiert.



4.4.8.16 1 Bit zu dezimal LSB

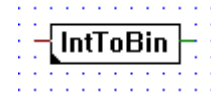


[_conbimove]

Eingang: – Binärformat
Ausgang: = Ganzzahlformat

Am dezimalen Ausgang erscheint eine 0 wenn der Eingang 0 ist, anderenfalls erscheint am Ausgang eine 1.

4.4.8.17 Dezimal zu 1 Bit



[_conitobin]

Eingang: = Ganzzahlformat

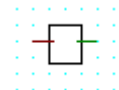
Ausgang: - Binärformat

Schreibt ein Bit des Dezimalwertes am Eingang zum binären Ausgang. Das zum Ausgang zu übertragende Bit wird im Einstellfenster angegeben.

Bit Nummer: Dies ist die Nummer desjenigen Bits des Eingangswertes das am Ausgang erscheinen wird.



4.4.8.18 Dezimal LSB zu 1 Bit



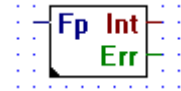
[_conibmove]

Eingang: = Ganzzahlformat

Ausgang: - Binärformat

Der binäre Ausgang wird = L, wenn am dezimalen Eingang Null oder eine andere gerade Zahl anliegt. Anderenfalls wird der Ausgang = H.

4.4.8.19 Fließpunkt- zu Ganzzahlformat



[_confpi2]

Eingang: – Fp: Fließpunktformat

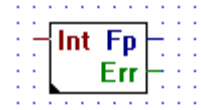
Ausgänge: = Int: Ganzzahlformat
– Err: Binärformat

Wandelt einen Wert im Fließpunktformat am Eingang "FP" in einen Ganzzahlwert am Ausgang "Int".

Das Resultat wird mit dem Exponenten, welcher im Einstellfenster angegeben ist, multipliziert. Liegt am Eingang "FP" z.B. die Zahl 1234.56 bzw. 1.234560E+3 an und im Einstellfenster steht -2, so wird 12 am Ausgang "Int" ausgegeben.

Bei einem Ueberlauf wird der Ausgang "Err" = H. (Error)



4.4.8.20 Ganzzahl- zu Fließpunktformat

[_conifp2]

Eingang: = Int: Ganzzahlformat

Ausgänge: – Fp: Fließpunktformat
 – Err: Binärformat

Wandelt einen Wert im Ganzzahlformat am Eingang "Int" in einen Fließpunktwert am Ausgang "FP".

Das Resultat wird mit dem Exponenten, welcher im Einstellfenster angegeben ist, multipliziert. Liegt am Eingang "Int" z.B. die Zahl 12 an und im Einstellfenster steht 3, so wird 12000.0 bzw. 1.200000E+4 am Ausgang "FP" ausgegeben.

Bei einem Ueberlauf wird der Ausgang "Err" = H. (Error)



Notizen:

4.4.9 Indirekte Adressierung [sfupindi]

Die vorliegende Funktionsfamilie dient der indirekten Adressierung von Einzelementen und Elementreihen. Die hauptsächliche Anwendungen sind:

- Sich wiederholende Funktionen mit unterschiedlichen Adressen
- Auswahl von Parametern in grossen Tabellen
- Ueberall wo Flexibilität bei der Adressierung gefragt ist

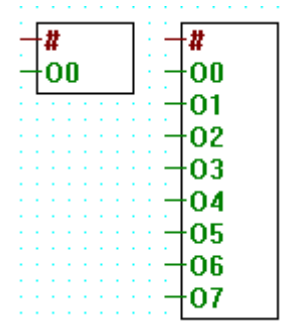
Diese Funktionsfamilie ist sehr universell einsetzbar, verlangt jedoch einiges an Disziplin in der Programmierung, da sich bei zu sorglosem Umgang schwierig auffindbare Adressierungsfehler einschleichen können. Es sind hauptsächlich zu beachten:

- Falscher Wert am Eingang #, vorallem beim Kopieren.
- Adressierungsfehler bei externer Eingabe von Adressen (Leitsystem, Terminal, BCD-Eingabe). Dies kann mit der vorgeschalteten FBox "Limit" und einem kombinierten Alarmsignal verhindert werden.
- Ueberlappung mit den dynamischen Adressenbereichen.
- Timer/Counter Aufteilung (Timer werden zu Countern)
- Ein-/Ausgangsbestückung

4.4.9.1 Sammeln einzelner binärer Informationen und Ausgabe auf aufeinanderfolgende Ausgänge.



[_IndIOO]



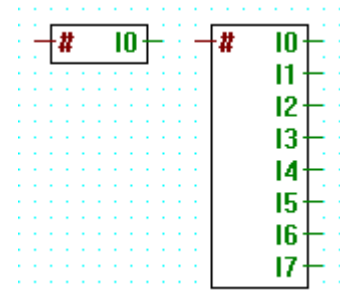
Eingänge: = #: Anfangs-Ausgangsadresse (0 - 8191)
 – 00 Binär-Format
 |
 – 07 Binär-Format

Mit dieser Funktion können 1 bis 8 binäre Einzel-Informationen auf aufeinanderfolgende Ausgänge kopiert werden. Die Adresse des ersten Ausgangs wird am numerischen Eingang # indirekt angegeben.

4.4.9.2 Lesen zusammenhängender Eingänge und Verteilung der binären Information.



[_IndIOI]



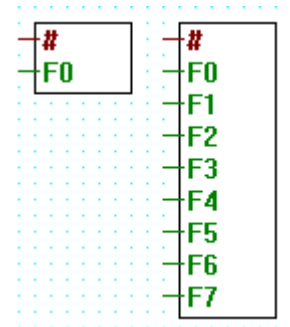
Eingang: = # Anfangs-Eingangsadresse (0 - 8191)
 Ausgänge: – I0 Binär-Format
 |
 – I7 Binär-Format

Mit dieser Funktion können binäre Informationen aus 1 bis 8 aufeinanderfolgenden Eingängen gelesen werden. Die Adresse des ersten Eingangs, ab welchem die Information gelesen werden soll, wird am numerischen Eingang # indirekt angegeben.

4.4.9.3 Sammeln einzelner binärer Informationen und Ausgabe auf aufeinanderfolgende Flags.



[_IndFlgO]



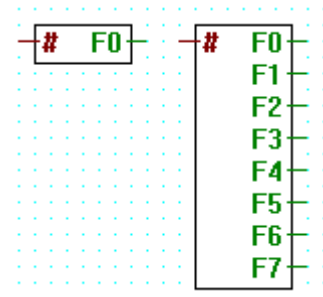
Eingänge: = # Flag Anfangs-Adresse (0 - 8191)
 - F0 Binär-Format
 |
 - F7 Binär-Format

Mit dieser Funktion können 1 bis 8 binäre Einzel-Informationen auf aufeinanderfolgende Flags kopiert werden. Die Adresse des ersten Flags wird am numerischen Eingang # indirekt angegeben.

4.4.9.4 Lesen zusammenhängender Flags und Verteilung der binären Information.



[_IndFlgI]



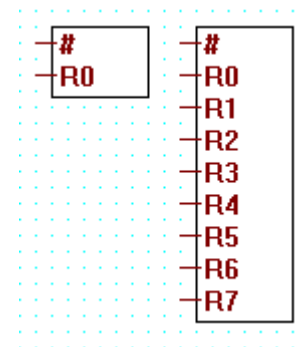
Eingang: = # Flag Anfangs-Adresse (0 - 8191)
 Ausgänge: - F0 Binär-Format
 |
 - F7 Binär-Format

Mit dieser Funktion können binäre Informationen aus 1 bis 8 aufeinanderfolgenden Flags gelesen werden. Die Adresse des ersten Flags, ab welchem die Information gelesen werden soll, wird am numerischen Eingang # indirekt angegeben.

4.4.9.5 Sammeln einzelner numerischer Informationen und Ausgabe auf aufeinanderfolgende Register.



[_IndRegO]



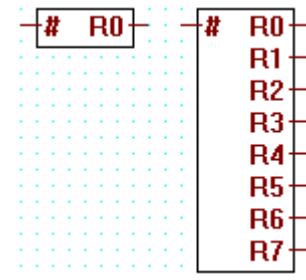
Eingänge: = # Register Anfangs-Adresse (0 - 4095)
 = R0 Ganzzahl-Format
 |
 = R7 Ganzzahl-Format

Mit dieser Funktion können 1 bis 8 numerische Einzelwerte in aufeinanderfolgende Register kopiert werden. Die Adresse des ersten Registers wird am numerischen Eingang # indirekt angegeben.

4.4.9.6 Lesen zusammenhängender Register und Verteilung der numerischen Informationen.



[_IndRegI]



Eingang: = # Register Anfangs-Adresse (0 - 4095)

Ausgänge: = R0 Ganzzahl-Format
 |
 = R7 Ganzzahl-Format

Mit dieser Funktion können numerischen Werte aus 1 bis 8 aufeinanderfolgenden Registern gelesen werden. Die Adresse des ersten Registers wird am numerischen Eingang # indirekt angegeben.

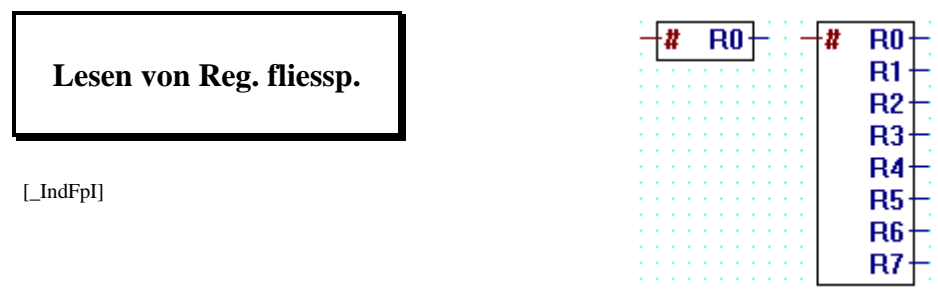
4.4.9.7 Sammeln einzelner Informationen im Fliesspunktformat und Ausgabe auf aufeinanderfolgende Register.



Eingänge: = # Register Anfangs-Adresse (0 - 4095)
 = R0 Fliesspunkt-Format
 |
 = R7 Fliesspunkt-Format

Mit dieser Funktion können 1 bis 8 Einzelwerte im Fliesspunktformat in aufeinanderfolgende Register kopiert werden. Die Adresse des ersten Registers wird am numerischen Eingang # indirekt angegeben.

4.4.9.8 Lesen zusammenhängender Register und Verteilung der Informationen im Fliesspunktformat.



Eingang: = # Register Anfangs-Adresse (0 - 4095)

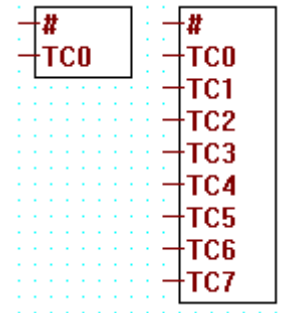
Ausgänge: = R0 Fliesspunkt-Format
 |
 = R7 Fliesspunkt-Format

Mit dieser Funktion können Werte im Fliesspunktformat aus 1 bis 8 aufeinanderfolgenden Registern gelesen werden. Die Adresse des ersten Registers wird am numerischen Eingang # indirekt angegeben.

4.4.9.9 Sammeln einzelner numerischer Informationen und Ausgabe auf aufeinanderfolgende Timer/Counter.



[_IndTCO]



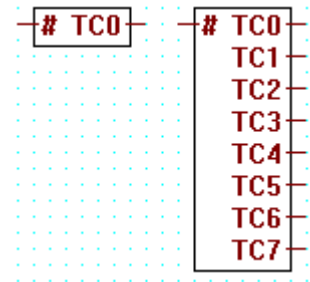
Eingänge: = # T/C Anfangs-Adresse (0 - 1599)
 = TC0 Ganzzahl-Format
 |
 = TC7 Ganzzahl-Format

Mit dieser Funktion können 1 bis 8 numerische Einzelwerte in aufeinanderfolgende Timer oder Counter kopiert werden. Die Adresse des ersten Timer/Counters wird am numerischen Eingang # indirekt angegeben.

4.4.9.10 Lesen zusammenhängender Timer/Counter und Verteilung der numerischen Informationen.



[_IndTCI]

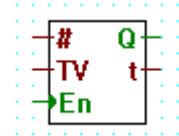


Eingang: = # T/C Anfangs-Adresse (0 - 1599)

Ausgänge: = TC0 Ganzzahl-Format
 |
 = TC7 Ganzzahl-Format

Mit dieser Funktion können numerischen Werte aus 1 bis 8 aufeinanderfolgenden Timern oder Countern gelesen werden. Die Adresse des ersten Timer/Counter wird am numerischen Eingang # indirekt angegeben.

4.4.9.11 Timer mit indirekter Adressierung.



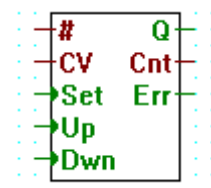
[_IndTmr]

| | | | |
|-----------|-----------------------------------|---------------------------------|-----------------|
| Eingänge: | = # Timer Adresse | = TV Timer Wert in 1/10 sek. | Ganzzahl-Format |
| | > En Enable (Start des Timers) | | Binär-Format |
| Ausgänge: | – Q binärer Timerausgang | | Binär-Format |
| | = t aktueller Timerstand | | Ganzzahl-Format |

Timer mit indirekter Adressierung.

Am numerischen Eingang # wird die Timeradresse indirekt angegeben.

4.4.9.12 Counter mit indirekter Adressierung.



[_IndCnt]

| | | | |
|-----------|---------------------------------|-------------------|-----------------|
| Eingänge: | = # Counter Adresse | = CV Lade-Wert | Ganzzahl-Format |
| | > Set Laden mit "CV" | | Binär-Format |
| | > Up Inkrementieren (+1) | | Binär-Format |
| | > Dwn Dekrementieren (-1) | | Binär-Format |
| Ausgänge: | – Q Ausg. (H, wenn C > 0) | | Binär-Format |
| | = Cnt aktueller Counterstand | | Ganzzahl-Format |
| | – Err Error (Ueberlauf) | | Binär-Format |

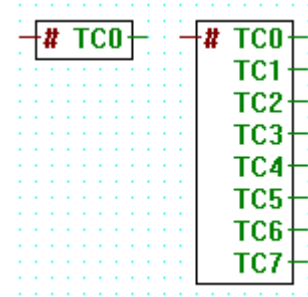
Counter mit indirekter Adressierung

Am numerischen Eingang # wird die Counteradresse indirekt angegeben.

4.4.9.13 Lesen der logischen Zustände von Timern/Countern.



[_IndTCL]



Eingang: = # T/C Anfangs-Adresse (0 - 1599)

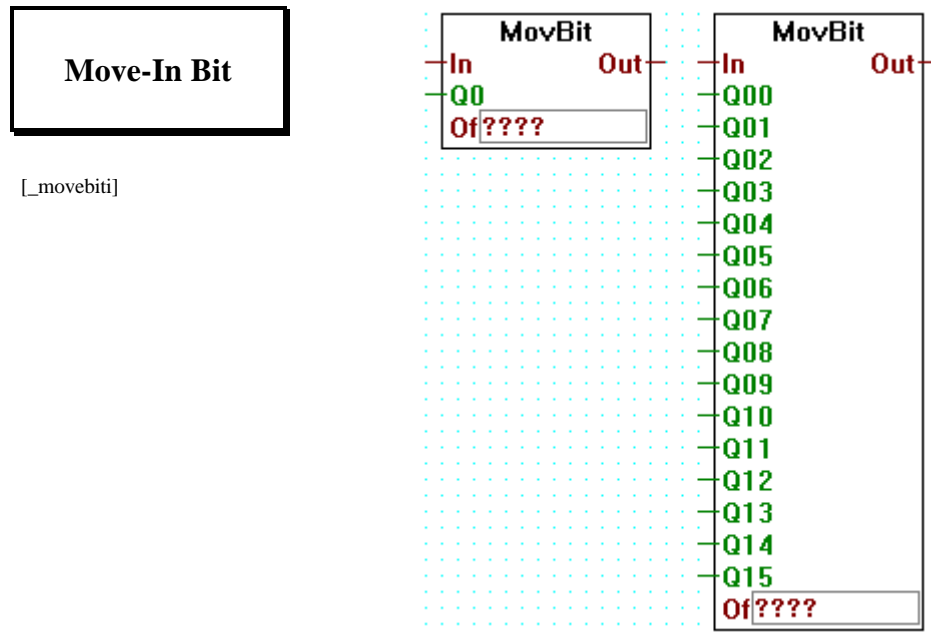
Ausgänge: – TC0 H/L Timer/Counter Binär-Format
 |
 – TC7 H/L Timer/Counter Binär-Format

Lesen der logischen Zustände von Timern/Countern.

Mit dieser Funktion können die logischen Zustände von 1 bis 8 aufeinanderfolgenden Timern oder Countern gelesen werden. Die Adresse des ersten Timer/Counters wird am numerischen Eingang # indirekt angegeben.

4.4.10 Move Data [sfupmove]

4.4.10.1 Move-In Bit



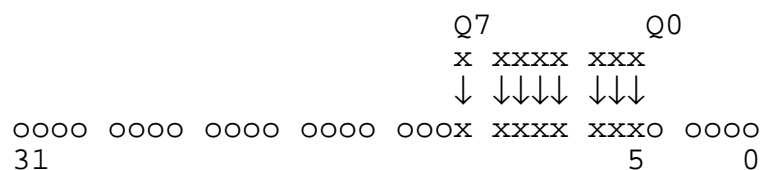
Eingänge: = In: Ganzzahlwert
 – Q00: Binärwert (Q = Quantum) (0 oder 1)
 |
 – Q15: Binärwert
 = Of: Offset, Ganzzahlwert (0 -31)

Ausgang: = Out: Ganzzahlwert

Am numerischen Eingang "In" liegt die Information (Register oder Konstante) welche mit der Binärinformation an den Eingängen "Q0"..."Q.." (max. "Q15") kombiniert werden soll.

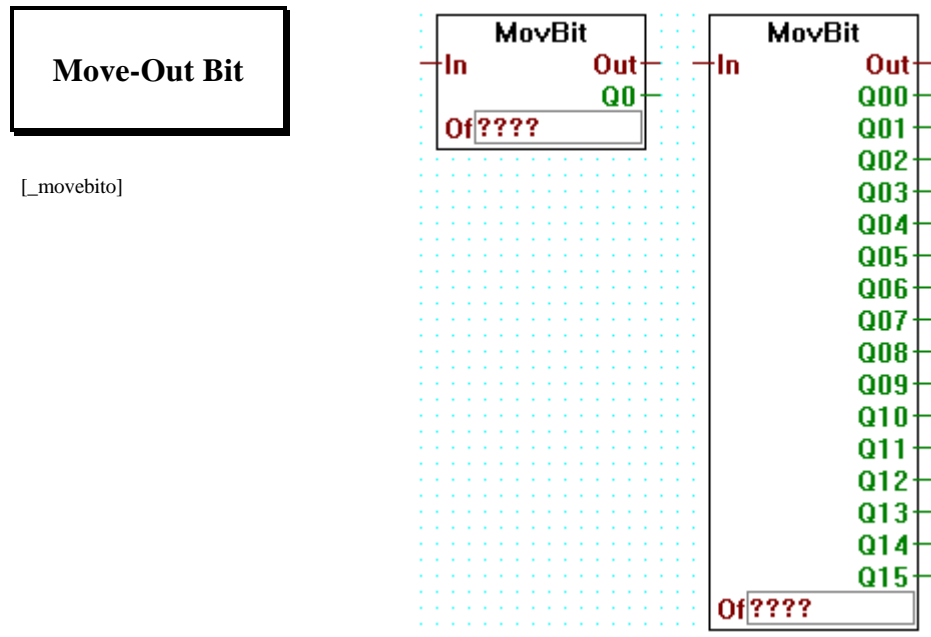
Im Eintragsfeld "Of" (Offset) wird die Bitstelle angegeben, ab wo die Binärinformation ins Register abgelegt werden soll. Wird z.B. <5> angegeben, bedeutet dies, dass die an den Eingängen "Q0" bis z.B. "Q7" anliegende Information auf die Bit 5 bis Bit 12 im Register abgelegt wird, wobei Q0 das LSB ist. Die anderen Bits im Register, für das erwähnte Beispiel Bit 0...4 und 13...31, werden NICHT verändert.

nach "Move-In Bit" mit "Of" = 5:



Am numerischen Ausgang "Out" erscheint der kombinierte Registerinhalt.

4.4.10.2 Move-Out Bit



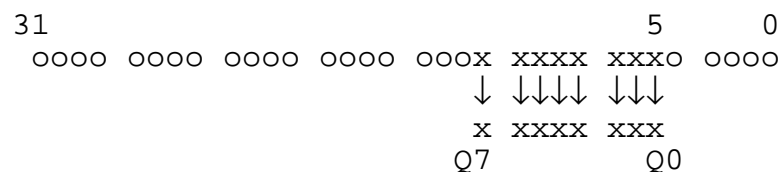
Eingänge: = In: Ganzzahlwert
 = Of: Offset, Ganzzahlwert (0 -31)

Ausgänge: = Out: Ganzzahlwert
 – Q00: Binärwert (Q = Quantum) (0 oder 1)
 |
 – Q15: Binärwert

Verschiebt eine Reihe von Bits aus einem Register, dessen Wert am numerischen Eingang "In" anliegt, an die Ausgänge "Q0" bis "Q.." (max. "Q15").

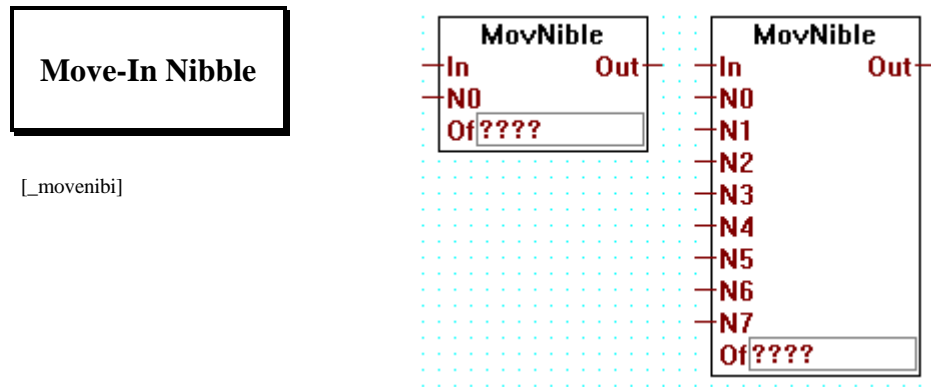
Im Eintragsfeld "Of" (Offset) wird die Bitstelle im Register angegeben, welche als LSB an Q0 liegen soll. Wird z.B. <5> angegeben, bedeutet dies, dass die im Register auf Bit 5 bis z.B. Bit 12 liegende Information an die Ausgänge Q0...Q7 zu liegen kommt.

"Move-Out Bit", "Of" = 5:



Am numerischen Ausgang "Out" erscheint der unveränderte Wert des Eingangs "In".

4.4.10.3 Move-In Nibble (4 Bit binär)



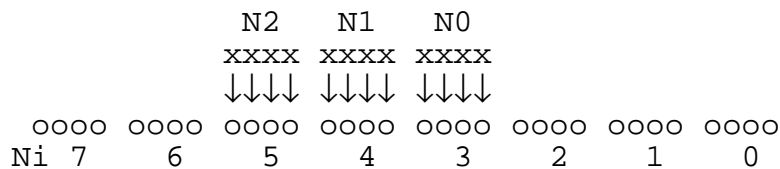
Eingänge: = In: Ganzzahlwert
 = N0: Ganzzahlwert (0 - 15)
 |
 = N7: Ganzzahlwert
 = Of: Offset, Ganzzahlwert (0 - 7)

Ausgang: = Out: Ganzzahlwert

Am numerischen Eingang "In" liegt die Information (Register oder Konstante) welche mit der Nibbleinformation an den Eingängen "N0"... "N.." (max. "N7") kombiniert werden soll.

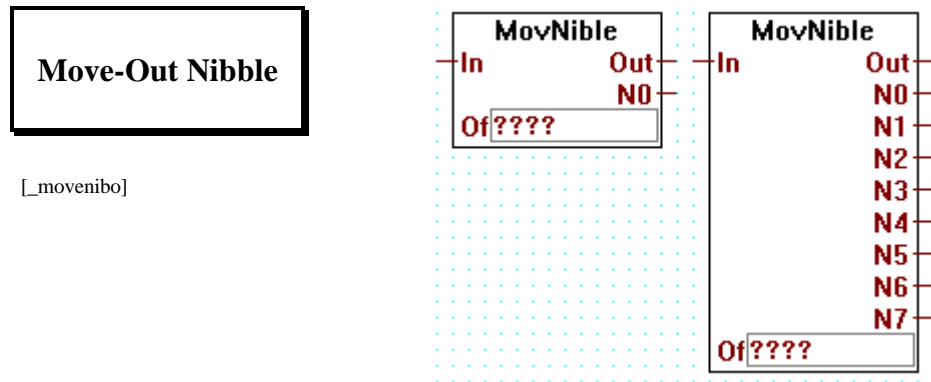
Im Eintragungsfeld "Of" (Offset) wird die Nibblestelle angegeben, ab wo die Nibbleinformation ins Register abgelegt werden soll. Wird z.B. <3> angegeben, bedeutet dies, dass die an den Eingängen "N0" bis z.B. "N2" anliegende Information auf die Nibbles 2 bis Nibble 4 im Register abgelegt wird, wobei N0 das LS-Nibble ist. Die anderen Nibbles im Register, für das erwähnte Beispiel Nibbles 0...2 und 6...7, werden NICHT verändert.

nach "Move-In Nibbles" mit "Of" = 3:



Am numerischen Ausgang "Out" erscheint der kombinierte Registerinhalt.

4.4.10.4 Move-Out Nibble (4 Bit binär)

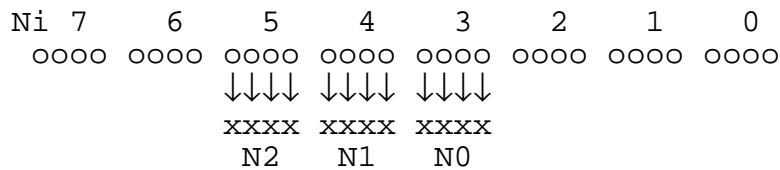


- Eingänge: = In: Ganzzahlwert
 = Of: Offset, Ganzzahlwert (0 - 7)
- Ausgänge: = Out: Ganzzahlwert
 = N0: Ganzzahlwert (0 - 15)
 |
 = N7: Ganzzahlwert

Verschiebt eine Reihe von Nibbles aus einem Register, dessen Wert am numerischen Eingang "In" anliegt, an die Ausgänge "N0" bis "N.." (max. "N7").

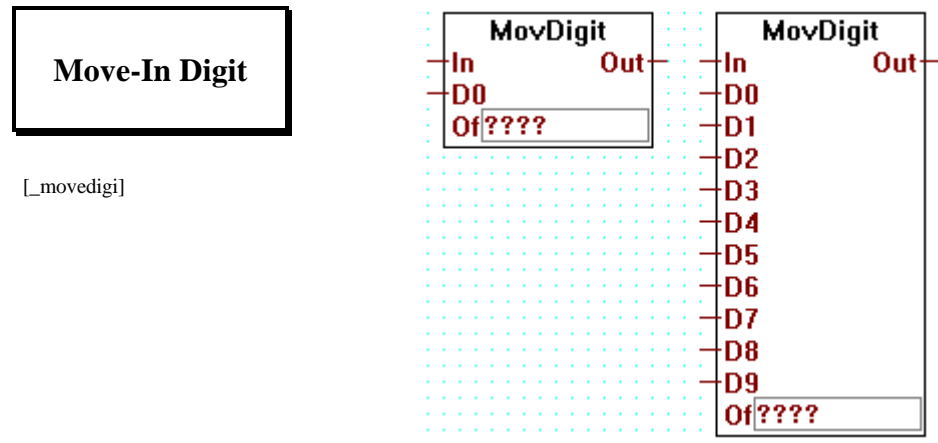
Im Eintragsfeld "Of" (Offset) wird die Nibblestelle im Register angegeben, welche als LS-Nibble an N0 liegen soll. Wird z.B. <3> angegeben, bedeutet dies, dass die im Register auf Nibble 3 bis z.B. Nibble 5 liegende Information an die Ausgänge N0...N2 zu liegen kommt.

"Move-Out Nibble", "Of" = 3:



Am numerischen Ausgang "Out" erscheint der unveränderte Wert des Eingangs "In".

4.4.10.5 Move-In Digit (4 Bit BCD)



Eingänge: = In: Ganzzahlwert
 = D0: Ganzzahlwert (0 - 9)
 |
 = D9: Ganzzahlwert
 = Of: Offset, Ganzzahlwert (0 - 9)

Ausgang: = Out: Ganzzahlwert

Am numerischen Eingang "In" liegt die Information (Register oder Konstante) welche mit der Digitinformation an den Eingängen "D0"... "D.." (max. "D9") kombiniert werden soll.

Im Eintragsfeld "Of" (Offset) wird die Digitstelle angegeben, ab wo die Digitinformation ins Register abgelegt werden soll. Wird z.B. <3> angegeben, bedeutet dies, dass die an den Eingängen "D0" bis z.B. "D2" anliegende Information auf die Digits 3 bis Digit 5 im Register abgelegt wird, wobei D0 das LS-Digit ist. Die anderen Digits im Register, für das erwähnte Beispiel Digits 0...2 und 6...9, werden NICHT verändert.

nach "Move-In Digit" mit "Of" = 3:

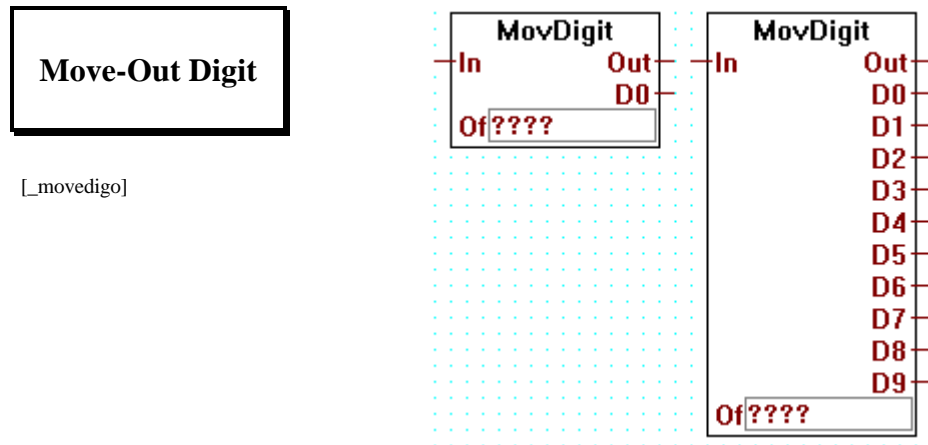
| | | | | | | | | | | |
|-------|---|---|---|---|----|----|----|---|---|---|
| | | | | | D2 | D1 | D0 | | | |
| | | | | | x | x | x | | | |
| | | | | | ↓ | ↓ | ↓ | | | |
| Digit | ○ | ○ | ○ | ○ | x | x | x | ○ | ○ | ○ |
| | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Es können nur positive Werte verarbeitet werden.

Der max. Wert für Digit 9 ist 2.

Am numerischen Ausgang "Out" erscheint der kombinierte Registerinhalt.

4.4.10.6 Move-Out Digit (4 Bit BCD)



- Eingänge: = In: Ganzzahlwert
 = Of: Offset, Ganzzahlwert (0 - 9)
- Ausgänge: = Out: Ganzzahlwert
 = D0: Ganzzahlwert (0 - 9)
 |
 = D9: Ganzzahlwert

Verschiebt eine Reihe von Digits aus einem Register, dessen Wert am numerischen Eingang "In" anliegt, an die Ausgänge "D0" bis "D.." (max. "D9").

Im Eintragfeld "Of" (Offset) wird die Digitstelle im Register angegeben, welche als LS-Digit an N0 liegen soll. Wird z.B. <3> angegeben, bedeutet dies, dass die im Register auf Digit 3 bis z.B. Digit 5 liegende Information an die Ausgänge D0...D2 zu liegen kommt.

"Move-Out Digit", "Of" = 3:

| | | | | | | | | | | |
|-------|---|---|---|---|---|----|----|----|---|---|
| Digit | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | o | o | o | o | x | x | x | o | o | o |
| | | | | | ↓ | ↓ | ↓ | | | |
| | | | | | x | x | x | | | |
| | | | | | | D2 | D1 | D0 | | |

Es können nur positive Werte verarbeitet werden.

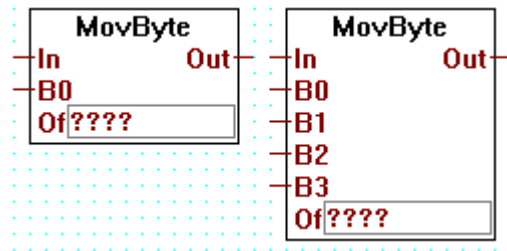
Der max. Wert für Digit 9 ist 2.

Am numerischen Ausgang "Out" erscheint der unveränderte Wert des Eingangs "In".

4.4.10.7 Move-In Byte (8 Bit)



[_movebyti]



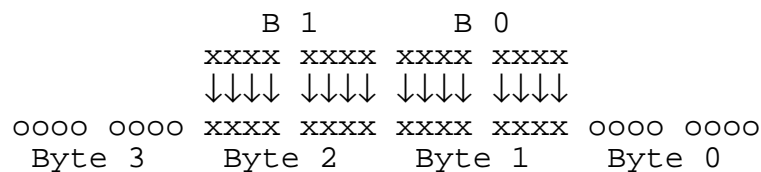
- Eingänge: = In: Ganzzahlwert
- = B0: Ganzzahlwert (0 - 255)
- |
- = B3: Ganzzahlwert
- = Of: Offset, Ganzzahlwert (0 - 3)

Ausgang: = Out: Ganzzahlwert

Am numerischen Eingang "In" liegt die Information (Register oder Konstante) welche mit der Byteinformation an den Eingängen "B0"..."B.." (max. "B3") kombiniert werden soll.

Im Eintragsfeld "Of" (Offset) wird die Bytestelle angegeben, ab wo die Byteinformation ins Register abgelegt werden soll. Wird z.B. <1> angegeben, bedeutet dies, dass die an den Eingängen "B0" bis z.B. "B1" anliegende Information auf die Byte 1 bis Byte 2 im Register abgelegt wird, wobei B0 das LS-Byte ist. Die anderen Bytes im Register, für das erwähnte Beispiel Byte 0 und Byte 3, werden NICHT verändert.

nach "Move-In Byte" mit "Of" = 1:

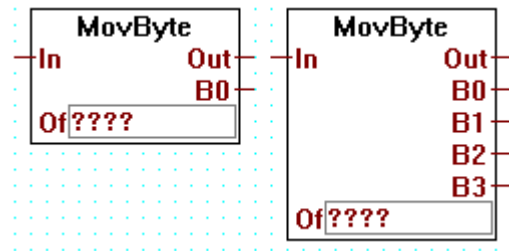


Am numerischen Ausgang "Out" erscheint der kombinierte Registerinhalt.

4.4.10.8 Move-Out Byte (8 Bit)



[_movebyte]



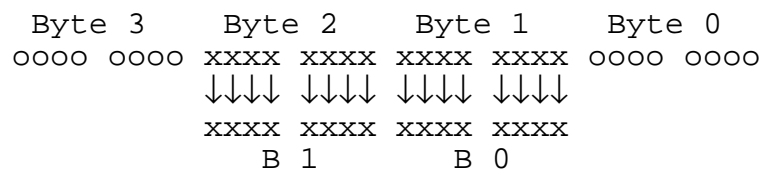
Eingänge: = In: Ganzzahlwert
 = Of: Offset, Ganzzahlwert (0 - 3)

Ausgänge: = Out: Ganzzahlwert
 = B0: Ganzzahlwert (0 - 255)
 |
 = B3: Ganzzahlwert (0 - 255)

Verschiebt eine Reihe von Bytes aus einem Register, dessen Wert am numerischen Eingang "In" anliegt, an die Ausgänge "B0" bis "B.." (max. "B3").

Im Eintragsfeld "Of" (Offset) wird die Bytestelle im Register angegeben, welche als LS-Byte an B0 liegen soll. Wird z.B. <1> angegeben, bedeutet dies, dass die im Register auf Byte 1 bis z.B. Byte 2 liegende Information an die Ausgänge B0...B1 zu liegen kommt.

"Move-Out Byte", "Of" = 1:

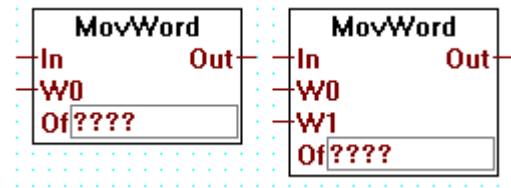


Am numerischen Ausgang "Out" erscheint der unveränderte Wert des Eingangs "In".

4.4.10.9 Move-In Word (16 Bit)



[_movewori]



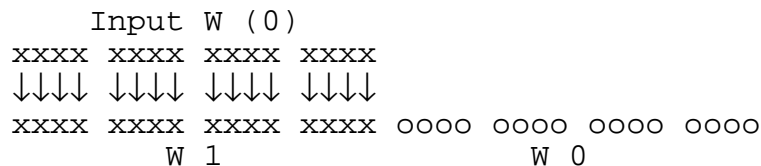
Eingänge: = In: Ganzzahlwert
 = W0: Ganzzahlwert (0 - 65535)
 |
 = W1: Ganzzahlwert
 = Of: Offset, Ganzzahlwert (0 - 1)

Ausgang: = Out: Ganzzahlwert

Am numerischen Eingang "In" liegt die Information (Register oder Konstante) welche mit der Wordinformation an den Eingängen "W0"..."W.." (max. "W1") kombiniert werden soll.

Im Eintragsfeld "Of" (Offset) wird die Wordstelle angegeben, ab wo die Wordinformation ins Register abgelegt werden soll. Wird z.B. <1> angegeben, bedeutet dies, dass die am Eingang "W (0)" anliegende Information auf das Word 1 im Register abgelegt wird, wobei W0 das LS-Word ist. Das andere Word im Register, für das erwähnte Beispiel Word 0, wird NICHT verändert.

nach "Move-In Word" mit "Of" = 1:

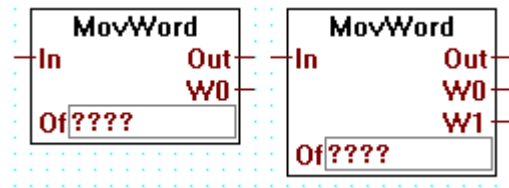


Am numerischen Ausgang "Out" erscheint der kombinierte Registerinhalt.

4.4.10.10 Move-Out Word (16 Bit)



[_moveworo]



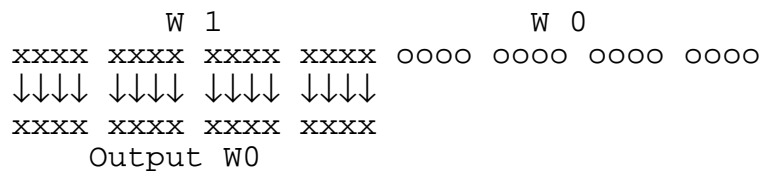
Eingänge: = In: Ganzzahlwert
 = Of: Offset, Ganzzahlwert (0 - 1)

Ausgänge: = Out: Ganzzahlwert
 = W0: Ganzzahlwert (0 - 65535)
 |
 = W1: Ganzzahlwert

Verschiebt eine Reihe von Words aus einem Register, dessen Wert am numerischen Eingang "In" anliegt, an die Ausgänge "W0" bis "W.." (max. "W1").

Im Eintragsfeld "Of" (Offset) wird die Wordstelle im Register angegeben, welche als LS-Word an W0 liegen soll. Wird z.B. <1> angegeben, bedeutet dies, dass die im Register in Word 1 liegende Information an den Ausgang W0 zu liegen kommt.

"Move-Out Word", "Of" = 1:

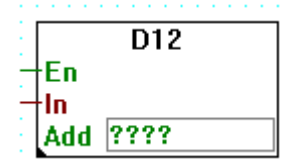


Am numerischen Ausgang "Out" erscheint der unveränderte Wert des Eingangs "In".

4.4.11 Anzeigen (Displays)

[sfupdisp]

4.4.11.1 Displaymodul PCA2.D12



[_dispd12]

Eingänge: – En: Enable
 = In: Ganzzahlformat

Bringt einen 4-stelligen Ganzzahlwert an die Anzeige eines PCA2.D12-Moduls.

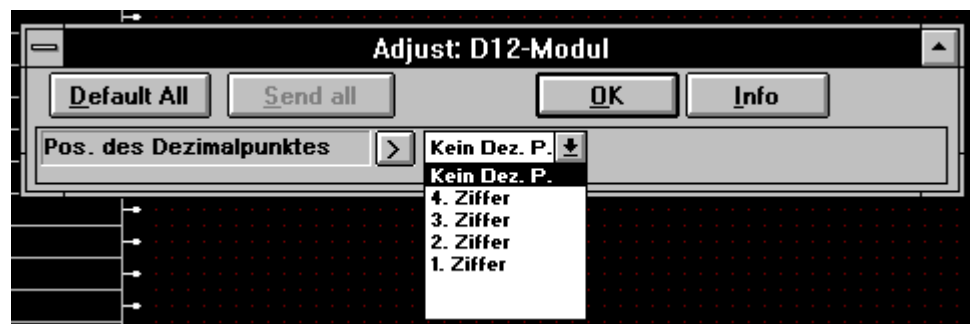
Liegt am Eingang "En" (Enable) ein H-Signal, werden die vier niederwertigsten Digits, die am Eingang "In" anliegen, zur Anzeige gebracht.

Im Eintragsfeld "Add" wird die niedrigste Adresse der 3 aufeinanderfolgenden PCD-Ausgänge, an welchen das Displaymodul angeschlossen ist, angegeben.

Add + 0 = Clock
 Add + 1 = Data
 Add + 2 = Enable

Wird das Eingangssignal "En" = L, bleibt die letzte Anzeige erhalten.

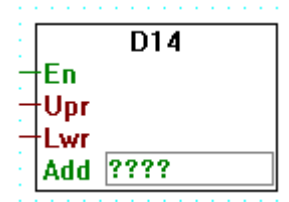
Im Einstellfenster kann die Position des Dezimalpunkts gewählt werden.



4.4.11.2 Displaymodul PCA2.D14



[_dispd14]



Eingänge: – En: Enable
 = Upr: Upper display, obere Anzeige - Ganzzahlformat
 = Lwr: Lower display, untere Anzeige.- Ganzzahlformat

Bringt zwei 6-stellige Ganzzahlwerte an die Anzeigen eines PCA2.D14-Moduls.

Liegt am Eingang "En" (Enable) ein H-Signal, werden je die sechs niederwertigsten Digits, die an den Eingängen "Upr" und "Lwr" anliegen, zur Anzeige gebracht. Die Information am Eingang "Upr" (Upper) kommt auf die obere, die Information am Eingang "Lwr" (Lower) kommt auf die untere Anzeige.

Im Eintragsfeld "Add" wird die niedrigste Adresse der 3 aufeinanderfolgenden PCD-Ausgänge, an welchen das Displaymodul angeschlossen ist, angegeben.

Add + 0 = Clock
 Add + 1 = Data
 Add + 2 = Enable

Wird das Eingangssignal "En" = L, bleiben die letzten Anzeigen erhalten.

4.4.11.3 PCD2 F510 für numerische Anzeigen



[_disppcd23]

Eingänge: – Enable Binärformat
 = Anzeigender Wert Ganzzahlformat

Ansteuerung des PCD2.F510-Displays für numerische Anzeigen.

Es wird im allgemeinen der Ganzzahlwert, der an die Funktionsbox angelegt ist, angezeigt. (-99999 ... 999999).

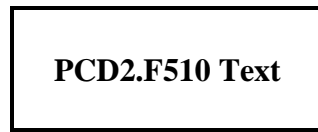
Es können auch vordefinierte numerische Ausdrücke angezeigt werden. Die Wahl der Anzeige geschieht im Einstellfenster.

Ist der Enable Eingang "En" = L, bleibt das letzt definierte Format erhalten. Die Standardeinstellung ist "6 Ziffern"



| Einstellung | Anzeige |
|--------------|---|
| 6 Ziffern | Die 6 Ziffern die angelegt sind. Liegt der Wert ausserhalb von -99'999 und 999'999 wird die Anzeige gelöscht. |
| • Hlp nn | 'HLP nn' (nn = 2 Ziffern 00-99) |
| • Err nn | 'Err nn' (nn = 2 Ziffern 00-99) liegt als Ganzzahlwert am Eingang der F-Box. Liegt der Wert ausserhalb von 00 - 99 wird die ganze Anzeige gelöscht. |
| • Führende 0 | Anzeige mit führenden Nullen. |
| • 2 Ziffern | Die 2 Ziffern die angelegt sind. Liegt der Wert ausserhalb von 00 - 99 wird die ganze Anzeige gelöscht. |
| • Uhr | Anzeige der angeschlossenen Uhr (FBox: Uhr) im Format hh.mm.ss. |

4.4.11.4 PCD2 F510 für Text-Anzeigen



[_disppcd24]

Eingang: – Enable Binärformat

Ansteuerung des PCD2.F510-Displays für Textausgaben.

Es werden vordefinierte Texte angezeigt.. Die Wahl der Texte geschieht im Einstellfenster.

Ist der Enable Eingang "En" = L, bleibt das letzt definierte Format erhalten. Die Standardeinstellung ist "=SAIA="

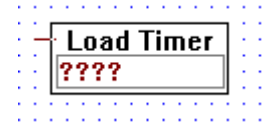


| Einstellung | Anzeige |
|-------------|---------------------------|
| =SAIA= | '=SAIA=' |
| =PCd2= | '=PCd2=' |
| Error | ' Error' |
| Blank | Die Anzeige wird gelöscht |

4.4.12 GRAFTEC-Funktionen

sfupgraf]

4.4.12.1 Lade Timer ohne Bedingung



[_grafltdt]

Eingang: = Lade-Wert: Register oder Konstante (Ganzzahlformat)

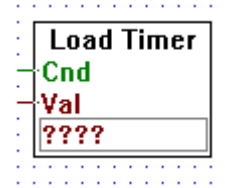
FBox zum Laden und Aktivieren (Starten) eines Timers (Zeitgliedes) in einem STEP.

Im Eintragsfeld ist entweder ein symbolischer Name oder die absolute Timer-Adresse einzutragen. (Symbolische Namen sind zu bevorzugen, da die Verwaltung automatisch erfolgt).

Der zu ladende Wert ist am numerischen Eingang anzulegen.

Um den Inhalt des Timers ONLINE verfolgen zu können ist der symbolische Namen oder die absolute Timer-Adresse in ein Eingabefeld am linken Bildschirmrand des FUPLA einzutragen. Im ONLINE-Betrieb kann an dieses Eingabefeld eine Online-Probe (Online-Kästchen) angebracht werden.

4.4.12.2 Lade Timer bedingt



[_grafeldt]

Eing.: – Cnd: Condition (Bedingung) H/L, (Binärformat)
 = Val: Lade-Wert: Register oder Konstante (Ganzzahlformat)

FBox zum bedingten Laden und Aktivieren (Starten) eines Timers (Zeitgliedes) in einem STEP.

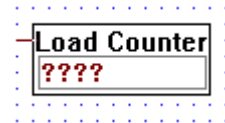
Im Eintragsfeld ist entweder ein symbolischer Name oder die absolute Timer-Adresse einzutragen. (Symbolische Namen sind zu bevorzugen, da die Verwaltung automatisch erfolgt).

Der zu ladende Wert ist am numerischen Eingang anzulegen.

Die Funktion wird nur ausgeführt, wenn am binären Eingang "Cnd" ein H-Signal anliegt.

Um den Inhalt des Timers ONLINE verfolgen zu können ist der symbolische Namen oder die absolute Timer-Adresse in ein Eingabefeld am linken Bildschirmrand des FUPLA einzutragen. Im ONLINE-Betrieb kann an dieses Eingabefeld eine Online-Probe (Online-Kästchen) angebracht werden.

4.4.12.3 Lade Zähler ohne Bedingung



[_grafldc]

Eingang: = lade Wert: Register oder Konstante (Ganzzahlformat)

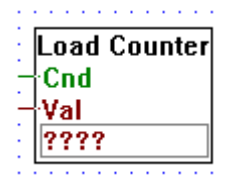
FBox zum Laden eines Counters (Zählers) in einem STEP.

Im Eintragsfeld ist entweder ein symbolischer Name oder die absolute Counter-Adresse einzutragen. (Symbolische Namen sind zu bevorzugen, da die Verwaltung automatisch erfolgt).

Der zu ladende Wert ist am numerischen Eingang anzulegen.

Um den Inhalt des Counters ONLINE verfolgen zu können ist der symbolische Namen oder die absolute Counter-Adresse in ein Eingabefeld am linken Bildschirmrand des FUPLA einzutragen. Im ONLINE-Betrieb kann an dieses Eingabefeld eine Online-Probe angebracht werden.

4.4.12.4 Lade Zähler bedingt



[_grafeldc]

Eing.: – Cnd: Condition (Bedingung) H/L, (Binärformat)
= Val: Lade-Wert: Register oder Konstante (Ganzzahlformat)

FBox zum bedingten Laden eines Counters (Zählers) in einem STEP.

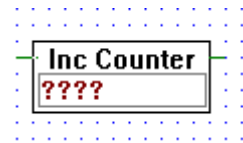
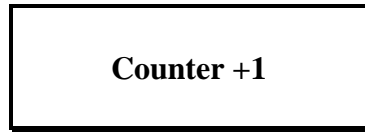
Im Eintragsfeld ist entweder ein symbolischer Name oder die absolute Counter-Adresse einzutragen. (Symbolische Namen sind zu bevorzugen, da die Verwaltung automatisch erfolgt).

Der zu ladende Wert ist am numerischen Eingang anzulegen.

Die Funktion wird nur ausgeführt, wenn am binären Eingang "Cnd" ein H-Signal anliegt.

Um den Inhalt des Counters ONLINE verfolgen zu können ist der symbolische Namen oder die absolute Counter-Adresse in ein Eingabefeld am linken Bildschirmrand des FUPLA einzutragen. Im ONLINE-Betrieb kann an dieses Eingabefeld eine Online-Probe (Online-Kästchen) angebracht werden.

4.4.12.5 Inkrementiere Zähler



[_grafincc]

Eingang: – Bedingung zum Inkrementieren: H/L (Binärformat)
 Ausgang: – Logischer Zustand des Counters: H/L (Binärformat)

FBox zum Inkrementieren eines Counters (+1), normalerweise in einem STEP.

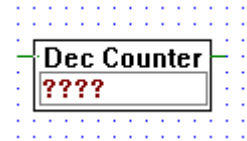
Der Counter muss zuvor mit der Funktion "Load Counter" geladen worden sein.

Im Eintragsfeld ist der dem "Load Counter"-Befehl korrespondierende Name bzw. die Counter-Adresse anzugeben.

Das Inkrementieren erfolgt nur, wenn die Eingangsbedingung erfüllt ist.

Um den Inhalt des Counters ONLINE verfolgen zu können ist der symbolische Namen oder die absolute Counter-Adresse in ein Eingabefeld am linken Bildschirmrand des FUPLA einzutragen. Im ONLINE-Betrieb kann an dieses Eingabefeld eine Online-Probe (Online-Kästchen) angebracht werden.

4.4.12.6 Dekrementiere Zähler



[_grafdecc]

Eingang: – Bedingung zum Dekrementieren: H/L (Binärformat)
Ausgang: – Logischer Zustand des Counters: H/L (Binärformat)

FBox zum Dekrementieren eines Counters (-1), normalerweise in einem STEP.

Der Counter muss zuvor mit der Funktion "Load Counter" geladen worden sein.

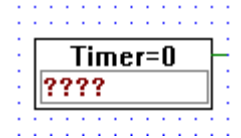
Im Eintragsfeld ist der dem "Load Counter"-Befehl korrespondierende Name bzw. die Counter-Adresse anzugeben.

Das Dekrementieren erfolgt nur, wenn die Eingangsbedingung erfüllt ist.

Um den Inhalt des Counters ONLINE verfolgen zu können ist der symbolische Namen oder die absolute Counter-Adresse in ein Eingabefeld am linken Bildschirmrand des FUPLA einzutragen. Im ONLINE-Betrieb kann an dieses Eingabefeld eine Online-Probe (Online-Kästchen) angebracht werden.

4.4.12.7 Ist Timer abgelaufen ?

[_graftisl]



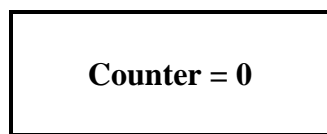
Ausgang: – H, wenn Timer abgelaufen. (Binärformat)

FBox zur Abfrage des logischen Zustandes eines Timers in einer TRANSITION. Ist der Timer abgelaufen, ist die Funktion erfüllt. Steht diese FBox allein in einer TRANSITION, wird nach Ablauf des Timers zum nächsten STEP weitergeschaltet.

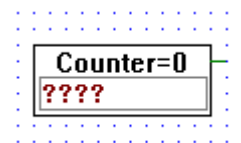
Die Funktion kann, vorzugsweise mit Kontaktplan-Symbolen, verknüpft werden.

Im Eintragsfeld ist der dem "Load Timer"-Befehl korrespondierende Name bzw. die Timer-Adresse anzugeben.

Um den Inhalt des Timers ONLINE verfolgen zu können ist der symbolische Namen oder die absolute Timer-Adresse in ein Eingabefeld am linken Bildschirmrand des FUPLA einzutragen. Im ONLINE-Betrieb kann an dieses Eingabefeld eine Online-Probe (Online-Kästchen) angebracht werden.

4.4.12.8 Ist Zähler = 0 ?

[_grafcisl]



Ausgang: – H, wenn Counter = 0. (Binärformat)

FBox zur Abfrage des logischen Zustandes eines Counters in einer TRANSITION. Ist der Counter = 0, ist die Funktion erfüllt.

Die Funktion kann, vorzugsweise mit Kontaktplan-Symbolen, verknüpft werden.

Im Eintragsfeld ist der dem "Load Counter"-Befehl korrespondierende Name bzw. die Counter-Adresse anzugeben.

4.4.12.9 Ende Transition



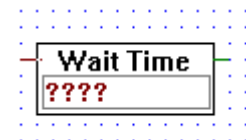
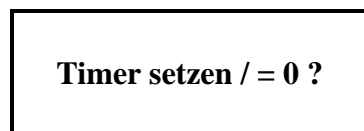
[_grafetrsv]

Eingang: – Binärformat

Diese FBox ist in jede TR einzufügen.

Wird der binäre Eingang der FBox = H, wird die TRANSITION richtung nächster STEP verlassen.

4.4.12.10 Timer setzen und Zeit abwarten



[_grafwait]

Eingang: = Zeit-Wert: Register oder Konstante (Ganzzahlformat)

Ausgang: – H, wenn Timer abgelaufen. (Binärformat)

Kombinierte FBox zum Programmieren einer Pause in einer TR. Bei der ersten Abarbeitung der TR wird der Timer mit dem Eingangswert geladen und gestartet. Der Ausgang wird = H, wenn der Timer abgelaufen ist.

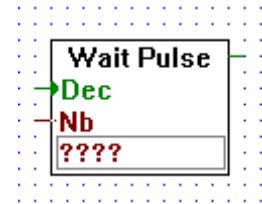
Im Eintragsfeld ist entweder ein symbolischer Name oder die absolute Timer-Adresse einzutragen. (Symbolische Namen sind zu bevorzugen, da die Verwaltung automatisch erfolgt).

Um den Inhalt des Timers ONLINE verfolgen zu können, ist der symbolische Namen oder die absolute Timer-Adresse in ein Eingabefeld am linken Bildschirmrand des FUPLA einzutragen. Im ONLINE-Betrieb kann an dieses Eingabefeld eine Online-Probe (Online-Kästchen) angebracht werden.

4.4.12.11 Counter setzen und warten bis Counter = 0



[_grafwpls]



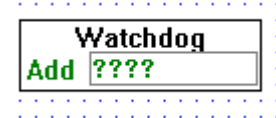
Eingänge: > Dec: Dekrement-Signal (Binärformat)
= Nb: Register oder Konstante (Ganzzahlformat)

Ausgang: – H, wenn Counter = 0. (Binärformat)

Kombinierte FBox zum Programmieren einer Pause in einer TR. Die Pause endet nach dem Eintreffen einer vorgewählten Anzahl Impulse. Bei der ersten Abarbeitung der TR wird der Counter mit dem Eingangswert geladen. Jeder Impuls am Eingang "Dec" dekrementiert den Counter um einen Wert. Der Ausgang wird = H, wenn der Counter den Wert 0 erreicht hat.

Im Eintragsfeld ist entweder ein symbolischer Name oder die absolute Counter-Adresse einzutragen. (Symbolische Namen sind zu bevorzugen, da die Verwaltung automatisch erfolgt).

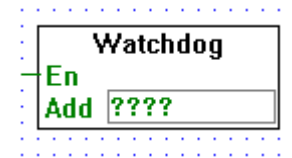
Um den Inhalt des Counters ONLINE verfolgen zu können, ist der symbolische Namen oder die absolute Counter-Adresse in ein Eingabefeld am linken Bildschirmrand des FUPLA einzutragen. Im ONLINE-Betrieb kann an dieses Eingabefeld eine Online-Probe (Online-Kästchen) angebracht werden.

4.4.13 Spezialfunktionen (Watch Dog)**[sfupspec]****4.4.13.1 Watch-dog (dauernd aktiv)**

[_watchdog]

Diese Funktion aktiviert den Watch Dog einer PCD wenn die CPU in RUN läuft. Als Elementadresse "Add" ist normalerweise "O 255" anzugeben. (siehe Hardware-Handbücher PCD2, PCD4, PCD6).

Die Funktion muss mindestens 5 mal pro Sekunde aktiviert werden.

4.4.13.2 Watch-Dog ein-/ausschaltbar (Enable)

[_watchdoga]

Eingang: – En Enable (Binärformat)

Ist der Enable-Eingang (En) = H und die CPU in RUN, wird der Watch Dog der PCD aktiviert. Als Elementadresse "Add" ist normalerweise "O 255" anzugeben. (siehe Hardware-Handbücher PCD2, PCD4, PCD6).

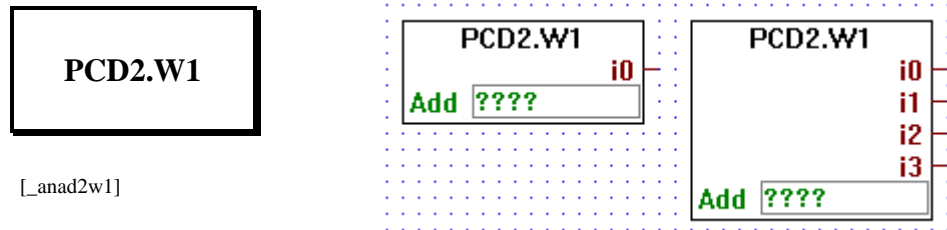
Die Funktion muss mindestens 5 mal pro Sekunde aktiviert werden.

Notizen:

4.4.14 Analog Module

[sfupanlg]

4.4.14.1 Analoges Eingangsmodul PCD2.W1 (12 Bit)



Eingänge: = i0
 | Ganzzahl Werte
 = i3

Zyklische Bearbeitung von 1 bis 4 Eingangskanälen eines PCD2.W1-Moduls. Für jedes in einer PCD2 bestückte Modul dieses Typs ist diese Funktion auszuführen. Bei "Add" ist die Basisadresse des Moduls anzugeben (Elementtyp = Output), z.B. O 16.

Ausziehbar von 1 auf 4 Analog-Eingänge.

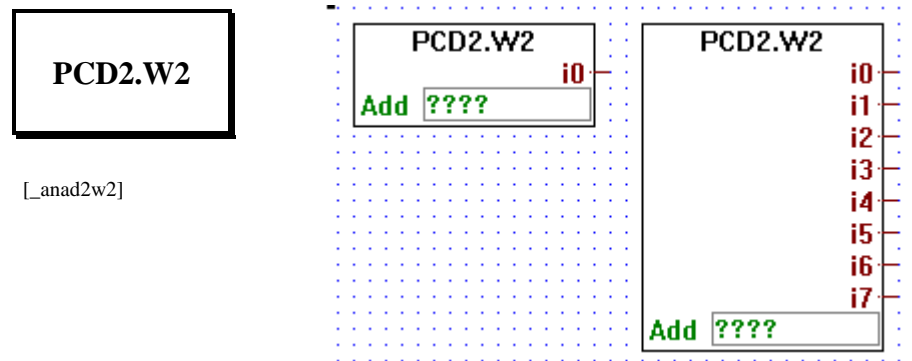
An den "Eingängen" i0 bis i3 liegt der bereits konvertierte Ganzzahlwert des entsprechenden Eingangskanals. i0 bis i3 sind also die Ausgänge des analogen Eingangsmoduls.

Bei jedem Aufruf dieser Funktion wird ein Eingangssignal bearbeitet. Ist die Konvertierung noch nicht abgeschlossen, bleibt der ausgegebene Wert unverändert. Wurde ein Kanal seit der Initialisierung des Systems noch nie bearbeitet, wird der Wert Null ausgegeben.

Der letzte Steckplatz des Erweiterungsgehäuses (O 240) darf nicht mit einem Analogmodul bestückt werden (Konflikt mit dem Watch Dog).

Die Konvertierung erfolgt vom letzt definierten zum Kanal Nr. 0.

4.4.14.2 Analoges Eingangsmodul PCD2.W2 (10 Bit)



Eingänge: = i0
 | Ganzzahl Werte
 = i7

Zyklische Bearbeitung von 1 bis 8 Eingangskanälen eines PCD2.W2-Moduls. Für jedes in einer PCD2 bestückte Modul dieses Typs ist diese Funktion auszuführen. Bei "Add" ist die Basisadresse des Moduls anzugeben (Elementtyp = Output), z.B. O 16.

Ausziehbar von 1 auf 8 Analog-Eingänge

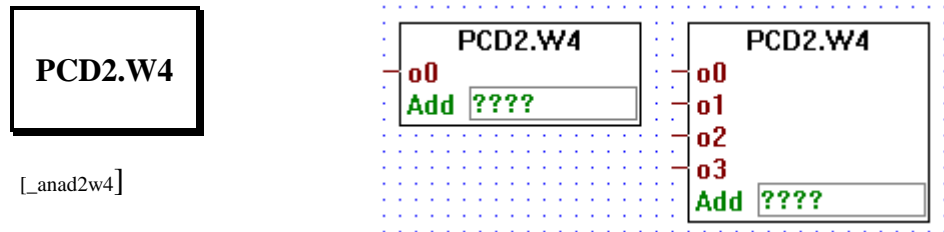
An den "Eingängen" i0 bis i7 liegt der bereits konvertierte Ganzzahlwert des entsprechenden Eingangskanals. i0 bis i7 sind also die Ausgänge des analogen Eingangsmoduls.

Bei jedem Aufruf dieser Funktion wird ein Eingangssignal bearbeitet. Ist die Konvertierung noch nicht abgeschlossen, bleibt der ausgegebene Wert unverändert. Wurde ein Kanal seit der Initialisierung des Systems noch nie bearbeitet, wird der Wert Null ausgegeben.

Der letzte Steckplatz des Erweiterungsgehäuses (O 240) darf nicht mit einem Analogmodul bestückt werden (Konflikt mit dem Watch Dog).

Die Konvertierung erfolgt vom letzt definierten zum Kanal Nr. 0.

4.4.14.3 Analoges Ausgangsmodul PCD2.W4 (8 Bit)



Ausgänge: = o0
 | Ganzzahl Werte
 = o3

Zyklische Bearbeitung von 1 bis 4 Ausgangskanälen eines PCD2.W4-Moduls. Für jedes in einer PCD2 bestückte Modul dieses Typs ist diese Funktion auszuführen. Bei "Add" ist die Basisadresse des Moduls anzugeben (Elementtyp = Output), z.B. O 16.

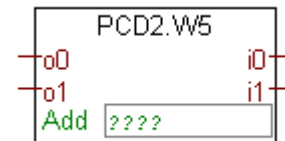
Ausziehbar von 1 auf 4 Analog-Ausgänge

An den "Ausgängen" o0 bis o3 wird der Ganzzahlwert, der in ein Analogsignal zu konvertieren ist, angelegt. o0 bis o3 sind also die Eingänge des analogen Ausgangsmoduls.

Der letzte Steckplatz des Erweiterungsgehäuses (O 240) darf nicht mit einem Analogmodul bestückt werden (Konflikt mit dem Watch Dog).

Die Konvertierung erfolgt vom letzt definierten zum Kanal Nr. 0.

4.4.14.4 Analoges Ein-/Ausgangsmodul PCD2.W5 (12 Bit)



[_anad2w5]

Eingänge: = i0 Ganzzahl Wert
 = i1 Ganzzahl Wert

Ausgänge: = o0 Ganzzahl Wert
 = o1 Ganzzahl Wert

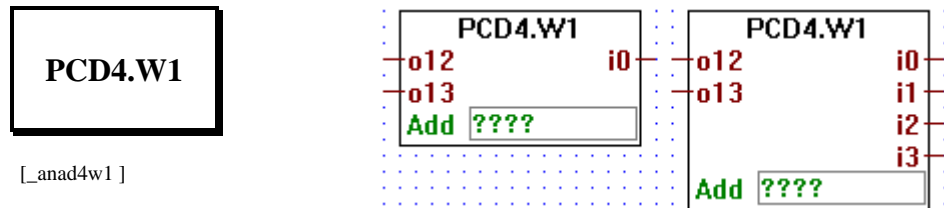
Zyklische Bearbeitung eines PCD2.W5-Moduls. Für jedes in einer PCD2 bestückte Modul dieses Typs ist diese Funktion auszuführen. Bei "Add" ist die Basisadresse des Moduls anzugeben (Elementtyp = Output), z.B. O 16.

An den "Ausgängen" o0 und o1 wird der Ganzzahlwert, der in ein Analogsignal zu konvertieren ist, angelegt. o0 und o1 sind also die Eingänge des analogen Ausgangsmoduls.

An den "Eingängen" i0 und i1 liegt der bereits konvertierte Ganzzahlwert des entsprechenden Eingangskanals. i0 und i1 sind also die Ausgänge des analogen Eingangsmoduls.

Der letzte Steckplatz des Erweiterungsgehäuses (O 240) darf nicht mit einem Analogmodul bestückt werden (Konflikt mit dem Watch Dog).

4.4.14.5 Analoges Ein- und Ausgangsmodul PCD4.W1 (12 Bit)



Eingänge: = i0
 | Ganzzahl Werte
 = i3

Ausgänge = o12
 | Ganzzahl Werte
 = o13

Zyklische Bearbeitung von 2 Ausgangs- und 1 bis 4 Eingangskanälen eines PCD4.W1-Moduls. Für jedes in einer PCD4 bestückte Modul dieses Typs ist diese Funktion auszuführen. Bei "Add" ist die Basisadresse des Moduls anzugeben (Elementtyp = Output), z.B. O 16.

Ausziehbar von 1 auf 4 Analog-Eingänge

An den "Ausgängen" o0 bis o1 wird der Ganzzahlwert, der in ein Analogsignal zu konvertieren ist, angelegt. o0 bis o1 sind also die Eingänge des analogen Ausgangsmoduls.

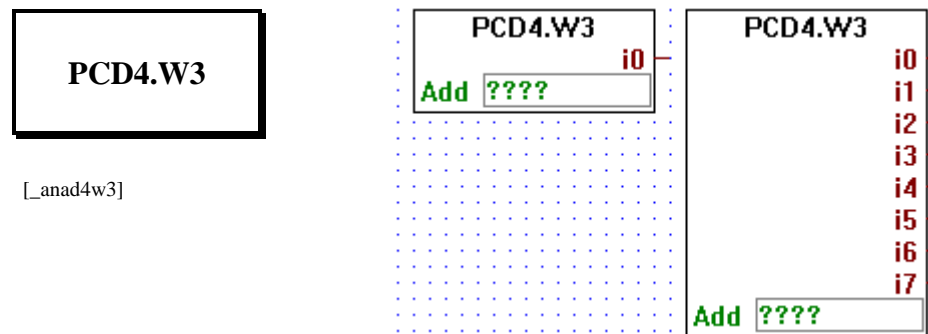
An den "Eingängen" i0 bis i3 liegt der bereits konvertierte Ganzzahlwert des entsprechenden Eingangskanals. i0 bis i3 sind also die Ausgänge des analogen Eingangsmoduls.

Bei jedem Aufruf dieser Funktion wird ein Eingangssignal bearbeitet. Ist die Konvertierung noch nicht abgeschlossen, bleibt der ausgegebene Wert unverändert. Wurde ein Kanal seit der Initialisierung des Systems noch nie bearbeitet, wird der Wert Null ausgegeben.

Die letzten Steckplätze rechts und links der CPU (O 240 und O 496) dürfen nicht mit einem Analogmodul bestückt werden (Konflikt mit dem Watch Dog).

Die Konvertierung erfolgt vom letzt definierten Eingangs-Kanal zum Kanal Nr. 0. Danach erfolgt die Konvertierung der Ausgangskanäle o12 und o13.

4.4.14.6 Analoges Eingangsmodul PCD4.W3 (12 Bit + Vorzeichen)



Eingänge: = i0
 | Ganzzahl Werte
 = i7

Zyklische Bearbeitung von 1 bis 8 Eingangskanälen eines PCD4.W3-Moduls. Für jedes in einer PCD4 bestückte Modul dieses Typs ist diese Funktion auszuführen. Bei "Add" ist die Basisadresse des Moduls anzugeben (Elementtyp = Output), z.B. O 16.

Ausziehbar von 1 auf 8 Analog-Eingänge

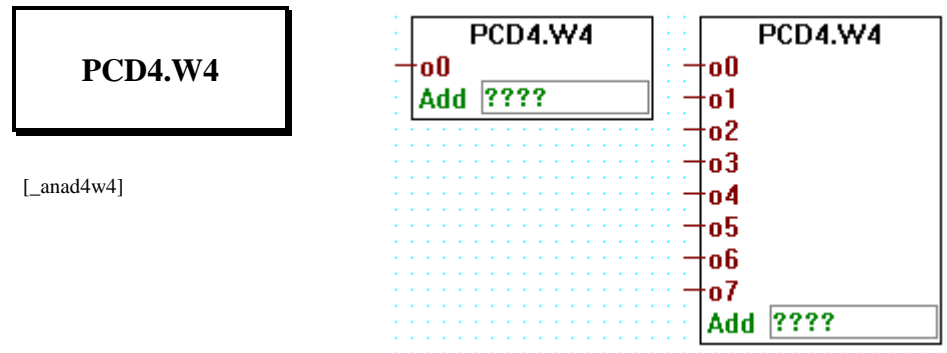
An den "Eingängen" i0 bis i7 liegt der bereits konvertierte Ganzzahlwert des entsprechenden Eingangskanal. i0 bis i7 sind also die Ausgänge des analogen Eingangsmoduls.

Bei jedem Aufruf dieser Funktion wird ein Eingangssignal bearbeitet. Ist die Konvertierung noch nicht abgeschlossen, bleibt der ausgegebene Wert unverändert. Wurde ein Kanal seit der Initialisierung des Systems noch nie bearbeitet, wird der Wert Null ausgegeben.

Die letzten Steckplätze rechts und links der CPU (O 240 und O 496) dürfen nicht mit einem Analogmodul bestückt werden (Konflikt mit dem Watch Dog).

Die Konvertierung erfolgt vom letzt definierten zum Kanal Nr. 0.

4.4.14.7 Analoges Ausgangsmodul PCD4.W4 (8 Bit)



Ausgänge: = o0
 | Ganzzahl Werte
 = o7

Zyklische Bearbeitung von 1 bis 8 Ausgangskanälen eines PCD4.W4-Moduls. Für jedes in einer PCD4 bestückte Modul dieses Typs ist diese Funktion auszuführen. Bei "Add" ist die Basisadresse des Moduls anzugeben (Elementtyp = Output), z.B. O 16.

Ausziehbar von 1 auf 8 Analog-Ausgänge

An den "Ausgängen" o0 bis o7 wird der Ganzzahlwert, der in ein Analogsignal zu konvertieren ist, angelegt. o0 bis o7 sind also die Eingänge des analogen Ausgangsmoduls.

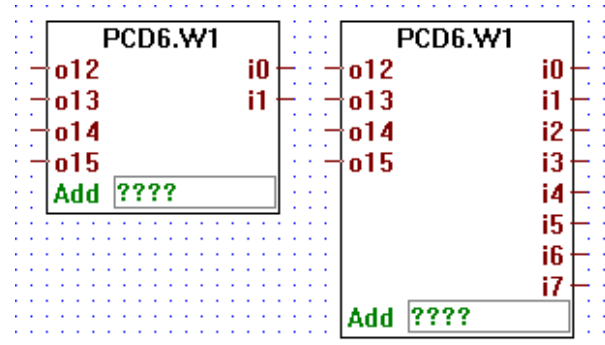
Die letzten Steckplätze rechts und links der CPU (O 240 und O 496) dürfen nicht mit einem Analogmodul bestückt werden (Konflikt mit dem Watch Dog).

Die Konvertierung erfolgt vom letzt definierten zum Kanal Nr. 0.

4.4.14.8 Analoges Ein- und Ausgangsmodul PCD6.W1 (12 Bit)



[_anad6w



Ausgänge: = o12
 | Ganzzahl Werte
 = o15

Eingänge: = i0
 | Ganzzahl Werte
 = i7

Zyklische Bearbeitung von 4 Ausgangs- und 1 bis 8 Eingangskanälen eines PCD6.W1-Moduls. Für jedes in einer PCD6 bestückte Modul dieses Typs ist diese Funktion auszuführen. Bei "Add" ist die Basisadresse des Moduls anzugeben (Elementtyp = Output), z.B. O 160.

Ausziehbar von 1 auf 8 Analog-Eingänge

An den "Ausgängen" o0 bis o3 wird der Ganzzahlwert, der in ein Analogsignal zu konvertieren ist, angelegt. o0 bis o3 sind also die Eingänge des analogen Ausgangsmoduls.

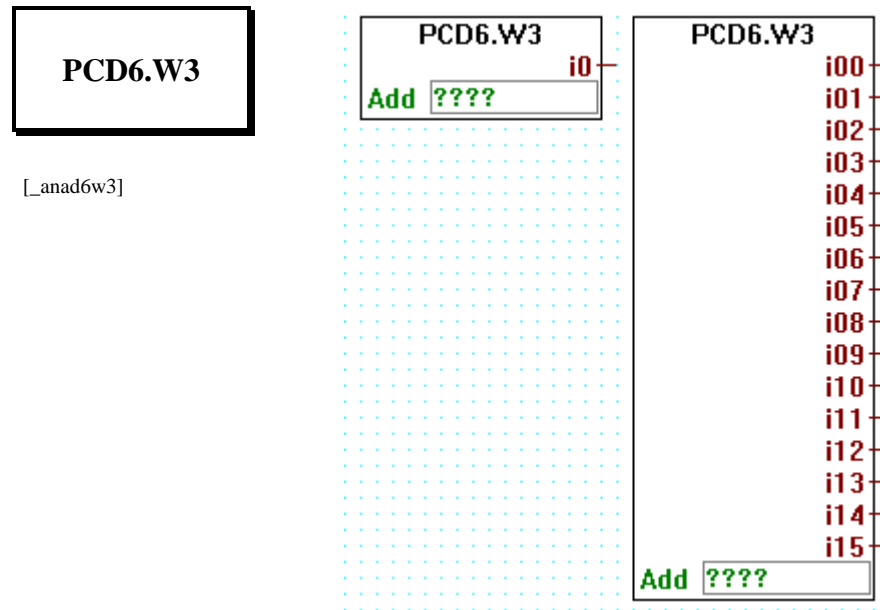
An den "Eingängen" i0 bis i7 liegt der bereits konvertierte Ganzzahlwert des entsprechenden Eingangskanal. i0 bis i7 sind also die Ausgänge des analogen Eingangsmoduls.

Bei jedem Aufruf dieser Funktion wird ein Eingangssignal bearbeitet. Ist die Konvertierung noch nicht abgeschlossen, bleibt der ausgegebene Wert unverändert. Wurde ein Kanal seit der Initialisierung des Systems noch nie bearbeitet, wird der Wert Null ausgegeben.

Die letzten Elementadressen jedes Gehäuses (Basis-Gehäuseadresse + 240) dürfen nicht mit einem Analogmodul bestückt werden (Konflikt mit dem Watch Dog).

Die Konvertierung erfolgt vom letzt definierten Eingangs-Kanal zum Kanal Nr. 0. Danach erfolgt die Konvertierung der Ausgangskanäle o12 bis o15.

4.4.14.9 Analoges Eingangsmodul PCD6.W3 (12 Bit + Vorzeichen)



Eingänge: = i00
 | Ganzzahl Werte
 = i15

Zyklische Bearbeitung von 1 bis 16 Eingangskanälen eines PCD6.W3-Moduls. Für jedes in einer PCD6 bestückte Modul dieses Typs ist diese Funktion auszuführen. Bei "Add" ist die Basisadresse des Moduls anzugeben (Elementtyp = Output), z.B. O 160.

Ausziehbar von 1 auf 16 Analog-Eingänge

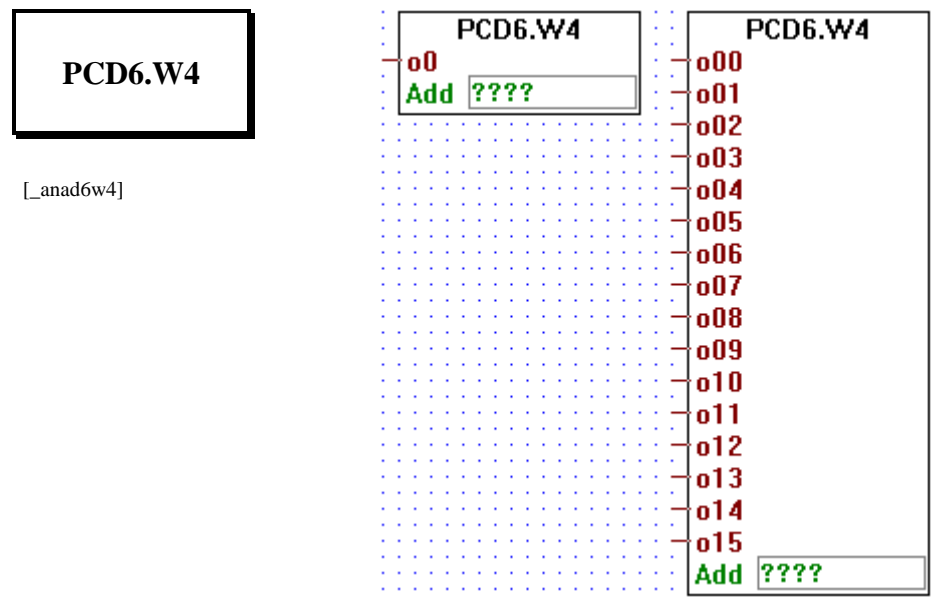
An den "Eingängen" i00 bis i15 liegt der bereits konvertierte Ganzzahlwert des entsprechenden Eingangskanals. i00 bis i15 sind also die Ausgänge des analogen Eingangsmoduls.

Bei jedem Aufruf dieser Funktion wird ein Eingangssignal bearbeitet. Ist die Konvertierung noch nicht abgeschlossen, bleibt der ausgegebene Wert unverändert. Wurde ein Kanal seit der Initialisierung des Systems noch nie bearbeitet, wird der Wert Null ausgegeben.

Die letzten Elementadressen jedes Gehäuses (Basis-Gehäuseadresse + 240) dürfen nicht mit einem Analogmodul bestückt werden (Konflikt mit dem Watch Dog).

Die Konvertierung erfolgt vom letzt definierten zum Kanal Nr. 0.

4.4.14.10 Analoges Ausgangsmodul PCD6.W4 (8 Bit)



Ausgänge: = o00
 | Ganzzahl Werte
 = o15

Zyklische Bearbeitung von 1 bis 16 Ausgangskanälen eines PCD6.W4-Moduls. Für jedes in einer PCD6 bestückte Modul dieses Typs ist diese Funktion auszuführen. Bei "Add" ist die Basisadresse des Moduls anzugeben (Elementtyp = Output), z.B. O 160.

Ausziehbar von 1 auf 16 Analog-Ausgänge

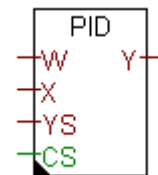
An den "Ausgängen" o00 bis o15 wird der Ganzzahlwert, der in ein Analogsignal zu konvertieren ist, angelegt. o00 bis o15 sind also die Eingänge des analogen Ausgangsmoduls.

Die letzten Elementadressen jedes Gehäuses (Basis-Gehäuseadresse + 240) dürfen nicht mit einem Analogmodul bestückt werden (Konflikt mit dem Watch Dog).

Die Konvertierung erfolgt vom letzt definierten zum Kanal Nr. 0.

4.4.15 Regelung**[sfupregu]****4.4.15.1 PID-Regelung**

[_regpid3]



| | | | |
|-----------|-----|-----------------------|----------------|
| Eingänge: | W : | Sollwert | Ganzzahlformat |
| | X : | Regelgrösse | Ganzzahlformat |
| | YS: | Kaltstart-Stellgrösse | Ganzzahlformat |
| | CS: | Kaltstart-Signal | Binärformat |

| | | | |
|----------|-----|-------------|----------------|
| Ausgang: | Y : | Stellgrösse | Ganzzahlformat |
|----------|-----|-------------|----------------|

Die Beschreibung basiert auf der Annahme, dass der Anwender gute Kenntnisse in der Regeltechnik besitzt. Der PID-Befehl der PCD ist im - Handbuch "Befehlssatz für die PCD-Familie" erläutert.

Kaltstart und Handbetrieb:

YS ist die Stellgrösse, welche vom Systemprogramm als Startwert für Y verwendet wird. Wenn das CS-Signal dynamisiert ist, wird nach einem Kaltstart weiter geregelt. Wenn CS über längere Zeit H bleibt, wird bei jeder Abtastung YS zu Y übergeben. Die Regelung ist ausser Betrieb und kann im 'Handbetrieb' weitergeführt werden.

Im Einstellfenster können die 6 folgenden Parameter definiert werden:

Abtastzeit To:

Definiert den Zeittakt in welchem der PID-Befehl zyklisch ausgeführt wird. Bereich: 0.1 - 3'600.0 sek.

Proportional Faktor Fp:

Dieser Faktor bestimmt die Proportional-Charakteristik (Verstärkung) der Regelung. Bereich: 1 - 65'535.

Der Proportionalfaktor Fp ist folgendermassen definiert:

$$F_p = (1/X_p) * 256$$

wobei Xp: Proportionalband

Integalfaktor F_i :

Dieser Faktor bestimmt die Integral-Charakteristik der Regelung.
Bereich: 0 - 65'535.

Der Integalfaktor F_i ist folgendermassen definiert:

$$F_i = (T_o/T_i) * 256$$

wobei T_o : Abtastzeit
 T_i : Integrationszeit

Differentialfaktor F_D :

Dieser Faktor bestimmt die Differential-Charakteristik der Regelung.
Bereich: 0 - 65'535.

Der Differentialfaktor F_d ist folgendermassen definiert:

$$F_d = (T_d/T_o) * 256$$

wobei T_o : Abtastzeit
 T_d : Differentialzeit

Totzone D_r :

Die Totzone gibt den Bereich an, in welchem Stellgrössenschwankungen auftreten können, ohne dass die Stellgrösse ändert. Bereich: 0 - 65'535.

Auflösung m :

Die Auflösung bestimmt die Maximalwerte der Regel- und Stellgrössen.

- Ist $m = 8$, wird mit 8 Bit gerechnet (0 - 255)
- Ist $m = 12$, wird mit 12 Bit gerechnet (0 - 4095)
- Ist $m = 16$, wird mit 16 Bit gerechnet (0 - 65'535)

Die Auflösung ist meistens durch das verwendete Analogmodul bestimmt.

Einstellfenster:



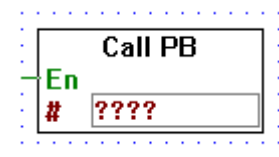
4.4.16 Benutzerdefinierte Funktionen

[sfupuser]

Ein komfortables Werkzeug zum Editieren eigener Funktionen kann für die vorliegende Version noch nicht angeboten werden.

Es wird empfohlen, wiederkehrende Programmteile als PB (Programm Block) oder FB (Funktions Block) in FUPLA oder IL zu editieren und dann bedingt oder unbedingt aus dem FUPLA aufzurufen.

4.4.16.1 Aufruf PB - Aufruf eines Programm Blocks



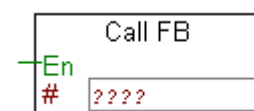
[_call_pb]

Eingänge: – En: Enable Binärformat
 # PB Adresse Ganzzahlformat oder Symbol

Wird am Eingang "En" ein H-Signal angelegt, wird der PB aufgerufen, wird ein L-Signal angelegt, wird der PB nicht aufgerufen.

Im Eintragfeld wird die PB-Nummer (0 - 299) oder ein symbolischer Name angegeben.

4.4.16.2 Aufruf FB - Aufruf eines Funktions Blocks



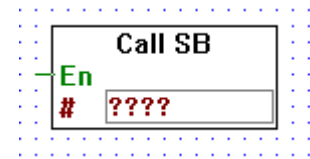
[_call_fb]

Eingänge: – En: Enable Binärformat
 # FB Adresse Ganzzahlformat oder Symbol

Wird am Eingang "En" ein H-Signal angelegt, wird der FB aufgerufen, wird ein L-Signal angelegt, wird der FB nicht aufgerufen. Der FB-Aufruf kann in der vorliegenden Version nicht mit Parametern versehen werden.

Im Eintragfeld wird die FB-Nummer (0 - 999) oder ein symbolischer Name angegeben.

4.4.16.3 Aufruf SB - Aufruf eines Sequential Blocks



[_call_sb]

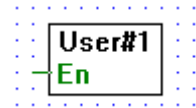
Eingänge: – En: Enable Binärformat
 # SB Adresse Ganzzahlformat

Wird am Eingang "En" ein H-Signal angelegt, wird der SB aufgerufen, wird ein L-Signal angelegt, wird der SB nicht aufgerufen.

Im Eintragsfeld wird die SB-Nummer (0 - 31) oder ein symbolischer Name angegeben.

In einfacheren Anwendungen erfolgt der Aufruf eines SB mit der Funktion 'Make' automatisch. (Siehe Handbuch PG4, Kapitel 5: GRAFTEC)

4.4.16.4 Durch Anwender definierten Block Nr. 1



[_user_1]

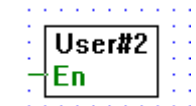
Eingang: – En: Enable Binärformat

Es wird in IL (Instruction List) im gleichen Verzeichnis oder Unterverzeichnis wie der Funktionsplan eine Datei mit dem Namen "USER_1.H" editiert.

Im Funktionsplan wird die Funktion "User#1" in gewohnter Weise platziert und der binäre Eingang "En" (Enable) verbunden. Mit einem H-Signal wird die Funktion ausgeführt, mit einem L-Signal nicht.

Beim Assemblieren wird der separat editierte Programmteil automatisch eingefügt (Included).

4.4.16.5 Durch Anwender definierten Block Nr. 2



[_user_2]

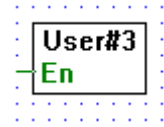
Eingang: – En: Enable Binärformat

Es wird in IL (Instruction List) im gleichen Verzeichnis oder Unterverzeichnis wie der Funktionsplan eine Datei mit dem Namen "USER_2.H" editiert.

Im Funktionsplan wird die Funktion "User#2" in gewohnter Weise platziert und der binäre Eingang "En" (Enable) verbunden. Mit einem H-Signal wird die Funktion ausgeführt, mit einem L-Signal nicht.

Beim Assemblieren wird der separat editierte Programmteil automatisch eingefügt (Included).

4.4.16.6 Durch Anwender definierten Block Nr. 3



[_user_3]

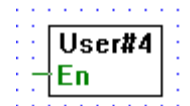
Eingang: – En: Enable Binärformat

Es wird in IL (Instruction List) im gleichen Verzeichnis oder Unterverzeichnis wie der Funktionsplan eine Datei mit dem Namen "USER_3.H" editiert.

Im Funktionsplan wird die Funktion "User#3" in gewohnter Weise platziert und der binäre Eingang "En" (Enable) verbunden. Mit einem H-Signal wird die Funktion ausgeführt, mit einem L-Signal nicht.

Beim Assemblieren wird der separat editierte Programmteil automatisch eingefügt (Included).

4.4.16.7 Durch Anwender definierten Block Nr. 4



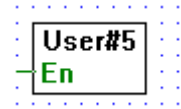
[_user_4]

Eingang: – En: Enable Binärformat

Es wird in IL (Instruction List) im gleichen Verzeichnis oder Unterverzeichnis wie der Funktionsplan eine Datei mit dem Namen "USER_4.H" editiert.

Im Funktionsplan wird die Funktion "User#4" in gewohnter Weise platziert und der binäre Eingang "En" (Enable) verbunden. Mit einem H-Signal wird die Funktion ausgeführt, mit einem L-Signal nicht.

Beim Assemblieren wird der separat editierte Programmteil automatisch eingefügt (Included).

4.4.16.8 Durch Anwender definierten Block Nr. 5

[_user_5]

Eingang: – En: Enable Binärformat

Es wird in IL (Instruction List) im gleichen Verzeichnis oder Unterverzeichnis wie der Funktionsplan eine Datei mit dem Namen "USER_5.H" editiert.

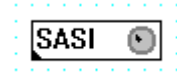
Im Funktionsplan wird die Funktion "User#5" in gewohnter Weise platziert und der binäre Eingang "En" (Enable) verbunden. Mit einem H-Signal wird die Funktion ausgeführt, mit einem L-Signal nicht.

Beim Assemblieren wird der separat editierte Programmteil automatisch eingefügt (Included).

Notizen:

4.4.17 Serielle Kommunikation (Datenaustausch) [sfupcomm]

4.4.17.1 Schnittstellen-Parameter - Assignierung einer seriellen Schnittstelle



[_comsasi]

Initialisierung der seriellen Kommunikation: SASI.

Beim Aufstarten der PCD wird eine serielle Schnittstelle mit dem definierten Kommunikations-Modus assigniert. Jede in einer PCD zur Verwendung kommende Schnittstelle muss mit dieser Funktion assigniert werden. (Für jeden Kanal eine eigene SASI-Funktion). Nach dem Aus- und dem Wiedereinschalten oder nach einem Kaltstart (Restart Cold) wird die Assignierung neu aktiviert, sobald die PCD in Run geschaltet wird, d.h. sobald die Kaltstart-Routine XOB 16 abgearbeitet wird.

Parametrierung:

Textnummer: Jede SASI-Funktion verwendet einen anderen Text (Nr. 0...3999). Es ist darauf zu achten, dass der Text nicht bereits an einer anderen Stelle des Programms verwendet wird, auch nicht in einem eventuell dazu gelinkten Programmteil.

Kanal: Nummer der seriellen Schnittstelle 0...3. Bei der PCD2, der PCD4 und der PCD6.M540 ist der Kanal 0 die Programmierschnittstelle (PGU). Wird diese mit SASI assigniert, funktioniert diese Schnittstelle nicht mehr für die Programmierung.

Modus:

Der Modus "Keiner" erlaubt es, die Assignierung auszuschalten, ohne dass alle Parameter der Funktion verloren gehen.

Mit dem Modus "Text" wird eine Assignierung im Modus C ausgeführt. Die Texte und das Anwenderprogramm welches diese Texte verwendet, muss mit den Programmierwerkzeugen des PG3 erstellt werden. Der Modus Master bzw. Slave hat keine Bedeutung. Die Assignierung wird immer im MC0-Modus ausgeführt.

Der Modus D, SBUS 0 und SBUS 1, kann sehr vorteilhaft mit FUPLA, dank der Funktionen für das Senden und das Empfangen von Einzel-Bit und Ganzzahl-Werten der Familie "Kommunikation", angewendet werden.

Mit dem Modus D (Punkt zu Punkt) ist immer eine Station als Master (Client) und die andere als Slave (Server) zu assignieren. Alle Funktionen für Senden und Empfangen sind an der Master-Station zu programmieren.

Beim Modus S-Bus (Mehrpunktverbindung) kann nur 1 Station als Master assigniert werden. Alle andern sind Slave-Stationen. Ist der Master ein Leitsystem, spielt dieses die Rolle des Masters. Alle andern Stationen sind auch hier Slave-Stationen. Die Funktion für das Senden und Empfangen können von den Slave-Stationen nicht verwendet werden.

Mehr Informationen zu den Kommunikationsmodi können den PCD-Handbüchern entnommen werden.

Uebertragungsgeschwindigkeit:

Diese muss für alle Stationen am Bus gleich sein. Die Uebertragungsgeschwindigkeit ist nach der Qualität des Bus und den möglichen Leistungen der angeschlossenen Apparate zu wählen. (Treiber zu PCD, Repeater, Modem, Kabel usw.)

Anzahl Bit, Parität und Stopbit:

Diese Einstellungen müssen bei allen angeschlossenen Stationen gleich sein. Beim Mode SBUS kommen diese Parameter nicht zur Anwendung.

Diagnoseflag und Diagnoseregister: Siehe Handbücher der PCD-Reihe.

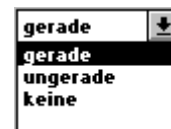
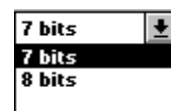
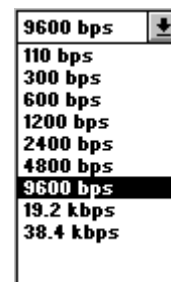
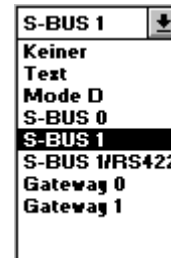
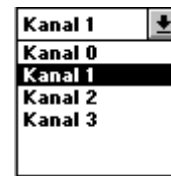
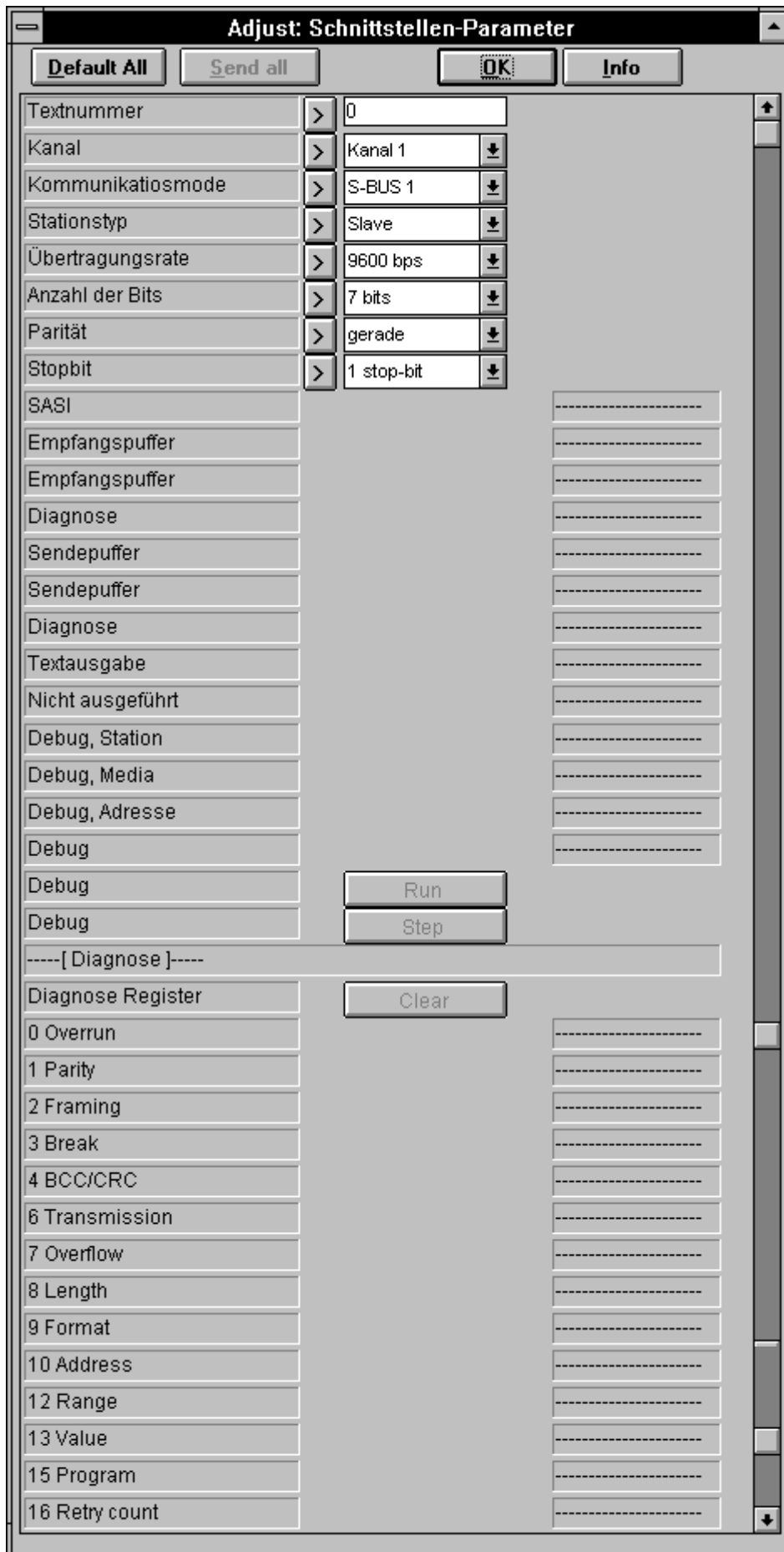
Der ONLINE-Parameter SASI ("OK" oder "Error") zeigt an, ob die Funktion SASI beim Einschalten der PCD korrekt ausgeführt wurde. Im Error-Fall ist auf diesem Kanal keine Kommunikation möglich.

Die Fehlerquellen können die folgenden sein:

- die Firmware unterstützt den gewählten Modus nicht (vorallem SBUS)
- die CPU verfügt nicht über diesen Kanal
- die CPU kann die (hohe) Uebertragungsrate nicht verarbeiten (hauptsächlich 38'400 Bd)
- der Kanal wurde mehrmals assigniert (z.B. von einem dazu gelinkten Programm)
- Syntaxfehler im Assignierungstext. Dieser Fehler dürfte mit dem vorliegenden Programmpaket nicht auftreten, es sei denn, der SASI-Text wurde später in der PCD modifiziert (z.B. im Debugger).

Debug:

Die Debugfunktion erlaubt es, in der Master-Station das richtige Funktionieren der Kommunikation zu überwachen. Ist die Kommunikation im Betrieb, werden die Stationen und die aufeinanderfolgenden adressierten Medien samt den Diagnosedaten zyklisch angezeigt. Um eine bessere Kontrolle der Diagnosedaten zu erhalten und ein Verbindungsproblem zu analysieren, kann die Kommunikation durch einen ersten Klick auf die Taste "Step" unterbrochen werden. Jeder weitere Klick löst den nächsten Kommunikationsbefehl aus. (Senden oder Empfangen). Mit der Taste "Run" kann die Kommunikation normal fortgesetzt werden. Die Informationen "Station, Medium und Adresse" gestatten die Identifikation der aktuellen Sende- oder Empfangsfunktion.



Fortsetzung des Einstellfensters auf der nächsten Seite.

| | | |
|----|--------------------|-------|
| 16 | Retry count | ----- |
| 18 | Transmission | ----- |
| 20 | Response | ----- |
| 21 | Response | ----- |
| 22 | Multiple NAK | ----- |
| 23 | Tx Buffer/TS Delay | ----- |
| 24 | Enquiry | ----- |
| 25 | Format | ----- |
| 28 | Range | ----- |
| 30 | Receive | ----- |
| 31 | Program | ----- |

4.4.17.2 Schnittstellen-Parameter für externe Dateien

Schnittstellen-P. extern

SASI-Extrn

[_comsase]

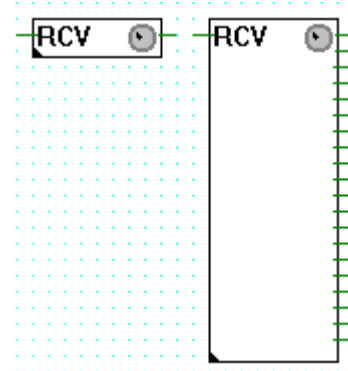
Wird in zwei oder in mehreren FUPLA-Dateien eines Projekts auf die gleiche serielle Schnittstelle zugegriffen, muss in einer (Haupt-) Datei die "normale" Assignierung mit der FBox "Schnittstellen-Parameter" durchgeführt werden. In den andern Dateien ist die vorliegende FBox "Schnittstellen-Parameter extern" einzufügen, wobei die gleichen Parameter wie in der Haupt-Datei einzusetzen sind.



4.4.17.3 Empfangen von 1 - 20 I/O/F



[_comrxb]



Eingang: – Enable

Ausgänge: – → Binärformat

Ist der Enable-Eingang = H (binärer Anschluss links an der Funktionsbox), werden Daten (1-Bit-Elemente = Eingänge, Ausgänge oder Flags) einer Partnerstation an die 1-Bit-Elemente der eigenen Station übertragen. Die Elemente der eigenen Station werden durch Anschluss an die rechte Seite der Funktionsbox definiert.

Die Funktion ist ausziehbar von 1 bis 20 Sende-Elemente.

Weitere Einstellungen, wie Kanalnummer, Typ der Ziel-Elemente usw. werden im zugehörigen Einstellfenster gemacht.



Empfang von 1 - 20 I/O/F (Detail-Beschreibung)

Diese Funktion darf nur auf einer Master-PCD, die mit der Funktion SASI im Modus D oder SBUS assigniert wurde, ausgeführt werden.

Die Funktion gestattet das Empfangen von 1 bis 20 1-Bit-Elementen (I/O/F) von einer Slave-Station. Die Uebertragung wird zyklisch mit der maximal möglichen Geschwindigkeit ausgeführt, solange das Aktivierungssignal = H ist. (Eingang "Send"). Wird eine positive Flanke an den Eingang gelegt, erfolgt mindestens 1 Uebertragung, wenn der Impuls kürzer als 1 Uebertragungszyklus der Kommunikation ist.

Mögliche Fehlerquellen:

Fehlende Assignierung: Der gewählte Kanal wurde nicht oder fehlerhaft assigniert. (Kein oder fehlerhafter SASI). Eine andere Assignierung als durch die SASI-Funktion ist nicht zulässig.

Nicht Master: Der Kanal ist nicht als Master assigniert.

SRXM: Es wurde ein Fehler bei der Ausführung der Instruktion SRXM festgestellt. Dies dürfte bei Verwendung der vorliegenden Funktion nicht auftreten, es sei denn, ein anderer Programmteil habe auch Zugriff auf diesen Kanal.

Diagnose: Es wurde ein Kommunikationsfehler festgestellt. Dieser kann dank der Diagnosemöglichkeiten der SASI-Funktion im Detail analysiert werden. Die Analyse kann auch im Modus "Debug" erfolgen.

Step: Der Kanal wurde mittels des Modus "Debug" der Funktion SASI in "Step" gebracht. Es kann keine Kommunikation mehr stattfinden bis der Kanal wieder in "Run" gebracht wurde.

Parametrierung:

Initialisierung: Diese Option erlaubt eine Uebertragung während dem Start der PCD, auch wenn das binäre Aktivierungssignal = L ist. Dies erlaubt ein Initialisieren der Elemente der Slave-Station nach einem Halt der Master-Station.

Kanal: Zu verwendende Kanalnummer.

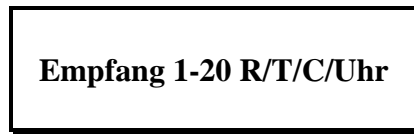
Station: Nummer der Slavestation, wohin die Daten übertagen werden oder wurden. Keine Bedeutung im Mode D.

Typ und Adresse: Basis-Elementadresse der Slave-Station wo die übertragenen Elemente abgelegt werden sollen. Beim Uebertragen mehrerer Elemente werden diese automatisch auf die nachfolgenden Adressen abgelegt.

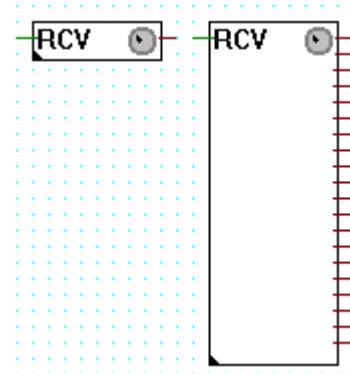
Für die Uebertragung der Hardwareuhr werden immer 2 Register verwendet. Die Uebertragung muss entsprechend gestaltet werden.

Die Taste "Befehl" erlaubt das Ausführen einer Uebertragung, auch wenn das Aktivierungssignal = L ist.

4.4.17.4 Empfangen von 1 - 20 R/T/C/Uhr



[_comrxix]



Eingang: – Enable

Ausgänge: = → Ganzzahlwerte

Ist der Enable-Eingang = H (binärer Anschluss links an der Funktionsbox), werden Daten (32-Bit-Elemente = Register, Timer, Counter oder die Hardware-Uhr) einer Partnerstation an die 32-Bit-Elemente der eigenen Station übertragen. Die Elemente der eigenen Station werden durch Anschluss an die rechte Seite der Funktionsbox definiert.

Die Funktion ist ausziehbar von 1 bis 20 Empfangselemente.

Weitere Einstellungen, wie Kanalnummer, Typ der Quell-Elemente usw. werden im zugehörigen Einstellfenster gemacht.



Empfang von 1 - 20 R/T/C/Uhr (Detail-Beschreibung)

Diese Funktion darf nur auf einer Master-PCD, die mit der Funktion SASI im Modus D oder SBUS assigniert wurde, ausgeführt werden.

Die Funktion gestattet das Empfangen von 1 bis 20 Werten (R/T/C/Uhr) von einer Slave-Station. Die Uebertragung wird zyklisch mit der maximal möglichen Geschwindigkeit ausgeführt, solange das Aktivierungssignal = H ist. (Eingang "Send"). Wird eine positive Flanke an den Eingang gelegt, erfolgt mindestens 1 Uebertragung, wenn der Impuls kürzer als 1 Uebertragungszyklus der Kommunikation ist.

Mögliche Fehlerquellen:

Fehlende Assignierung: Der gewählte Kanal wurde nicht oder fehlerhaft assigniert. (Kein oder fehlerhafter SASI). Eine andere Assignierung als durch die SASI-Funktion ist nicht zulässig.

Nicht Master: Der Kanal ist nicht als Master assigniert.

SRXM: Es wurde ein Fehler bei der Ausführung der Instruktion SRXM festgestellt. Dies dürfte bei Verwendung der vorliegenden Funktion nicht auftreten, es sei denn, ein anderer Programmteil habe auch Zugriff auf diesen Kanal.

Diagnose: Es wurde ein Kommunikationsfehler festgestellt. Dieser kann dank der Diagnosemöglichkeiten der SASI-Funktion im Detail analysiert werden. Die Analyse kann auch im Modus "Debug" erfolgen.

Step: Der Kanal wurde mittels des Modus "Debug" der Funktion SASI in "Step" gebracht. Es kann keine Kommunikation mehr stattfinden bis der Kanal wieder in "Run" gebracht wurde.

Parametrierung:

Initialisierung: Diese Option erlaubt eine Uebertragung während dem Start der PCD, auch wenn das binäre Aktivierungssignal = L ist. Dies erlaubt ein Initialisieren der Elemente der Slave-Station nach einem Halt der Master-Station.

Kanal: Zu verwendende Kanalnummer.

Station: Nummer der Slavestation, wohin die Daten übertagen werden oder wurden. Keine Bedeutung im Mode D.

Typ und Adresse: Basis-Elementadresse der Slave-Station wo die übertragenen Elemente abgelegt werden sollen. Beim Uebertragen mehrerer Elemente werden diese automatisch auf die nachfolgenden Adressen abgelegt.

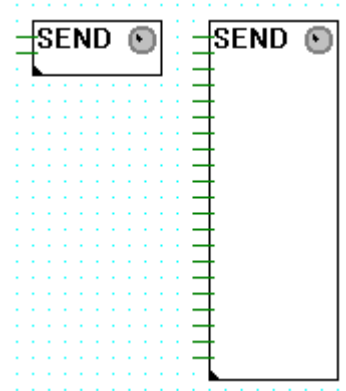
Für die Uebertragung der Hardwareuhr werden immer 2 Register verwendet. Die Uebertragung muss entsprechend gestaltet werden.

Die Taste "Befehl" erlaubt das Ausführen einer Uebertragung, auch wenn das Aktivierungssignal = L ist.

4.4.17.5 Senden von 1-20 I/O/F



[_comtxb]



- Eingänge: – Enable
- → Binärformat

Ist der Enable-Eingang = H (binärer Anschluss links zuoberst an der Funktionsbox), werden Daten (1-Bit-Elemente = Eingänge, Ausgänge oder Flags) von der eigenen zu einer Partner-Station übertragen. Die Elemente der eigenen Station werden durch Anschluss an die linke Seite der Funktionsbox definiert (2. bis 21. Anschluss).

Die Funktion ist ausziehbar von 1 bis 20 Sende-Elemente.

Weitere Einstellungen, wie Kanalnummer, Typ der Ziel-Elemente usw. werden im zugehörigen Einstellfenster gemacht.



Senden von 1 - 20 I/O/F (Detail-Beschreibung)

Diese Funktion darf nur auf einer Master-PCD, die mit der Funktion SASI im Modus D oder SBUS assigniert wurde, ausgeführt werden.

Die Funktion gestattet das Uebertragen von 1 bis 20 1-Bit-Elementen (I/O/F) zu einer Slave-Station. Die Uebertragung wird zyklisch mit der maximal möglichen Geschwindigkeit ausgeführt, solange das Aktivierungssignal = H ist. (Eingang "Send"). Wird eine positive Flanke an den Eingang gelegt, erfolgt mindestens 1 Uebertragung, wenn der Impuls kürzer als 1 Uebertragungszyklus der Kommunikation ist.

Mögliche Fehlerquellen:

Fehlende Assignierung: Der gewählte Kanal wurde nicht oder fehlerhaft assigniert. (Kein oder fehlerhafter SASI). Eine andere Assignierung als durch die SASI-Funktion ist nicht zulässig.

Nicht Master: Der Kanal ist nicht als Master assigniert.

STXM: Es wurde ein Fehler bei der Ausführung der Instruktion STXM festgestellt. Dies dürfte bei Verwendung der vorliegenden Funktion nicht auftreten, es sei denn, ein anderer Programmteil habe auch Zugriff auf diesen Kanal.

Diagnose: Es wurde ein Kommunikationsfehler festgestellt. Dieser kann dank der Diagnosemöglichkeiten der SASI-Funktion im Detail analysiert werden. Die Analyse kann auch im Modus "Debug" erfolgen.

Step: Der Kanal wurde mittels des Modus "Debug" der Funktion SASI in "Step" gebracht. Es kann keine Kommunikation mehr stattfinden bis der Kanal wieder in "Run" gebracht wurde.

Parametrierung:

Initialisierung: Diese Option erlaubt eine Uebertragung während dem Start der PCD, auch wenn das binäre Aktivierungssignal = L ist. Dies erlaubt ein Initialisieren der Elemente der Slave-Station nach einem Halt der Master-Station.

Kanal: Zu verwendende Kanalnummer.

Station: Nummer der Slavestation, wohin die Daten übertagen werden oder wurden. Keine Bedeutung im Mode D.

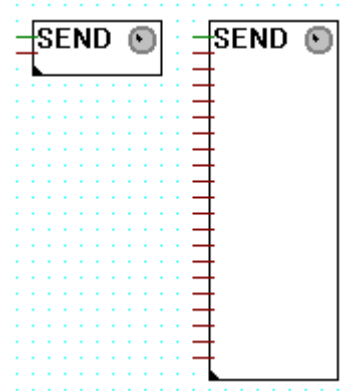
Typ und Adresse: Basis-Elementadresse der Slave-Station wo die übertragenen Elemente abgelegt werden sollen. Beim Uebertragen mehrerer Elemente werden diese automatisch auf die nachfolgenden Adressen abgelegt.

Für die Uebertragung der Hardwareuhr werden immer 2 Register verwendet. Die Uebertragung muss entsprechend gestaltet werden. Die Taste "Befehl" erlaubt das Ausführen einer Uebertragung, auch wenn das Aktivierungssignal = L ist.

4.4.17.6 Senden von 1-20 R/T/C/Uhr



[_comtxi]



Eingänge: – Enable
 = Ganzzahlformat

Ist der Enable-Eingang = H (binärer Anschluss links zuoberst an der Funktionsbox), werden Daten (32-Bit-Elemente = Register, Timer, Counter oder die Hardware-Uhr) von der eigenen zu einer Partner-Station übertragen. Die Elemente der eigenen Station werden durch Anschluss an die linke Seite der Funktionsbox definiert (2. bis 21. Anschluss).

Die Funktion ist ausziehbar von 1 bis 20 Sende-Elemente.

Weitere Einstellungen, wie Kanalnummer, Typ der Ziel-Elemente usw. werden im zugehörigen Einstellfenster gemacht.



Senden von 1 - 20 R/T/C/Uhr

Diese Funktion darf nur auf einer Master-PCD, die mit der Funktion SASI im Modus D oder SBUS assigniert wurde, ausgeführt werden.

Die Funktion gestattet das Uebertragen von 1 bis 20 Werten (R/T/C/Uhr) zu einer Slave-Station. Die Uebertragung wird zyklisch mit der maximal möglichen Geschwindigkeit ausgeführt, solange das Aktivierungssignal = H ist. (Eingang "Send"). Wird eine positive Flanke an den Eingang gelegt, erfolgt mindestens 1 Uebertragung, wenn der Impuls kürzer als 1 Uebertragungszyklus der Kommunikation ist.

Mögliche Fehlerquellen:

Fehlende Assignierung: Der gewählte Kanal wurde nicht oder fehlerhaft assigniert. (Kein oder fehlerhafter SASI). Eine andere Assignierung als durch die SASI-Funktion ist nicht zulässig.

Nicht Master: Der Kanal ist nicht als Master assigniert.

STXM: Es wurde ein Fehler bei der Ausführung der Instruktion STXM festgestellt. Dies dürfte bei Verwendung der vorliegenden Funktion nicht auftreten, es sei denn, ein anderer Programmteil habe auch Zugriff auf diesen Kanal.

Diagnose: Es wurde ein Kommunikationsfehler festgestellt. Dieser kann dank der Diagnosemöglichkeiten der SASI-Funktion im Detail analysiert werden. Die Analyse kann auch im Modus "Debug" erfolgen.

Step: Der Kanal wurde mittels des Modus "Debug" der Funktion SASI in "Step" gebracht. Es kann keine Kommunikation mehr stattfinden bis der Kanal wieder in "Run" gebracht wurde.

Parametrierung:

Initialisierung: Diese Option erlaubt eine Uebertragung während dem Start der PCD, auch wenn das binäre Aktivierungssignal = L ist. Dies erlaubt ein Initialisieren der Elemente der Slave-Station nach einem Halt der Master-Station.

Kanal: Zu verwendende Kanalnummer.

Station: Nummer der Slavestation, wohin die Daten übertagen werden oder wurden. Keine Bedeutung im Mode D.

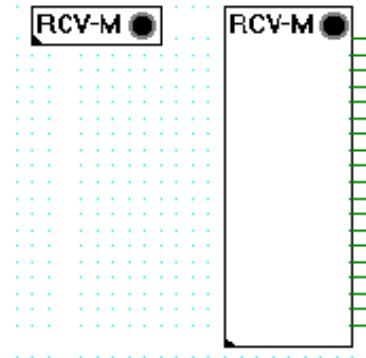
Typ und Adresse: Basis-Elementadresse der Slave-Station wo die übertragenen Elemente abgelegt werden sollen. Beim Uebertragen mehrerer Elemente werden diese automatisch auf die nachfolgenden Adressen abgelegt.

Für die Uebertragung der Hardwareuhr werden immer 2 Register verwendet. Die Uebertragung muss entsprechend gestaltet werden. Die Taste "Befehl" erlaubt das Ausführen einer Uebertragung, auch wenn das Aktivierungssignal = L ist.

4.4.17.7 Empfang I/O/F von mehreren Stationen



[_comrxbm]



Eingang: – Enable

Ausgänge: – → Binärformat

Ist der Enable-Eingang = H (binärer Anschluss links an der Funktionsbox), wird je ein im Einstellfenster definiertes binäres Element (Eingang, Ausgang oder Flag) von jeder Partner-Station in die eigene Station übertragen.

Die Funktion kann von 1 bis 20 Ausgangselemente erweitert werden (1 Element pro Station). Die Größe der FBox (ausziehbar von 1 auf 20 Ausgangselemente) definiert auch die Anzahl aufeinanderfolgender Stationen, ab welchen gelesen werden soll. Die Station mit der tiefsten Adresse wird im Einstellfenster angegeben.

Zusätzliche Angaben, wie Kanal-Nummer, Typ des Quellelements usw. werden auch im Einstellfenster angegeben.



Diese Funktion kann nur auf einer Master-PCD, welche mit der Schnittstellen-Parameter FBox als S-BUS-Master assigniert wurde, ausgeführt werden.

Die Funktion erlaubt den Empfang je eines binären Elementes von 1 - 20 Slave-Stationen. Die Übertragung erfolgt zyklisch mit der maximal möglichen Geschwindigkeit und zwar solange, wie das Enable-Signal = H ist. Erhält der Enable-Eingang eine positive Flanke, wird mindestens jede

Station einmal abgefragt, auch wenn das Enable-Signal kürzer als ein Kommunikationszyklus sein sollte.

Mögliche Fehlerquellen:

Fehlende Assignierung: Der gewählte Kanal wurde nicht oder fehlerhaft assigniert. (Kein oder fehlerhafter SASI). Eine andere Assignierung als durch die SASI-Funktion ist nicht zulässig.

Nicht Master: Der Kanal ist nicht als Master assigniert.

Nicht S-BUS: Der Kanal ist nicht im S-BUS-Modus assigniert. Nur das S-BUS-Protokoll unterstützt diese Funktion.

SRXM: Es wurde ein Fehler bei der Ausführung der Instruktion SRXM festgestellt. Dies dürfte bei Verwendung der vorliegenden Funktion nicht auftreten, es sei denn, ein anderer Programmteil habe auch Zugriff auf diesen Kanal.

Diagnose: Es wurde ein Kommunikationsfehler festgestellt. Dieser kann dank der Diagnosemöglichkeiten der SASI-Funktion im Detail analysiert werden.

Zu gross: Die FBox wurde ausgezogen, wobei die letzte Station (definierte 1. Station + Auszugsindex) höher als 254 wird. Nur die Stationen im Adressbereich 1 bis 254 sind erlaubt.

Step: Der Kanal wurde mittels des Modus "Debug" der Funktion SASI in "Step" gebracht. Es kann keine Kommunikation mehr stattfinden bis der Kanal wieder in "Run" gebracht wurde.

Parametrierung:

Initialisierung: Diese Option erlaubt eine Uebertragung während dem Start der PCD, auch wenn das binäre Aktivierungssignal = L ist. Dies erlaubt ein Initialisieren der Elemente der Slave-Station nach einem Halt der Master-Station.

Kanal: Zu verwendende Kanalnummer.

Station: Nummer der Slavestation, wohin die Daten übertragen werden oder wurden.

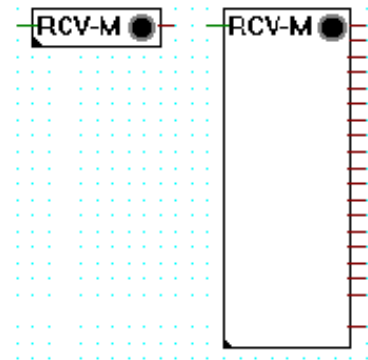
Typ und Adresse: Adresse und Typ des Elementes der Slave-Stationen welche übertragen werden sollen. Pro Station kann nur 1 Element übertragen werden.

Die Taste "Befehl" erlaubt das Ausführen einer Uebertragung, auch wenn das Aktivierungssignal = L ist.

4.4.17.8 Empfang R/T/C von mehreren Stationen



[_comrxim]



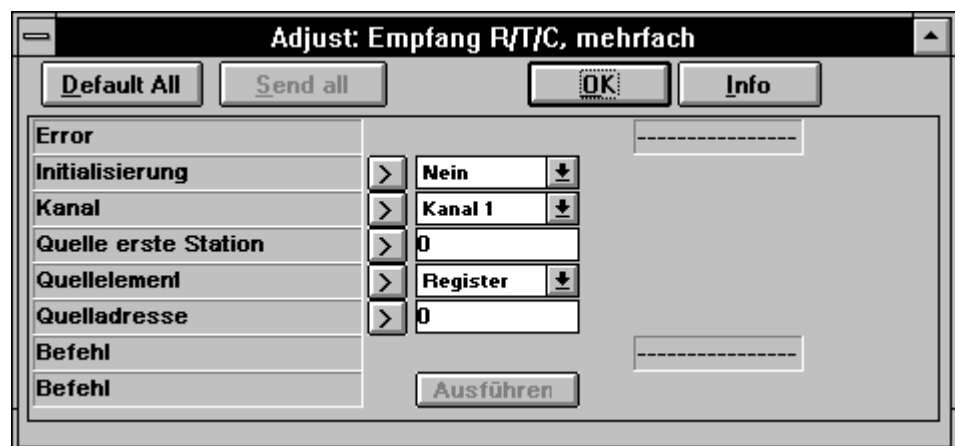
Eingang: – Enable

Ausgänge: = → Ganzzahlformat

Ist der Enable-Eingang = H (binärer Anschluss links an der Funktionsbox), wird je ein im Einstellfenster definiertes numerisches Element (Register, Timer oder Counter) von jeder Partner-Station in die eigene Station übertragen.

Die Funktion kann von 1 bis 20 Ausgangselemente erweitert werden (1 Element pro Station). Die Größe der FBox (ausziehbar von 1 auf 20 Ausgangselemente) definiert auch die Anzahl aufeinanderfolgenden Stationen, ab welchen gelesen werden soll. Die Station mit der tiefsten Adresse wird im Einstellfenster angegeben.

Zusätzliche Angaben, wie Kanal-Nummer, Typ des Quellelements usw. werden auch im Einstellfenster angegeben.



Diese Funktion kann nur auf einer Master-PCD, welche mit der Schnittstellen-Parameter FBox als S-BUS-Master assigniert wurde, ausgeführt werden.

Die Funktion erlaubt den Empfang je eines numerischen Elementes von 1 - 20 Slave-Stationen. Die Uebertragung erfolgt zyklisch mit der maximal möglichen Geschwindigkeit und zwar solange, wie das Enable-Signal = H ist. Erhält der Enable-Eingang eine positive Flanke, wird mindestens jede

Station einmal abgefragt, auch wenn das Enable-Signal kürzer als ein Kommunikationszyklus sein sollte.

Mögliche Fehlerquellen:

Fehlende Assignierung: Der gewählte Kanal wurde nicht oder fehlerhaft assigniert. (Kein oder fehlerhafter SASI). Eine andere Assignierung als durch die SASI-Funktion ist nicht zulässig.

Nicht Master: Der Kanal ist nicht als Master assigniert.

Nicht S-BUS: Der Kanal ist nicht im S-BUS-Modus assigniert. Nur das S-BUS-Protokoll unterstützt diese Funktion.

SRXM: Es wurde ein Fehler bei der Ausführung der Instruktion SRXM festgestellt. Dies dürfte bei Verwendung der vorliegenden Funktion nicht auftreten, es sei denn, ein anderer Programmteil habe auch Zugriff auf diesen Kanal.

Diagnose: Es wurde ein Kommunikationsfehler festgestellt. Dieser kann dank der Diagnosemöglichkeiten der SASI-Funktion im Detail analysiert werden.

Zu gross: Die FBox wurde ausgezogen, wobei die letzte Station (definierte 1. Station + Auszugsindex) höher als 254 wird. Nur die Stationen im Adressbereich 1 bis 254 sind erlaubt.

Step: Der Kanal wurde mittels des Modus "Debug" der Funktion SASI in "Step" gebracht. Es kann keine Kommunikation mehr stattfinden bis der Kanal wieder in "Run" gebracht wurde.

Parametrierung:

Initialisierung: Diese Option erlaubt eine Uebertragung während dem Start der PCD, auch wenn das binäre Aktivierungssignal = L ist. Dies erlaubt ein Initialisieren der Elemente der Slave-Station nach einem Halt der Master-Station.

Kanal: Zu verwendende Kanalnummer.

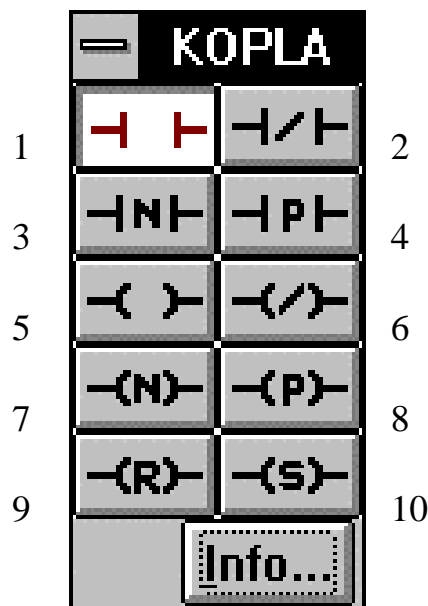
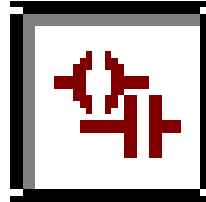
Station: Nummer der Slavestation, wohin die Daten übertragen werden oder wurden.

Typ und Adresse: Adresse und Typ des Elementes der Slave-Stationen welche übertragen werden sollen. Pro Station kann nur 1 Element übertragen werden.

Die Taste "Befehl" erlaubt das Ausführen einer Uebertragung, auch wenn das Aktivierungssignal = L ist.

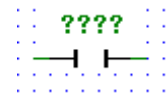
4.5 Die Funktionen des KOPLA (Ladder Diagram: LD)

Der Aufruf des KOPLA geschieht direkt von der Toolbox des FUPLA aus und nicht über die Funktionswahl via das "FBox Selection"-Menü.



Die einzelnen Funktionen des KOPLA sind die folgenden:

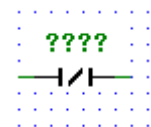
- 4.5.1 Eingang, im Ein-Zustand aktiv (normally open)
- 4.5.2 Eingang, im Aus-Zustand aktiv (normally closed)
- 4.5.3 Dynamischer Eingang, auf negative Flanke
- 4.5.4 Dynamischer Eingang, auf positive Flanke
- 4.5.5 Binärer Ausgang, Status des ACCU (OUT)
- 4.5.6 Binärer Ausgang, inverser Status des ACCU
- 4.5.7 Binärer Ausgang, Setzen bei negativer Flanke
- 4.5.8 Binärer Ausgang, Setzen bei positiver Flanke
- 4.5.9 Binärer Ausgang, Rücksetzen (RES)
- 4.5.10 Binärer Ausgang, Setzen (SET)

Kontakt-Plan (Ladder diagram)**(sfupkopI)****4.5.1****Eingang, im Ein-Zustand aktiv (normally open)**

[_contact]

Der Status des linken Anschlusses wird zum rechten Anschluss kopiert, wenn das im Eintragungsfeld bezeichnete Element = H ist. (I, O, F, T, C). Ist das Element = L, ist der rechte Anschluss = L.

Diese Funktion entspricht dem "STH" oder dem "ANH" aus dem PCD-Befehlssatz.

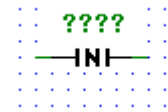
4.5.2**Eingang, im Aus-Zustand aktiv (normally closed)**

[_contactc]

Der Status des linken Anschlusses wird zum rechten Anschluss kopiert, wenn das im Eintragungsfeld bezeichnete Element = L ist. (I, O, F, T, C). Ist das Element = H, ist der rechte Anschluss = L.

Diese Funktion entspricht dem "STL" oder dem "ANL" aus dem PCD-Befehlssatz.

4.5.3 Dynamischer Eingang, auf negative Flanke

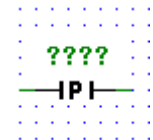


[_contactnv]

Der Status des rechten Anschlusses wird = H, wenn eine negative Flanke am bezeichneten Element erkannt wurde und wenn der Status des linken Anschlusses = H ist. In allen anderen Fällen ist der Status des rechten Anschlusses = L.

Diese Funktion entspricht dem "STL" bzw. "ANL" - "DYN" aus dem PCD-Befehlssatz.

4.5.4 Dynamischer Eingang, auf positive Flanke

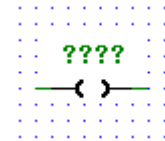


[_contactps]

Der Status des rechten Anschlusses wird = H, wenn eine positive Flanke am bezeichneten Element erkannt wurde und wenn der Status des linken Anschlusses = H ist. In allen anderen Fällen ist der Status des rechten Anschlusses = L.

Diese Funktion entspricht dem "STH" bzw. "ANH" - "DYN" aus dem PCD-Befehlssatz.

4.5.5 Binärer Ausgang, Status des ACCU (OUT)

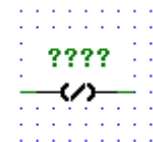


[_coil]

Der Status am linken Anschluss wird zum im Eintragsfeld bezeichneten Element (O, F) und zum rechts angeschlossenen Element weitergegeben.

Diese Funktion entspricht dem "OUT" aus dem PCD-Befehlssatz.

4.5.6 Binärer Ausgang, inverser Status des ACCU

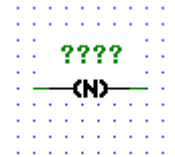


[_coil]

Der Status am linken Anschluss wird zum rechts angeschlossenen Element weitergegeben. Der inverse Status wird zum im Eintragsfeld bezeichneten Element (O, F) übertragen.

Beispiel: Ist der Status am linken Anschluss = L, wird der bezeichnete Ausgang oder das bezeichnete Flag = H und umgekehrt.

4.5.7

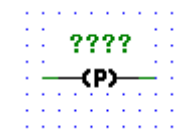
Binärer Ausgang, Setzen bei negativer Flanke

[_coilnv]

Der Status des bezeichneten Elementes (O, F) wird = H, wenn diese Funktion zwischen 2 Programmzyklen eine negative Flanke am linken Eingang erkannt hat.

Der Status am linken Anschluss wird zum rechten Anschluss kopiert.

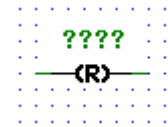
4.5.8

Binärer Ausgang, Setzen bei positiver Flanke

[_coilps]

Der Status des bezeichneten Elementes (O, F) wird = H, wenn diese Funktion zwischen 2 Programmzyklen eine positive Flanke am linken Eingang erkannt hat.

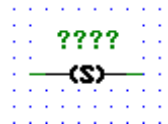
Der Status am linken Anschluss wird zum rechten Anschluss kopiert.

4.5.9 Binärer Ausgang, Rücksetzen (RES)

[_coilreset]

Der Status des bezeichneten Elementes (O, F) wird und bleibt = L, wenn am linken Anschluss ein H-Signal erkannt wird.

Diese Funktion entspricht dem "RES" aus dem PCD-Befehlssatz.

4.5.10 Binärer Ausgang, setzen (SET)

[_coilset]

Der Status des bezeichneten Elementes (O, F) wird und bleibt = H wenn am linken Anschluss ein H-Signal erkannt wird.

Diese Funktion entspricht dem "SET" aus dem PCD-Befehlssatz.

Absender:

Firma
Abteilung
Name
Adresse

Tel.

Datum

An:

SAIA-Burgess Electronics AG
Bahnhofstrasse 18
CH-3280 Murten (Schweiz)
<http://www.saia-burgess.com>

GB: Electronic Controllers

Funktionsfamilien des FUPLA und des KOPLA
PG4 - Version 1.3

Falls Sie Vorschläge zu SAIA[®] PCD zu machen oder Fehler in diesem Handbuch gefunden haben, sind wir Ihnen für einen kurzen Bericht dankbar.

Ihre Vorschläge: