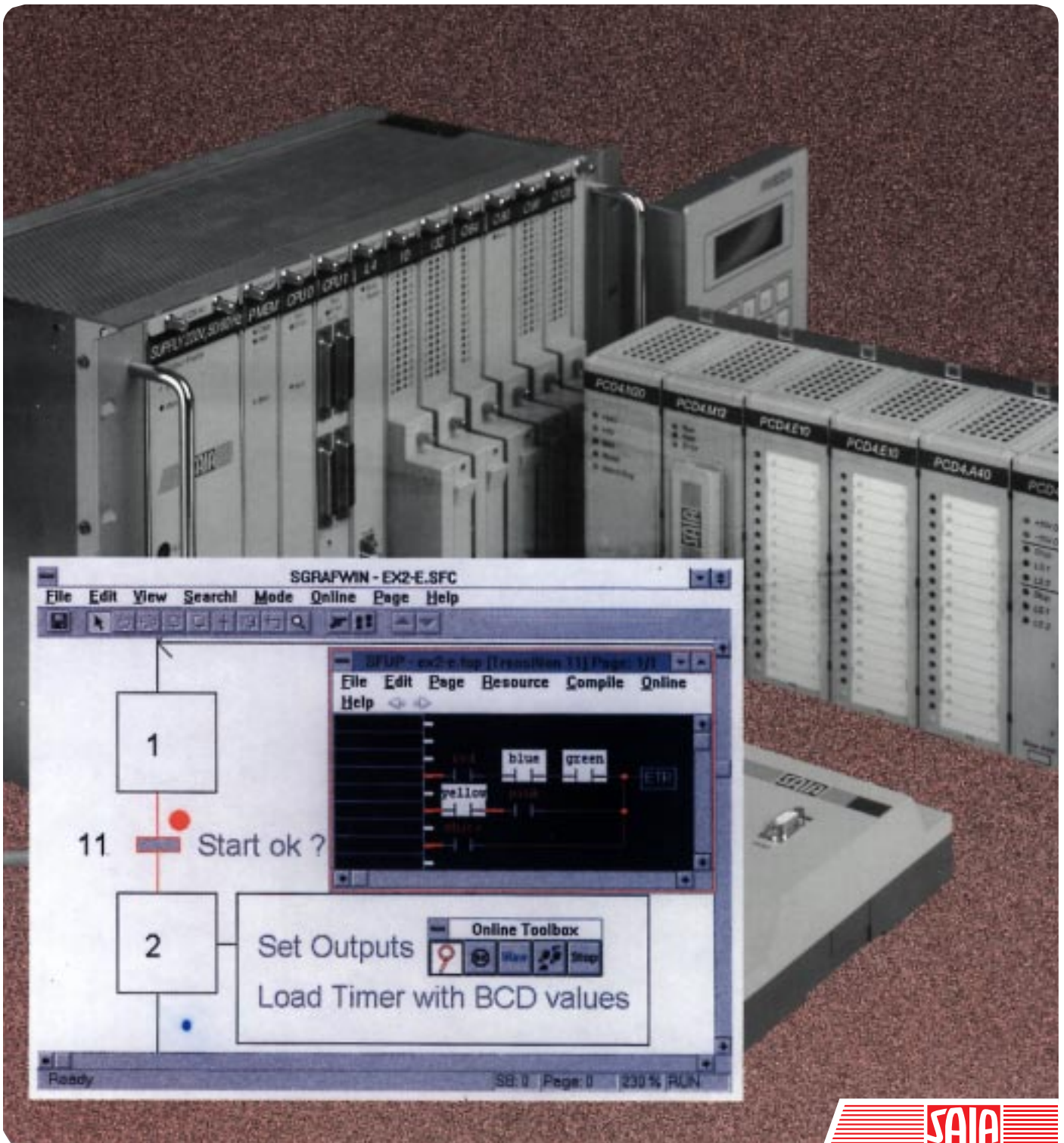


**SAIA® PCD**  
Process Control Devices

**Les fonctions du  
FUPLA et du KOPLA**



---

## Sociétés SAIA-Burgess

<b>Suisse</b>	SAIA-Burgess Electronics SA Rue de Fribourg 33 CH-3280 Morat ☎ 026 672 77 77, Fax 026 670 19 83	<b>France</b>	SAIA-Burgess Electronics Sàrl. 10, Bld. Louise Michel F-92230 Gennevilliers ☎ 01 46 88 07 70, Fax 01 46 88 07 99
<b>Allemagne</b>	SAIA-Burgess Electronics GmbH Daimlerstrasse 1k D-63303 Dreieich ☎ 06103 89 060, Fax 06103 89 06 66	<b>Pays-Bas</b>	SAIA-Burgess Electronics B.V. Hanzeweg 12c NL-2803 MC Gouda ☎ 0182 54 31 54, Fax 0182 54 31 51
<b>Autriche</b>	SAIA-Burgess Electronics Ges.m.b.H. Schallmooser Hauptstrasse 38 A-5020 Salzburg ☎ 0662 88 49 10, Fax 0662 88 49 10 11	<b>Belgique</b>	SAIA-Burgess Electronics Belgium Avenue Roi Albert 1er, 50 B-1780 Wemmel ☎ 02 456 06 20, Fax 02 460 50 44
<b>Italie</b>	SAIA-Burgess Electronics S.r.l. Via Cadamosto 3 I-20094 Corsico MI ☎ 02 48 69 21, Fax 02 48 60 06 92	<b>Hongrie</b>	SAIA-Burgess Electronics Automation Kft. Liget utca 1. H-2040 Budaörs ☎ 23 501 170, Fax 23 501 180

---

## Représentations

<b>Grande-Bretagne</b>	Canham Controls Ltd. 25 Fenlake Business Centre, Fengate Peterborough PE1 5BQ UK ☎ 01733 89 44 89, Fax 01733 89 44 88	<b>Portugal</b>	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
<b>Danemark</b>	Malthe Winje Automation AS Håndværkerbyen 57 B DK-2670 Greve ☎ 70 20 52 01, Fax 70 20 52 02	<b>Espagne</b>	Tecnosistemas Medioambientales, S.L. Poligono Industrial El Cabril, 9 E-28864 Ajalvir, Madrid ☎ 91 884 47 93, Fax 91 884 40 72
<b>Norvège</b>	Malthe Winje Automasjon AS Haukelivn 48 N-1415 Oppegård ☎ 66 99 61 00, Fax 66 99 61 01	<b>Tchéquie</b>	ICS Industrie Control Service, s.r.o. Modranská 43 CZ-14700 Praha 4 ☎ 2 44 06 22 79, Fax 2 44 46 08 57
<b>Suède</b>	Malthe Winje Automation AB Truckvägen 14A S-194 52 Upplands Väsby ☎ 08 795 59 10, Fax 08 795 59 20	<b>Pologne</b>	SABUR Ltd. ul. Druzynowa 3A PL-02-590 Warszawa ☎ 22 844 63 70, Fax 22 844 75 20
<b>Suomi/ Finlande</b>	ENERGEL OY Atomitie 1 FIN-00370 Helsinki ☎ 09 586 2066, Fax 09 586 2046		
<b>Australie</b>	Siemens Building Technologies Pty. Ltd. Landis & Staefa Division 411 Ferntree Gully Road AUS-Mount Waverley, 3149 Victoria ☎ 3 9544 2322, Fax 3 9543 8106	<b>Argentine</b>	MURTEN S.r.l. Av. del Libertador 184, 4° "A" RA-1001 Buenos Aires ☎ 054 11 4312 0172, Fax 054 11 4312 0172

---

## Service après-vente

<b>USA</b>	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**

**Outils de programmation pour MS WINDOWS**

# **Les fonctions du FUPLA et du KOPLA**

**PG4 - Version 1.3**

SAIA-Burgess Electronics SA 1996. Tous droits réservés  
Edition 26/749 F1 - 05.1996

Sous réserve de modifications

# Mise à jour

---

Manuel : Les fonctions du FUPLA et du KOPLA - PG4 Version 1.3 - édition F1

Date	Chapitre	Page	Description
27.10.2000	---	---	Petites mises à jour pour la « Homepage » support

# Table des matières

---

Dans ce manuel, toutes les fonctions du FUPLA standard et du KOPLA sont décrites.

Le manuel représente le chapitre 4.4 et 4.5 du manuel « Outils de programmation PG4 (26/748 F)

Les descriptions des fonctions sont normalement identiques aux "Infos" des fonctions sur l'écran.

<b>Vue générale des fonctions des diverses familles</b>		<b>Page</b>
<b>4.4</b>	<b>Les familles des fonctions FUPLA</b>	<b>4-5</b>
4.4.1	Fonctions binaires	4-15
4.4.2	Bascule (flip-flops)	4-27
4.4.3	Compteurs	4-37
4.4.4	Temporisateurs	4-43
4.4.5	Clignoteurs	4-53
4.4.6	Arithmétique en format entier	4-55
4.4.7	Arithmétique en format flottant (virgule flottante)	4-71
4.4.8	Convertisseurs (binaire- entier - flottant)	4-85
4.4.9	Adressage indirect	4-99
4.4.10	Move Data	4-107
4.4.11	Afficheurs	4-117
4.4.12	Fonctions GRAFTEC	4-121
4.4.13	Fonctions spéciales (Chien de garde)	4-129
4.4.14	Modules d'entrées/sorties analogiques	4-131
4.4.15	Régulation (PID)	4-141
4.4.16	Fonctions utilisateur	4-143
4.4.17	Communication série	4-149
<b>4.5</b>	<b>Plan de contact (KOPLA)</b>	<b>4-165</b>

**Notes personnelles :**



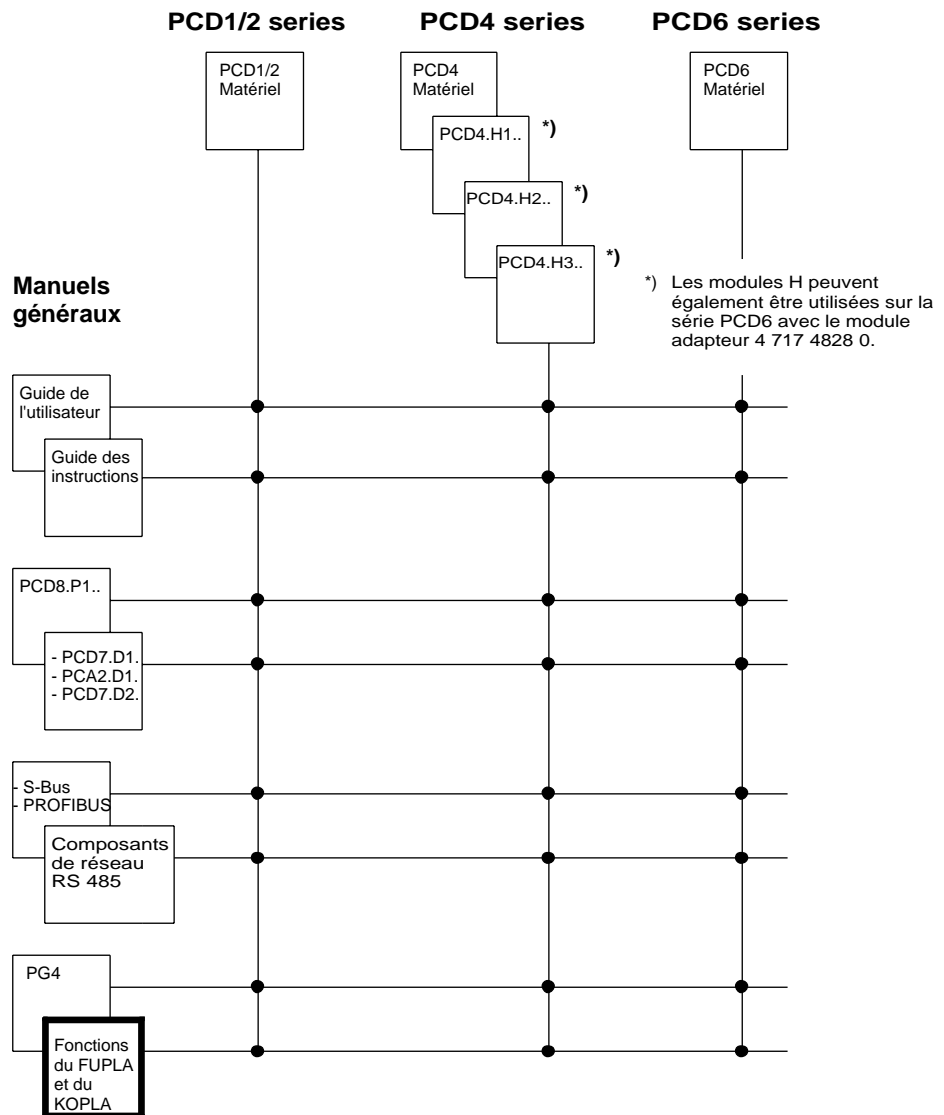
**Veillez prendre note :**

De nombreux manuels techniques précis et détaillés ont été élaborés par SAIA afin de faciliter la mise en oeuvre de ses automates programmables; ils s'adressent à un personnel qualifié ayant suivi au préalable nos stages de formation.

Afin d'obtenir une performance optimale de votre PCD SAIA, suivez scrupuleusement les instructions de montage, de câblage, de programmation et de mise en service reportées dans ce manuel. De cette façon, vous deviendrez vous aussi un utilisateur enthousiaste des PCD SAIA.

Vos suggestions ou recommandations concernant une amélioration des manuels sont bienvenues. Veuillez les reporter sur le formulaire qui se trouve à la dernière page de ce document.

**Vue d'ensemble de la gamme et de la documentation PCD**



## **Fiabilité et sécurité des automates électroniques**

---

Soucieuse d'offrir à sa clientèle des produits performants, SAIA voue une attention particulière au design, au développement et à la production de ses produits:

- technologie de pointe
- conformité aux normes
- certification ISO 9001
- approbation internationale: ex. Germanischer Lloyd, Det Norske Veritas, CE marking ...
- choix de composants de haute qualité
- contrôles de qualité durant le processus de production
- essais en condition réelle de fonctionnement
- déverminage (à 85°C pendant 48h)

Malgré ces constants efforts, l'excellente qualité qui en résulte a ses limites. Il est donc nécessaire, par exemple, de tenir compte des défauts naturels des composants. Pour cette raison, SAIA SA fournit une garantie selon les "Conditions générales de livraison".

L'ingénieur de site doit à son tour contribuer au fonctionnement fiable de l'installation. Il est responsable que l'automate soit utilisé conformément aux spécifications techniques. Il s'assure qu'aucune contrainte excessive, température, surtension, champ magnétique, contrainte mécanique soit exercée sur les appareils.

En outre, l'ingénieur de site est responsable de s'assurer qu'aucun produit défectueux soit la cause de blessures, d'accident, ou de dommages à biens d'autrui. Les règles de sécurité en vigueur doivent être respectées en toutes circonstances. Les défauts dangereux doivent être signalés spécialement et des mesures préventives doivent être prises afin d'éviter toutes conséquences graves. Par exemple, les sorties importantes pour la sécurité doivent être liées à des entrées pouvant être commandées par le software. Les éléments de diagnostic du PCD, soit : chien de garde, bloc d'organisation exceptionnel (XOB), instructions de test ou de diagnostic doivent être utilisés lorsque la situation l'exige.

Si tous ces points sont respectés, le PCD SAIA sera pour vous pendant de longues années un automate programmable moderne, fiable pour le contrôle, la régulation et la commande de votre installation.



## 4.4 Les familles de fonctions FUPLA

---

**Vue générale des fonctions des diverses familles.**  
(réparties selon fonction et utilité \*)

### 4.4.1 Fonctions binaires

- 4.4.1.1 ET de 2 à 10 entrées
- 4.4.1.2 OU de 2 à 10 entrées
- 4.4.1.3 OU EXCL. de 2 à 10 entrées
- 4.4.1.4 Mouvement
- 4.4.1.5 Dynamisation - Détection d'un front
  
- 4.4.1.6 Haut - Positionner "H" (high)
- 4.4.1.7 Bas - Positionner "L" (low)
- 4.4.1.8 Non connecté
  
- 4.4.1.9 Multiplexage avec présélection binaire
- 4.4.1.10 Multiplexage avec présélection numérique
- 4.4.1.11 Démultiplexage avec présélection binaire
- 4.4.1.12 Démultiplexage avec présélection numérique
  
- 4.4.1.13 Passage d'entrées/sorties
- 4.4.1.14 Passage de variables
  
- 4.4.1.15 Pair, 2 - 10 entrées
- 4.4.1.16 Impair, 2 - 10 entrées

### 4.4.2 Bascule (Flip-Flops)

- 4.4.2.1 Bascule pas à pas (bistable simple)
- 4.4.2.2 Bascule type "D"
  
- 4.4.2.3 Bascule type "RS" dynamisée avec priorité sur "R"
- 4.4.2.4 Bascule type "RS" dynamisée avec priorité sur "S"
  
- 4.4.2.5 Bascule type "JK"
  
- 4.4.2.6 Bascule type "RS" temporisée avec priorité sur "R"
- 4.4.2.7 Bascule type "RS" temporisée avec priorité sur "S"
  
- 4.4.2.8 Bascule type "RS" avec priorité sur "R"
- 4.4.2.9 Bascule type "RS" avec priorité sur "S"

\*) Dans FUPLA, toutes les fonctions sont réparties automatiquement par ordre alphabétique.

**4.4.3 Compteurs**

- 4.4.3.1 Comptage avec présélection
- 4.4.3.2 Décomptage avec présélection
- 4.4.3.3 Comptage avec mise à zéro
- 4.4.3.4 Comptage / Décomptage avec présélection
- 4.4.3.5 Comptage / Décomptage avec présélection et mise à zéro

**4.4.4 Temporisateurs**

- 4.4.4.1 Enclenchement retardé
- 4.4.4.2 Enclenchement retardé avec mémorisation
- 4.4.4.3 Impulsion unique
  
- 4.4.4.4 Déclenchement retardé
- 4.4.4.5 Enclenchement et déclenchement retardés
- 4.4.4.6 Déclenchement retardé avec reset
- 4.4.4.7 Impulsion
  
- 4.4.4.8 Chronomètre
- 4.4.4.9 Horloge du PCD
  
- 4.4.4.10 Start retardé

**4.4.5 Clignoteurs**

- 4.4.5.1 Clignoteur 1 temps
- 4.4.5.2 Clignoteur 2 temps
  
- 4.4.5.3 Signal d'échantillonnage

<b>4.4.6</b>	<b>Arithmétique en format entier</b>
4.4.6.1	Addition
4.4.6.2	Soustraction
4.4.6.3	Multiplication
4.4.6.4	Division
4.4.6.5	Racine carrée
4.4.6.6	Moyenne
4.4.6.7	Constante
4.4.6.8	Absolue
4.4.6.9	ET sur mot
4.4.6.10	OU sur mot
4.4.6.11	OU EXCL. sur mot
4.4.6.12	Inverse
4.4.6.13	Egal à
4.4.6.14	Plus grand ou égal à
4.4.6.15	Plus grand que
4.4.6.16	Plus petit ou égal à
4.4.6.17	Plus petit que
4.4.6.18	Egal à zéro
4.4.6.19	Limite
4.4.6.20	Maximum
4.4.6.21	Minimum
4.4.6.22	Copie
4.4.6.23	Copie conditionnelle
4.4.6.24	Copie et sauve
4.4.6.25	Commute
4.4.6.26	Multiplexage avec présélection binaire
4.4.6.27	Multiplexage avec présélection numérique
4.4.6.28	Démultiplexage avec présélection binaire
4.4.6.29	Démultiplexage avec présélection numérique
4.4.6.30	Décalage à gauche
4.4.6.31	Décalage à droite
4.4.6.32	Rotation à gauche
4.4.6.33	Rotation à droite
4.4.6.34	Copie registre
4.4.6.35	Copie T/C
4.4.6.36	Non connecté

<b>4.4.7</b>	<b>Arithmétique en format flottant</b>
4.4.7.1	Addition
4.4.7.2	Soustraction
4.4.7.3	Multiplication
4.4.7.4	Division
4.4.7.5	Racine carrée
4.4.7.6	Moyenne
4.4.7.7	Constante
4.4.7.8	Absolu
4.4.7.9	Sinus
4.4.7.10	Cosinus
4.4.7.11	ARC tangente
4.4.7.12	Exponentiel
4.4.7.13	Log népérien
4.4.7.14	Egal à
4.4.7.15	Plus grand ou égal à
4.4.7.16	Plus grand que
4.4.7.17	Plus petit ou égal à
4.4.7.18	Plus petit que
4.4.7.19	Egal à zéro
4.4.7.20	Limite
4.4.7.21	Maximum
4.4.7.22	Minimum
4.4.7.23	Copie
4.4.7.24	Copie conditionnelle
4.4.7.25	Copie et sauve
4.4.7.26	Commute
4.4.7.27	Multiplexage avec présélection binaire
4.4.7.28	Multiplexage avec présélection numérique
4.4.7.29	Démultiplexage avec présélection binaire
4.4.7.30	Démultiplexage avec présélection numérique
4.4.7.31	Non connecté

<b>4.4.8</b>	<b>Convertisseurs (binaire-entier-flottant)</b>
4.4.8.1	Binaire en entier à partir de 1 à 8 I/O/F
4.4.8.2	Binaire en entier à partir de 1 à 24 I/O/F
4.4.8.3	Binaire en entier rapide (format PCD)
4.4.8.4	Binaire en entier inversé rapide (format PCA)
4.4.8.5	Entier en binaire sur 1 à 8 O/F
4.4.8.6	Entier en binaire sur 1 à 24 O/F
4.4.8.7	Entier en binaire rapide (format PCD)
4.4.8.8	Entier en binaire inversé rapide (format PCA)
4.4.8.9	BCD en entier à partir de 4 ou 8 I/O/F
4.4.8.10	BCD en entier rapide (format PCD)
4.4.8.11	BCD en entier inversé rapide (format PCA)
4.4.8.12	Entier en BCD sur 4 ou 8 O/F
4.4.8.13	Entier en BCD rapide (format PCD)
4.4.8.14	Entier en BCD inversé rapide (format PCA)
4.4.8.15	1-bit en entier avec shift
4.4.8.16	1-bit en entier LSB
4.4.8.17	Entier en 1-bit avec shift
4.4.8.18	Entier LSB en 1-bit
4.4.8.19	Format flottant en format entier
4.4.8.20	Format entier en format flottant
<b>4.4.9</b>	<b>Adressage indirect</b>
4.4.9.1	Copie vers sorties
4.4.9.2	Lecture d'entrées
4.4.9.3	Copie vers indicateurs
4.4.9.4	Lecture d'indicateurs
4.4.9.5	Copie vers registres en format entier
4.4.9.6	Lecture de registres en format entier
4.4.9.7	Copie vers registres en format flottant
4.4.9.8	Lecture de registres en format flottant
4.4.9.9	Copie vers temporisateurs/compteurs
4.4.9.10	Lecture de temporisateurs/compteurs
4.4.9.11	Adressage indirect de Temporisateur
4.4.9.12	Adressage indirect de Compteur
4.4.9.13	Lecture de l'état logique de T/C

**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
- 4.4.10.4 Move-Out Nibble
- 4.4.10.5 Move-In Digit
- 4.4.10.6 Move-Out Digit
- 4.4.10.7 Move-In Byte
- 4.4.10.8 Move-Out Byte
- 4.4.10.9 Move-In Word
- 4.4.10.10 Move-Out Word

**4.4.11 Afficheurs**

- 4.4.11.1 Module d'affichage PCA2.D12
- 4.4.11.2 Module d'affichage PCA2.D14
- 4.4.11.3 Module d'affichage PCD2.F510, numérique
- 4.4.11.4 Module d'affichage PCD2.F510, texte

**4.4.12 Fonctions GRAFTEC**

- 4.4.12.1 Chargement d'un Temporisateur sans condition
- 4.4.12.2 Chargement conditionnel d'un Temporisateur
- 4.4.12.3 Chargement d'un Compteur sans condition
- 4.4.12.4 Chargement conditionnel d'un Compteur
- 4.4.12.5 Fonction pour incrémenter un Compteur
- 4.4.12.6 Fonction pour décrémenter un Compteur
- 4.4.12.7 Est-ce que Temporisateur est écoulé ?
- 4.4.12.8 Est-ce que Compteur = 0 ?
- 4.4.12.9 Fin Transition
- 4.4.12.10 Charger un Tempo. et attendre la fin de la temporisation
- 4.4.12.11 Charger un Compteur et attendre jusqu'à Compteur = 0

**4.4.13 Fonctions spéciales (Chien de garde)**

- 4.4.13.1 Chien de garde (toujours actif)
- 4.4.13.2 Chien de garde activable

**4.4.14 Modules d'entrées/sorties analogiques**

4.4.14.1	PCD2.W1	Entrées analogiques PCD2.W1 (12 Bit)
4.4.14.2	PCD2.W2	Entrées analogiques PCD2.W2 (10 Bit)
4.4.14.3	PCD2.W4	Sorties analogiques PCD2.W4 (8 Bit)
4.4.14.4	PCD2.W5	Entrées/Sorties analog. PCD2.W5 (12 Bit)
4.4.14.5	PCD4.W1	Entrées/Sorties analog. PCD4.W1 (12 Bit)
4.4.14.6	PCD4.W3	Entrées analog. PCD4.W3 (12 Bit + signe)
4.4.14.7	PCD4.W4	Sorties analogiques PCD4.W4 (8 Bit)
4.4.14.8	PCD6.W1	Entrées/Sorties analog. PCD6.W1 (12 Bit)
4.4.14.9	PCD6.W3	Entrées analog. PCD6.W3 (12 Bit + signe)
4.4.14.10	PCD6.W4	Sorties analogiques PCD6.W4 (8 Bit)

**4.4.15 Régulation**

4.4.15.1	Régulateur PID
----------	----------------

**4.4.16 Fonctions utilisateur**

4.4.16.1	Appel PB
4.4.16.2	Appel FB
4.4.16.3	Appel SB
4.4.16.4	Bloc utilisateur 1
4.4.16.5	Bloc utilisateur 2
4.4.16.6	Bloc utilisateur 3
4.4.16.7	Bloc utilisateur 4
4.4.16.8	Bloc utilisateur 5

**4.4.17 Communication série**

4.4.17.1	Paramètres d'interface série : SASI
4.4.17.2	Paramètres d'interface série : SASI, externe
4.4.17.3	Lecture de 1 à 20 I/O/F
4.4.17.4	Lecture de 1 à 20 R/T/C/Horloge
4.4.17.5	Ecriture de 1 à 20 I/O/F
4.4.17.6	Ecriture de 1 à 20 R/T/C/Horloge
4.4.17.7	Lecture I/O/F, multiple
4.4.17.8	Lecture R/T/C, multiple

**Explications concernant le format et les symboles utilisés pour la description des fonctions :**

**Nom de la fonction**

[\_fupxyz]



**Nom de la fonction**

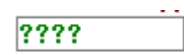
Nom de la fonction du menu FUPLA.(de Fbox sélection . . .)

Les entrées sont placées à gauche de la FBox, par exemple:

Entrées: - En	→	Enable	"-"	entrée binaire (H/L)
> Set	→	Set	">"	entrée binaire dynamique
= Val	→	Valeur	"="	entrée numérique (valeur)

Les sorties sont placées à droite de la Fbox, par exemple:

Sorties: - Q	→	Sortie	"-"	sortie binaire (H/L)
- /Q	→	Sortie inv.	"-"	sortie binaire (H/L)
= R	→	Résultat	"="	sortie numérique (valeur)



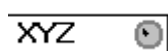
Champ pour l'adresse de base de cette fonction. Normalement "O" ou "o" (output), par ex. "o 16".



Le point au coin inférieur à gauche indique que cette fonction possède une fenêtre d'ajustage.



Exemple pour une fenêtre d'ajustage: 'Afficheur' - 'Module D12'



Les fonctions plus complexes, comme p.ex. la communication, possèdent une LED. Celle-ci est de couleur grise au moment de la programmation. En RUN (online) la LED devient verte, en cas d'erreur rouge. Seules les fonctions avec une LED portent ce symbole.

[fupabc]

(dans la première ligne d'une famille de fonctions) montre le nom du fichier de cette famille, p.ex. (sfuptime) pour la famille des temporisateurs: "sfuptime.def", "sfuptime.hlp", "sfuptime.idx", "sfuptime.lib".

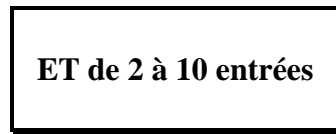
[\_fupxyz]:

Nom de la fonction dans les fichiers ".lib", p.ex. (ondel) pour retardement à l'enclenchement. Les noms sont identiques pour toutes les versions de langage et servent de "cross reference" des fonctions entre les différents langages.

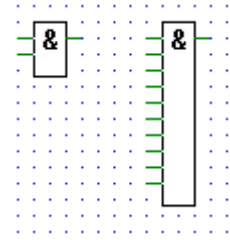
Le nom interne de la fonction peut être visualisé en cliquant sur 'Advanced Info':



**Notes personnelles :**

**4.4.1 Fonctions binaires****[fupbina]****4.4.1.1 ET de 2 à 10 entrées**

[\_band]



Entrées/Sorties: – toutes → format binaire

Résultat de la fonction binaire ET par rapport aux entrées. La sortie est haute si toutes les entrées sont hautes.

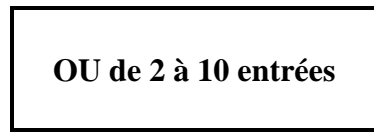
La boite est étirable de 2 à 10 entrées.

Table de vérité pour 2 entrées:

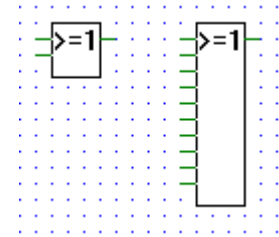
In1	In2	Out
0	0	0
0	1	0
1	0	0
1	1	1

## 4.4.1.2

## OU de 2 à 10 entrées



[\_bor]



Entrées/Sorties: – toutes → format binaire

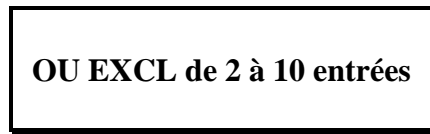
Le résultat de la fonction binaire OU en fonction des entrées.

La sortie est haute si une ou plus d'une entrée est haute. (OU inclusif) la boîte est étirable de 2 à 10 entrées.

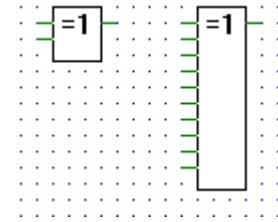
Table de vérité pour un OU de 2 entrées:

In1	In2	Out
0	0	0
0	1	1
1	0	1
1	1	1

### 4.4.1.3 OU Exclusif de 2 à 10 entrées



[\_bxor]



Entrées/Sorties: – toutes → format binaire

Le résultat de la fonction binaire OU EXCLUSIF en fonction des entrées.

La sortie est haute si une et une seule entrée est haute.

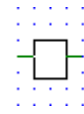
La boîte est étirable de 2 à 10 entrées.

Table de vérité pour un OU EXCLUSIF de 2 entrées:

In1	In2	Output
0	0	0
0	1	1
1	0	1
1	1	0

---

#### 4.4.1.4 Mouvement Entrée → Sortie



[\_bmove]

Entrées/Sorties: – toutes → format binaire

Cette fonction permet de réaliser un transfert direct entre l'état logique binaire d'entrée et une sortie.

---

#### 4.4.1.5 Dynamisation - Détection d'un front



[\_bdyn2]

Cette fonction permet la détection du front montant.

La sortie de cette fonction est haute seulement lorsque l'entrée passe de bas à haut. Elle permet de dynamiser l'entrée d'une fonction ne changeant d'état que lorsqu'il y a un niveau logique H.

Dans la fenêtre d'ajustage 3 options pour le comportement au démarrage de l'automate peuvent être choisies:



"Toujours": Un flanc est toujours pris en compte si l'entrée est à haut au démarrage.

"Jamais": Un flanc n'est jamais pris en compte si l'entrée est à haut au démarrage.

"En=0/Hors=1": Un flanc est pris en compte si l'entrée était à bas au déclenchement et haut au démarrage.

Attention:

Cette dernière option peut entraîner des phénomènes aléatoires lorsque après une correction du programme les flags ne sont pas mis à bas!

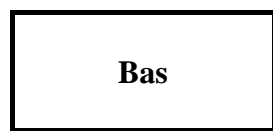
**4.4.1.6 Positionner "H" (high)**

[\_bhigh]

Sortie: – → format binaire

Cette fonction donne à sa sortie un état logique haut.

---

**4.4.1.7 Positionner "L" (Low)**

[\_blow]

Sortie: – → format binaire

Cette fonction donne à sa sortie un état logique bas.

---

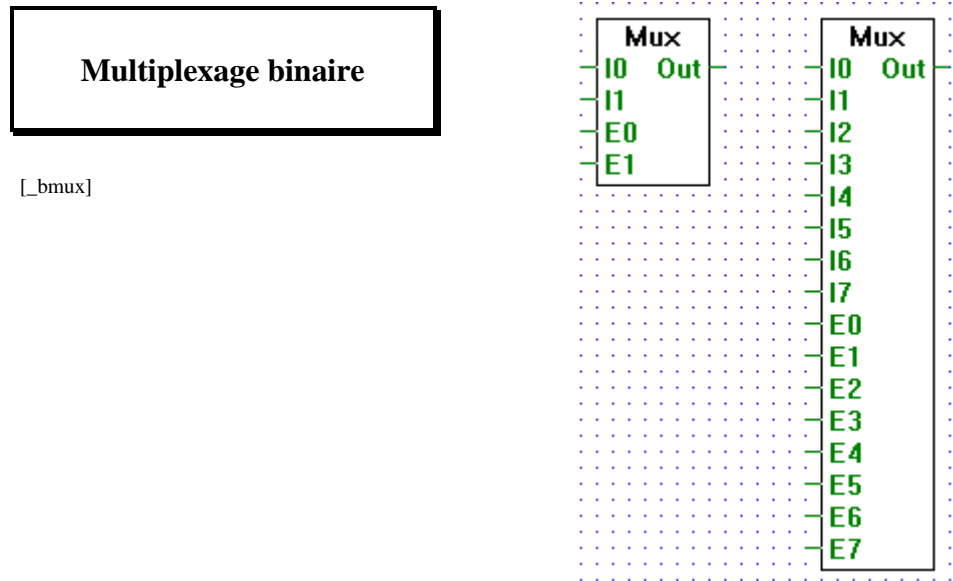
**4.4.1.8 Non connecté**

[\_bnotcn]

La boîte "Non connecté" permet la fermeture d'une liaison binaire inutilisée.

---

#### 4.4.1.9 Multiplexage avec présélection binaire



Entrées/Sorties: – toutes → format binaire

Cette fonction réalise le transfert de l'état logique binaire de `I0` à `I7` dans la sortie "Out" quand la validation correspondante `E0` à `E7` est haute.

Elle est étirable de 2 à 8 éléments.

Si toutes les validations sont basses, la sortie est basse.

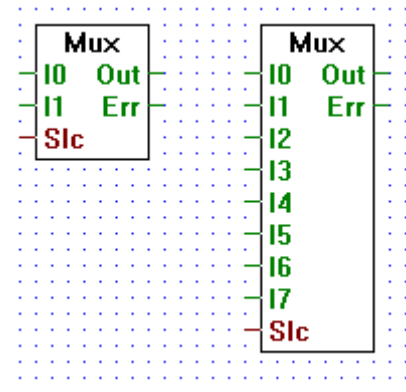
La ligne de validation `E0` a la plus haute priorité et `E7` la plus basse. En conséquence, si plusieurs entrées de validation sont hautes en même temps, la plus haute sera transférée dans la sortie en priorité.



#### 4.4.1.10 Multiplexage avec présélection numérique



[\_bmux2]



Entrées:       – I0:  
                  |     format binaire  
                  – I7:  
                  = Slc  format entier

Sorties :       – Out  format binaire  
                  – Err  format binaire

Cette fonction réalise le transfert de l'état logique de l'entrée "In" sélectionnée par "Slc" dans la sortie "Out".

( Par ex. si Slc = 0, l'état logique de I0 est transféré dans "Out").

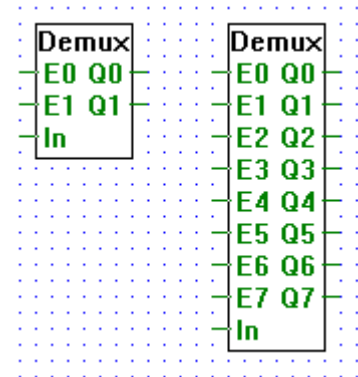
La boîte est étirable de 2 à 8 entrées "In".

Si "Slc" est en dehors de l'échelle (< 0 ou > n), "Out" est mise bas et "Err" est mis haut.

#### 4.4.1.11 Démultiplexage avec présélection binaire



[\_bdemux]



Entrées/Sorties: – toutes → format binaire

Cette fonction permet le transfert de l'état logique de "In" dans la sortie (Q0..Q7) quand la validation correspondantes (E0..E7) est active. Elle réalise l'aiguillage des informations logique de "In" vers la sortie ou la validation est vraie.

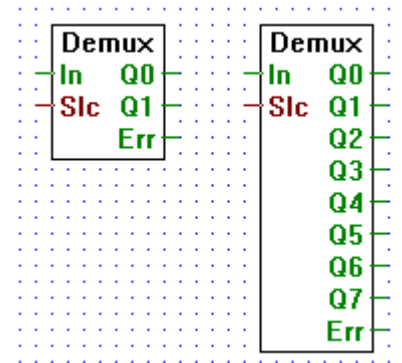
La boîte est étirable de 2 à 8 entrées.

Quand une ligne de validation est basse (E0..E7) la sortie correspondante (Q0...Q7) est mise bas.

#### 4.4.1.12 Démultiplexage avec présélection numérique



[\_bdemux2]



Entrées:      – In:  format binaire  
                   = Slc: format entier

Sorties:       – Q0  
                   |     format binaire  
                   – Q7  
                   – Err: format binaire

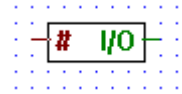
Cette fonction permet le transfert de l'état logique de "In" dans la sortie (Q0..Qn) sélectionnée par "Slc". Par exemple, si "Slc" = 1 l'entrée "In" est transférée dans "Q1".

Les sorties non sélectionnées sont mise bas.

La boîte est étirable de 2 à 8 sorties.

"Err" devient haut quand "Slc" est en dehors de l'échelle, ou lorsqu'elle est négative ou plus grande que "In".

#### 4.4.1.13 Passage d'entrées/sorties



[\_bioind]

Entrée:           = #    I/O adresse:   format entier  
Sortie:           - I/O  I/O état:       format binaire

Donne l'état de l'entrée ou de la sortie numéro #.

Par exemple, si l'entrée # = 5, l'entrée ou la sortie 5 est dans la sortie I/O.

---

#### 4.4.1.14 Passage de variable



[\_bflgin]

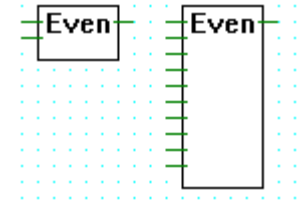
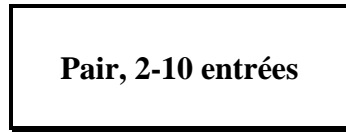
Entrée:           = #    Flag adresse:   format entier  
Sortie:           - Flg  Flag état:       format binaire

Cette fonction donne l'état de la variable numéro #.

Par exemple, si l'entrée # vaut 5, l'état de la variable F 5 est dans la sortie Flg.

---

**4.4.1.15 Pair, 2-10 entrées**



[\_beven]

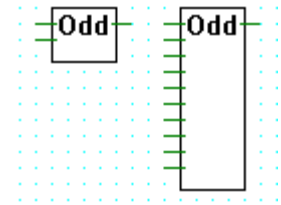
Entrées: – format binaire  
 Sortie: – format binaire

La sortie est haute si le nombre d'entrées qui sont hautes est pair. Si ce n'est pas le cas, la sortie est basse.

Par ex. pour 3 entrées:

In1	In2	In2	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 Impair, 2-10 entrées



[\_bodd]

Entrées: – format binaire  
Sortie: – format binaire

La sortie est haute si le nombre d'entrées qui sont hautes est impair. Si ce n'est pas le cas, la sortie est basse.

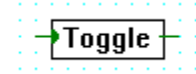
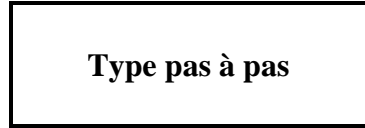
Par ex. pour 3 entrées:

In1	In2	In2	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 Bascule (flip-flops)

[sfupflip]

### 4.4.2.1 Bascule pas à pas (bistable simple)



[\_flitog]

Entrées/Sorties: – toutes → format binaire

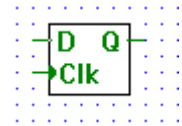
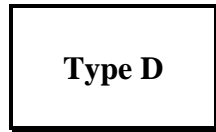
Fonction simple bistable (toggle).

A chaque flanc montant à l'entrée, la sortie est commutée.

A la mise en service, la sortie est basse. Si l'entrée est haute à la mise en service, la sortie devint aussi haute.

Entrée	_ _ _ _ _
Sortie	_ _ _ _ _

4.4.2.2 Bascule type "D"



[\_fid]

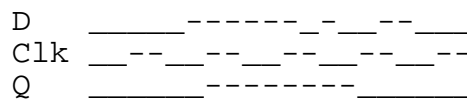
Entrées/Sorties: – toutes → format binaire

Bascule de type D avec enclenchement et déclenchement sur front montant d'un signal horloge.

Sa fonction consiste à mémoriser une information fugitive. La bascule a deux états stable et constitue ainsi une mémoire binaire élémentaire.

L'état de "D" est mémorisé lorsque "Clk" passe de "Bas" à "Haut".

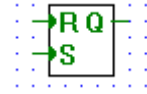
La sortie "Q" est toujours dans le même état que "D" mémorisé.





4.4.2.3 Bascule type "RS" dynamisée avec priorité sur "R"

Type RS dynamisée



[\_flirs]

Entrées/Sorties: – toutes → format binaire

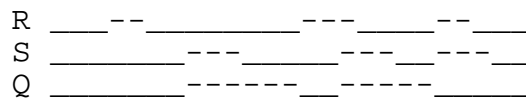
Bascule RS avec priorité au déclenchement.

Quand l'entrée R est "Haut", la sortie Q est mise "Bas".

Quand l'entrée S est "Haut" et que R est "Bas", la sortie Q est mise "Haut".

Quand les deux sont "Haut", la sortie est mise "Bas" car R est prioritaire..

Dans tous les autres cas la sortie reste inchangée.

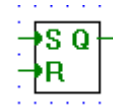


R	S	Q
X	X	inchangé
X	_/-	1
_/-	X	0
_/-	_/-	0

\_/- = flanc montant  
 X = état indéterminé

4.4.2.4 Bascule type "RS" dynamisée avec priorité sur "S"

**Type SR dynamisée**



[\_flisr]

Entrées/Sorties: – toutes → format binaire

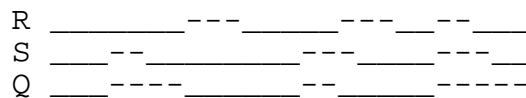
Bascule RS avec priorité à l'enclenchement.

Quand l'entrée R est "Haut" et S est "Bas", la sortie Q est mise "Bas".

Quand l'entrée S est "Haut", la sortie Q est mise "Haut".

Quand les deux sont "Haut", la sortie est mise "Haut" car S est prioritaire..

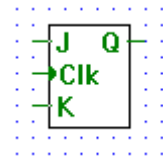
Dans tous les autres cas la sortie reste inchangée.



R	S	Q
X	X	inchangé
X	_/-	1
_/-	X	0
_/-	_/-	1

\_/- = flanc montant  
 X = état indéterminé

4.4.2.5 Bascule type "JK"



[\_flijk]

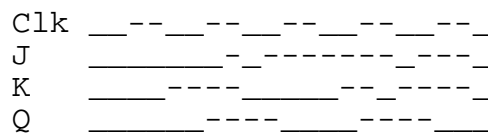
Entrées/Sorties: – toutes → format binaire

Bascule JK dépendant d'un signal d'horloge à front montant.

Les entrées J et K sont lues lorsqu'un front montant est détecté sur "Clk".  
Entre deux fronts, Q reste inchangé.

Dès qu'un front montant est détecté sur "Clk "...

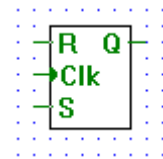
- et que J est "Haut" alors Q est remis "Bas".
- et que K est "Haut" alors Q est mis "Haut".
- et que J et K sont "Haut" alors Q est inversé.
- et que J et K sont "Bas" alors Q reste inchangé.



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)

\_-/- = flanc montant  
X = état indéterminé

**4.4.2.6 Bascule type "RS" temporisée avec priorité sur "R"**



[\_flirclk]

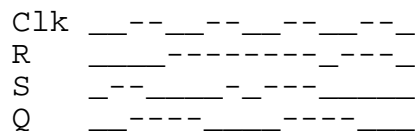
Entrées/Sorties: – toutes → format binaire

Bascule RS avec priorité au repos et déclenchement sur front montant d'un signal d'horloge.

Les entrées R et S ne sont lues qu'au front montant de "Clk". Si aucun front n'est détecté, la sortie Q reste inchangée.

Dès qu'un front est détecté a l'entrée "Clk"...

- et que R est "Haut", alors Q est mis "Bas".
- et que S est "Haut", alors Q est mis "Haut".
- et que R et S sont "Haut", alors Q est mis "Bas" car R est prioritaire.
- et que R et S sont "Bas", alors Q reste inchangé.

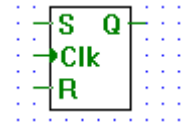


Clk	R	S	Q
X	X	X	inchangé
_/-	0	0	inchangé
_/-	0	1	1
_/-	1	0	0
_/-	1	1	1

\_/ - = flanc montant  
 X = état indéterminé

4.4.2.7 Bascule type "RS" temporisée avec priorité sur "S"

**Type SR temporisée**



[\_flisrelk]

Entrées/Sorties: – toutes → format binaire

Bascule RS avec priorité au travail et changement d'état sur front montant d'un signal d'horloge.

Les entrées R et S ne sont lues qu'au front montant de "Clk". Si aucun front n'est détecté, la sortie Q reste inchangée.

Dès qu'un front est détecté a l'entrée "Clk"...

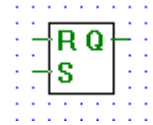
- et que R est "Haut", alors Q est mis "Bas".
- et que S est "Haut", alors Q est mis "Haut".
- et que R et S sont "Haut", alors Q est mis "Haut" car S est prioritaire.
- et que R et S sont "Bas", alors Q reste inchangé.

```
Clk  ___--___--___--___--
R    _____-_____-
S    ___-_____-_____-
Q    _____-_____-
```

Clk	R	S	Q
X	X	X	inchangé
_-/-	0	0	inchangé
_-/-	0	1	1
_-/-	1	0	0
_-/-	1	1	1

\_-/- = flanc montant  
 X = état indéterminé

4.4.2.8 Bascule type "RS" avec priorité sur "R"



[\_flirsndyn]

Entrées/Sorties: – toutes → format binaire

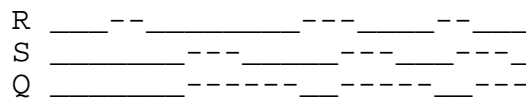
Bascule RS avec priorité au déclenchement. Chaque front de R ou de S engendre un changement d'état de la sortie Q.

Quand l'entrée R passe de "Bas" à "Haut, la sortie Q est remise "Bas".

Quand l'entrée S passe de "Bas" à "Haut, la sortie Q est mise "Haut".

Quand les deux passent au même instant de "Bas" à "Haut", la sortie est remise "Bas" car R est prioritaire.

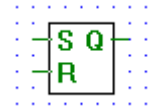
Dans tous les autres cas la sortie reste inchangée.



R	S	Q
0	0	inchangé
1	X	0
0	1	1

X = état indéterminé

4.4.2.9 Bascule type "RS" avec priorité sur "S"



[\_flismdyn]

Entrées/Sorties: – toutes → format binaire

Bascule RS avec priorité à l'enclenchement.

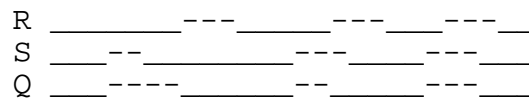
Chaque front de R et de S sont déclenchés.

Quand l'entrée R passe de "Bas" à "Haut", la sortie Q est remise "Bas".

Quand l'entrée S passe de "Bas" à "Haut", la sortie Q est mise "Haut".

Quand les deux passent au même instant de "Bas" à "Haut", la sortie est remise "Haut" car S est prioritaire.

Dans tous les autres cas la sortie reste inchangée.



R	S	Q
0	0	inchangé
X	1	1
1	0	0

X = état indéterminé

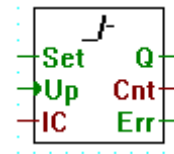
**Notes personnelles :**



### 4.4.3 Compteurs

[sfupcoun]

#### 4.4.3.1 Comptage avec présélection



[\_uppr2]

Entrées:	– Set: Présélection	format binaire
	> Up: +1	format binaire
	= IC: Valeur initiale	format entier
Sorties:	– Q: pos/neg	format binaire
	= Cnt: Conteneur	format entier
	– Err: Error	format binaire

Un compteur est incrémenté chaque fois qu'un flanc montant est détecté sur l'entrée "Up". Le contenu du compteur peut être lu sur la sortie "Cnt".

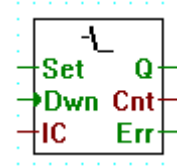
L'entrée "Set" charge le compteur avec la valeur de l'entrée "IC".

Si le contenu du compteur est  $> 0$ , la sortie "Q" est mise au niveau "H". Si le contenu est  $\leq 0$ , la sortie "Q" est mise au niveau "L".

Si le compteur dépasse la capacité, la sortie "Err" est mise au niveau "H".

### 4.4.3.2 Décomptage avec présélection

Décompt. avec présélect.



[\_dwnpr2]

Entrées:      – Set: Présélection      format binaire  
                   > Dwn:-1               format binaire  
                   = IC: Valeur initiale   format entier

Sorties:       – Q: pos/neg           format binaire  
                   = Cnt: Contenne       format entier  
                   – Err: Error           format binaire

Un compteur est décrémenté chaque fois qu'un flanc montant est détecté sur l'entrée "Dwn". Le contenu du compteur peut être lu sur la sortie "Cnt".

L'entrée "Set" charge le compteur avec la valeur de l'entrée "IC".

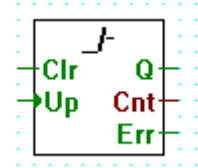
Si le contenu du compteur est  $> 0$ , la sortie "Q" est mise au niveau "H". Si le contenu est  $\leq 0$ , la sortie "Q" est mise au niveau "L".

Si le compteur dépasse la capacité (négative), la sortie "Err" est mise au niveau "H".

### 4.4.3.3 Comptage avec mise à zéro



[L\_up2]



Entrées:	- Clr: Clear	format binaire
	> Up: +1	format binaire
Sorties:	- Q: pos/neg	format binaire
	= Cnt: Conteneue	format entier
	- Err: Error	format binaire

Un compteur est incrémenté chaque fois qu'un flanc montant est détecté sur l'entrée "Up". Le contenu du compteur peut être lu sur la sortie "Cnt".

L'entrée "Clr" efface le contenu du compteur.

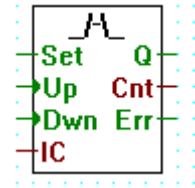
Si le contenu du compteur est  $> 0$ , la sortie "Q" est mise au niveau "H". Si le contenu est  $\leq 0$ , la sortie "Q" est mise au niveau "L".

Si le compteur dépasse la capacité, la sortie "Err" est mise au niveau "H".

#### 4.4.3.4 Comptage / Décomptage avec présélection

##### Compt./décompt, présél.

[\_updnpr2]



Entrées:	– Set:	Présélection	format binaire
	> Up:	+1	format binaire
	> Dwn:	–1	format binaire
	= IC:	Valeur initiale	format entier
Sorties:	– Q:	pos/neg	format binaire
	= Cnt:	Contenue	format entier
	– Err:	Error	format binaire

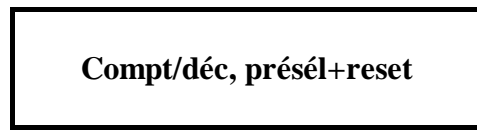
Un compteur est incrémenté ou décrémenté chaque fois qu'un flanc montant est détecté sur l'entrée correspondante "Up" ou "Dwn". Le contenu du compteur peut être lu sur la sortie "Cnt".

L'entrée "Set" charge le compteur avec la valeur de l'entrée "IC".

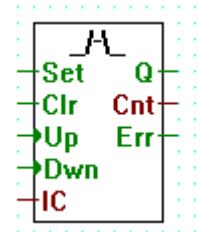
Si le contenu du compteur est  $> 0$ , la sortie "Q" est mise au niveau "H". Si le contenu est  $\leq 0$ , la sortie "Q" est mise au niveau "L".

Si le compteur dépasse la capacité (positive ou négative), la sortie "Err" est mise au niveau "H".

#### 4.4.3.5 Comptage/Décomptage avec présélection et mise à zéro



[\_updnsc]



Entrées:	– Set:	Présélection	format binaire
	– CLR:	Clear	format binaire
	> Up:	+1	format binaire
	> Dwn:	–1	format binaire
	= IC:	Valeur initiale	format entier
Sorties:	– Q:	pos/neg	format binaire
	= Cnt:	Contenue	format entier
	– Err:	Error	format binaire

Un compteur est incrémenté ou décrémente chaque fois qu'un flanc montant est détecté sur l'entrée correspondante "Up" ou "Dwn". Le contenu du compteur peut être lu sur la sortie "Cnt".

L'entrée "Set" charge le compteur avec la valeur de l'entrée "IC".

L'entrée "Clr" efface le contenu du compteur.

Si le contenu du compteur est  $> 0$ , la sortie "Q" est mise au niveau "H". Si le contenu est  $\leq 0$ , la sortie "Q" est mise au niveau "L".

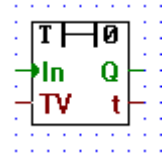
Si le compteur dépasse la capacité (positive ou négative), la sortie "Err" est mise au niveau "H".

**Notes personnelles :**

### 4.4.4 Temporiseurs

[sfuptime]

#### 4.4.4.1 Enclenchement retardé



[\_ondel]

Entrées:	> In: Départ	format binaire
	= TV: Valeur tempo	format entier
Sorties:	- Q: Sortie tempo	format binaire
	= t: Contenu tempo	format entier

Enclenchement retardé.

Cette fonction permet de retarder l'enclenchement de la sortie.

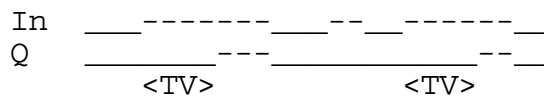
Dès l'apparition de "In", "Q" est mis au niveau "H" après l'écoulement du temps TV.

La mise à l'état bas de "In", entraîne la mise au niveau "L" de la sortie "Q".

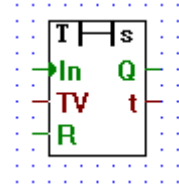
Si "In" passe rapidement de "H" à "L" puis "H" pendant l'écoulement de TV, ceci n'a aucun effet sur l'état de "Q".

La sortie "t" indique le contenu actuel de la temporisation.

Exemple:



**4.4.4.2 Enclenchement retardé avec mémorisation**



[\_stodel]

- |          |                    |                |
|----------|--------------------|----------------|
| Entrées: | > In: Départ       | format binaire |
|          | = TV: Valeur tempo | format entier  |
|          | - R: Reset         | format binaire |
| Sorties: | - Q: Sortie tempo  | format binaire |
|          | = t: Contenu tempo | format entier  |

Retard.

Cette fonction permet de retarder la propagation de l'entrée vers la sortie.

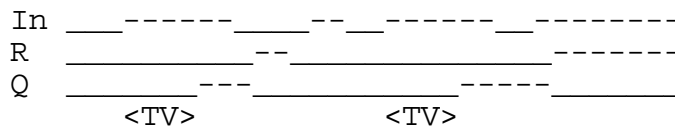
A l'apparition de l'entrée "In", la sortie "Q" est mise au niveau "H" après l'écoulement du temps TV .

Seule l'apparition de "R" (reset) peut remettre "Q" à l'état "L". Si "R" reste "Haut", la sortie reste "Bas" quel que soit l'état de "In".

Si "In" reste au niveau "H" pour un temps inférieur à TV, "Q" n'est pas modifié.

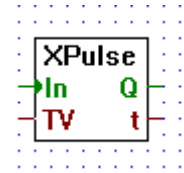
La sortie "t" indique le contenu actuel de la temporisation.

Exemple:





### 4.4.4.3 Impulsion unique



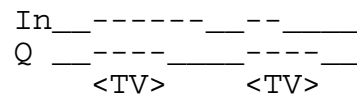
[\_xpulse]

Entrées:	> In: Départ	format binaire
	= TV: Valeur tempo	format entier
Sorties:	- Q: Sortie tempo	format binaire
	= t: Contenu tempo	format entier

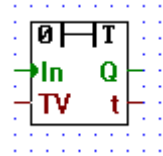
Cette fonction génère une impulsion à sa sortie sur apparition d'un front P en entrée. La largeur de cette impulsion (TV: temps correspondant à la durée de l'impulsion) est à définir.

La sortie "t" indique le contenu actuel de la temporisation.

Exemple:



**4.4.4.4 Déclenchement retardé**



[\_offdel]

Entrées: > In: Départ format binaire  
 = TV: Valeur tempo format entier

Sorties: - Q: Sortie tempo format binaire  
 = t: Contenu tempo format entier

Déclenchement retardé. Cette fonction permet de retarder le déclenchement de la sortie.

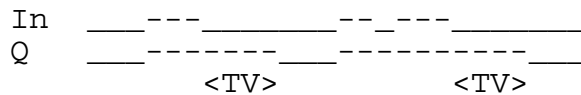
Dès l'apparition de "In", "Q" est mis au niveau "H".

La mise à l'état bas de "In", entraîne la mise au niveau "L" de "Q" après l'écoulement du temps TV.

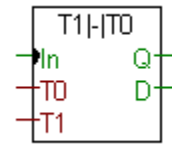
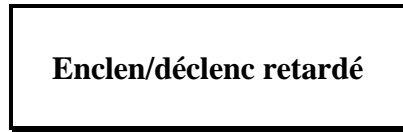
Si "In" passe rapidement de "H" à "L" puis "H" pendant l'écoulement de TV, ceci n'aura aucun effet sur l'état de "Q".

La sortie "t" indique le contenu actuel de la temporisation.

Exemple:



4.4.4.5 Enclenchement et déclenchement retardé



[\_onoffd2]

Entrées:	> In: Départ = T0: Valeur tempo 0 = T1: Valeur tempo 1	format binaire format entier format entier
Sorties:	– Q: Sortie tempo – D: Retardement actif/inactif	format binaire format binaire

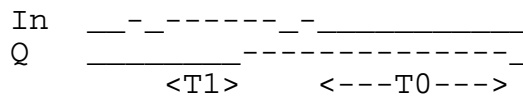
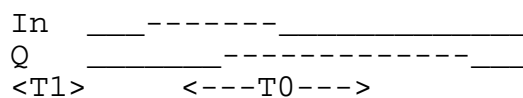
Enclenchement et déclenchement retardé. Cette fonction permet de retarder l'enclenchement et le déclenchement de la sortie.

A l'enclenchement de "In", "Q" est mis "H" après l'écoulement du temps T1.

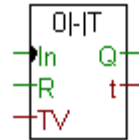
Au déclenchement de "In", "Q" est remis "L" après l'enclenchement du temps T0.

Sortie "D" = H: retardement actif,  
 Sorite "D" = L: retardement inactif.

Exemple:



**4.4.4.6 Déclenchement retardé avec reset**



[\_offdelr]

Entrées:            > In:  Départ                    format binaire  
                       - R:  Reset                    format binaire  
                       = TV: Valeur tempo            format entier

Sorties:            - Q:  Sortie tempo                format binaire  
                       = t:  Contenu tempo            format entier

Déclenchement retardé avec possibilité d'annulation du retardement.

Dès l'apparition de "In", "Q" est mis au niveau "H".

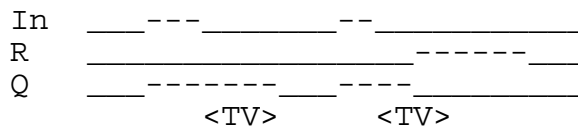
La mise à l'état bas de "In", entraîne la mise au niveau "L" de "Q" après l'écoulement du temps TV.

Si l'entrée "R" (Reset) = H, le temporisateur est remis à zéro et la sortie devient basse.

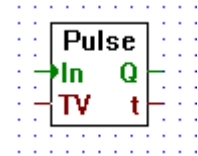
Si "In" passe rapidement de "H" à "L" puis "H" pendant l'écoulement de TV, ceci n'aura aucun effet sur l'état de "Q".

La sortie "t" indique le contenu actuel de la temporisation.

Exemple:



4.4.4.7 Impulsion



[\_pulse]

Entrées:           > In:  Départ                   format binaire  
                       = TV:  Valeur tempo       format entier

Sorties:           – Q:  Sortie tempo           format binaire  
                       = t:  Contenu tempo       format entier

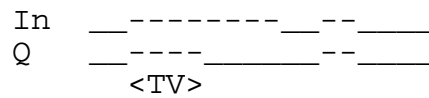
Générateur d'impulsion. Cette fonction permet de générer une impulsion dont on peut régler la largeur.

Dès l'apparition du signal de commande "In", la sortie "Q" est mis "H" en ayant une largeur d'impulsion égale au temps TV.

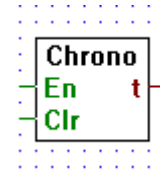
Si "In" retombe avant la fin de TV, l'impulsion est tronquée.

La sortie "t" indique le contenu actuel de la temporisation.

Exemple:



#### 4.4.4.8 Chronomètre



[\_chrono2]

Entrées:       – En: Enable           format binaire  
              – Clr: Clear           format binaire

Sortie:         = t: Contenu tempo   format entier

Cette fonction permet une incrémentation de la valeur du temps "t".

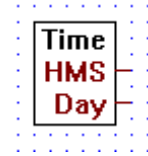
L'enclenchement de celle-ci se réalise par la présence d'un niveau "H" à "En". Si "En" est au niveau "L", le chrono arrête l'incrémentation. La remise à "H" de "En" le fait reprendre à la valeur d'arrêt.

Un niveau "H" sur "Clr" initialise le chronomètre. La valeur du temps est contenue dans la sortie "t".

L'incrément se fait une fois par seconde.

---

**4.4.4.9 Horloge du PCD**



[\_time]

Sorties:           = HMS:       Heures, Minutes, Seconds   format entier  
                       = Day:        Jour et Date                   format entier

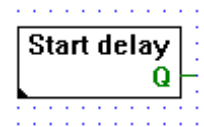
Cette fonction contient l'horloge interne propre au PCD qui est écrit dans deux registres de sortie.

Les sorties sont écrites de la manière suivante

HMS		Day	
<u>Contenu</u>	<u>No. décade</u>	<u>Contenu</u>	<u>No. décade</u>
0	9-6	0	9
Heures	5-4	Semaine	8-7
Minutes	3-2	Jour de sm.	6
Secondes	1-0	Année	5-4
		Mois	3-2
		Date	1-0

Voir également la description de l'instruction du PCD: "RTIME"

**4.4.4.10 Start retardé**



[\_stdel]

Sortie:           – Q:   format binaire

A l'enclenchement du PCD, la sortie "Q" devient = H après le délai défini dans la fenêtre d'ajustage.

Exemple:

```
PCD Start up  _____-----
Q             _____-----
                < Delay >
```

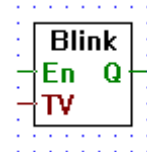




## 4.4.5 Clignoteurs

[sfupblin]

### 4.4.5.1 Clignoteur 1 temps



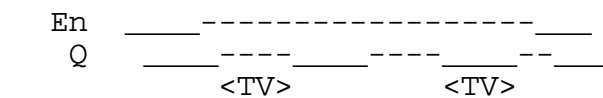
[\_blink1]

Entrées:       – En: Enable                   format binaire  
              = TV  Temporisation       format entier

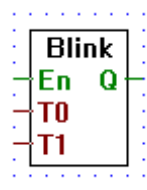
Sortie:        – Sortie clignotante       format binaire

Cette fonction permet d'obtenir le clignotement de la sortie "Q" suivant le temps TV aussi longtemps que l'entrée "En" est haut.

Si "En" est bas, "Q" est mise bas.



### 4.4.5.2 Clignoteur 2 temps



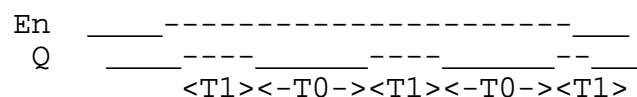
[\_blink2]

Entrées:       – En: Enable                   format binaire  
              = T0  Temporisation 0       format entier  
              = T1  Temporisation 1       format entier

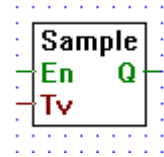
Sortie:        – Sortie clignotante       format binaire

Cette fonction permet d'obtenir le clignotement de la sortie "Q" suivant le temps T1 pour l'état haut et T0 pour l'état bas.

Q est mise bas si l'entrée "En" est bas.



**4.4.5.3 Signal d'échantillonnage**

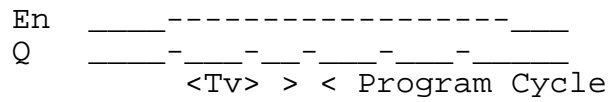


[\_blinksamp]

Entrées:      – En: Enable                   format binaire  
               = TV Temporisation       format entier

Sortie:       – Sortie d'échantillonnage   format binaire

Signal d'échantillonnage



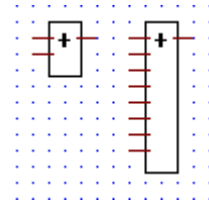
## 4.4.6 Arithmétique en format entier

**[sfupinte]**

### 4.4.6.1 Addition



[\_iadd]



Entrées/Sorties: = toutes → format entier

Cette fonction permet d'additionner les entrées entières et transfert dans la sortie le résultat.

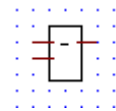
La boite est étirable de 2 à 8 entrées.

---

### 4.4.6.2 Soustraction



[\_isub]



Entrées/Sorties: = toutes → format entier

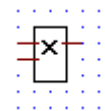
Cette fonction réalise la soustraction de la 2ème de la 1ère entrée entière.

---

### 4.4.6.3 Multiplication



[\_imul]

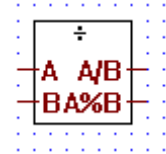


Entrées/Sorties: = toutes → format entier

Cette fonction permet d'obtenir en sortie le résultat de la multiplication des deux entrées entières.

---

#### 4.4.6.4 Division



[\_div]

Entrées/Sorties: = toutes → format entier

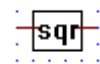
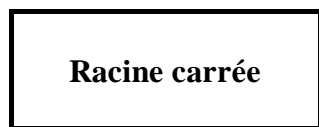
Cette fonction réalise la division des nombres entiers A et B.

La partie entière de la division de A par B est donnée dans A/B.

Le reste (exprimé en fonction des chiffres derrière la virgule) est sorti sur A%B.

---

#### 4.4.6.5 Racine carrée



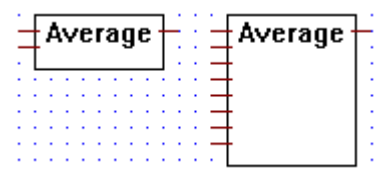
[\_isqr]

Entrées/Sorties: = toutes → format entier

Cette fonction donne la racine carrée de l'entrée entière.

---

#### 4.4.6.6 Moyenne



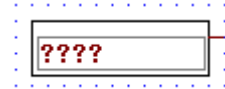
[\_iaverage]

Entrées/Sorties: = toutes → format entier

Cette fonction permet de transférer dans la sortie la moyenne des entrées entières.

La boîte est étirable de 2 à 8 entrées.

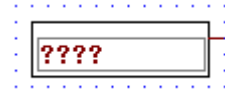
---

**4.4.6.7 Constante**

[\_iconst]

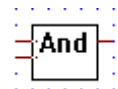
Entrées/Sorties: = toutes → format entier

Cette fonction délivre à sa sortie un nombre entier défini dans la fenêtre.

**4.4.6.8 Absolue**

[\_iabs]

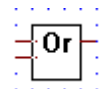
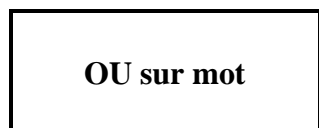
La sortie de cette fonction donne la valeur absolue (conversion de la valeur entière en positif si négatif) de l'entrée entière.

**4.4.6.9 ET sur mot**

[\_iand]

Entrées/Sorties: = toutes → format entier

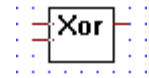
Cette fonction réalise la fonction ET logique de la 1ère et 2ème entrée entière.

**4.4.6.10 OU sur mot**

[\_ior]

Entrées/Sorties: = toutes → format entier

Cette fonction réalise le OU logique en sortie de la 1ère et 2ème entrée entière.

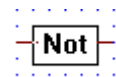
**4.4.6.11 OU EXCL. sur mot**

[\_ixor]

Entrées/Sorties: = toutes → format entier

Cette fonction réalise le OU logique en sortie de la 1ère et 2ème entrée entière.

---

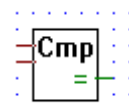
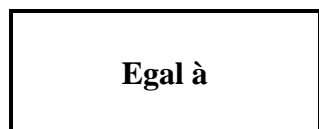
**4.4.6.12 Inverse**

[\_inot]

Entrées/Sorties: = toutes → format entier

Donne l'inverse de l'entrée (complément à 1).

---

**4.4.6.13 Egal à**

[\_icmpeq]

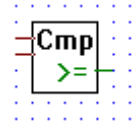
Entrées: = toutes → format entier

Sortie: – format binaire

Cette fonction réalise une comparaison des entrées entières.

La sortie binaire est mise à l'état "H" si les 2 entrées sont égales, dans le cas contraire, la sortie est mise à l'état "L".

---

**4.4.6.14 Plus grand ou égal à**

[\_icmpge]

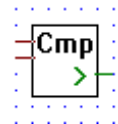
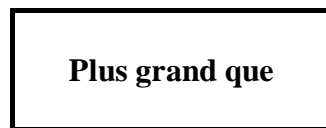
Entrées: = toutes → format entier

Sortie: – format binaire

Cette fonction réalise une comparaison des entrées entières.

La sortie binaire est mise à l'état "H" si l'entrée du haut est plus grande ou égale à celle du bas, dans le cas contraire, la sortie est "L".

---

**4.4.6.15 Plus grand que**

[\_icmpgt]

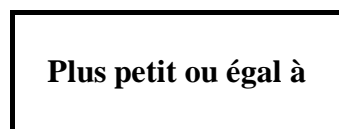
Entrées: = toutes → format entier

Sortie: – format binaire

Cette fonction réalise une comparaison des entrées entières.

La sortie binaire est "Haut" si l'entrée du haut est strictement plus grande que celle du bas, dans le cas contraire, la sortie est "L".

---

**4.4.6.16 Plus petit ou égal à**

[\_icmpse]

Entrées: = toutes → format entier

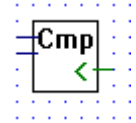
Sortie: – format binaire

Cette fonction réalise la comparaison des entrées entières.

La sortie binaire est mise à l'état "H" si l'entrée du haut est plus petite ou égale à celle du bas, dans le cas contraire, la sortie est à l'état "L".

---

#### 4.4.6.17 Plus petit que



[\_icmpst]

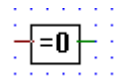
Entrées: = toutes → format entier

Sortie: – format binaire

Cette fonction réalise une comparaison des entrées entières.

La sortie binaire est mise à l'état "H" si l'entrée du haut est strictement plus petite que celle du bas, dans le cas contraire, la sortie est mise à l'état "L".

#### 4.4.6.18 Egal à zéro



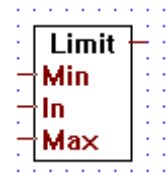
[\_iszero]

Entrées: = toutes → format entier

Sortie: – format binaire

Cette fonction permet à la sortie d'être mise à l'état "H" quand l'entrée entière est zéro, dans le cas contraire, elle est mise à l'état "L".

#### 4.4.6.19 Limite



[\_limit]

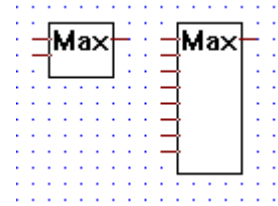
Entrées/Sorties: = toutes → format entier

Cette fonction permet de sortir la valeur de "In" si celle-ci est dans les limites "Max" et "Min". Dans le cas contraire, cette fonction aura en sortie la valeur entière minimum quand la valeur de "In" est inférieure à "Min" ou la valeur maximum quand la valeur de "In" est supérieure à "Max".



**4.4.6.20 Maximum**

[\_imax]



Entrées/Sorties: = toutes → format entier

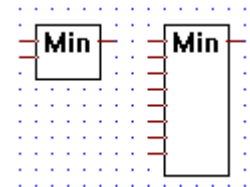
Cette fonction permet d'obtenir en sortie la valeur de la plus grande des entrées entières.

La boîte est étirable de 2 à 8 entrées.

---

**4.4.6.21 Minimum**

[\_imin]



Entrées/Sorties: = toutes → format entier

Cette fonction permet d'obtenir en sortie la valeur de la plus petite des entrées entières.

La boîte est étirable de 2 à 8 entrées.

---

**4.4.6.22 Mouve**

[\_imove]

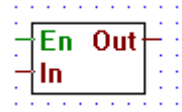


Entrées/Sorties: = toutes → format entier

Cette fonction permet le transfert de la valeur d'entrée entière vers la sortie. Elle est utilisée pour établir une liaison entre deux étiquettes d'entier.

---

#### 4.4.6.23 Mouve valide



[\_imovee]

Entrées: – En: Enable (format binaire)  
= In: Input (format entier)

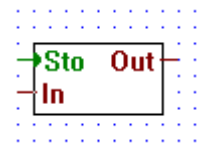
Sortie: = Out Output (format entier)

Cette fonction permet le transfert de la valeur d'entrée entière vers la sortie quand la ligne de validation est au niveau "H".

Si l'entrée "En" = L, la valeur à la sortie est zéro.

---

#### 4.4.6.24 Mouve sauve



[\_imoves]

Entrées: > Sto: Store (format binaire)  
= In: Input (format entier)

Sortie: = Out: Output (format entier)

Cette fonction permet de sauvegarder l'entrée dans un registre interne quand "Sto" passe de l'état "L" à l'état "H". La sortie contient toujours le contenu de ce registre interne.

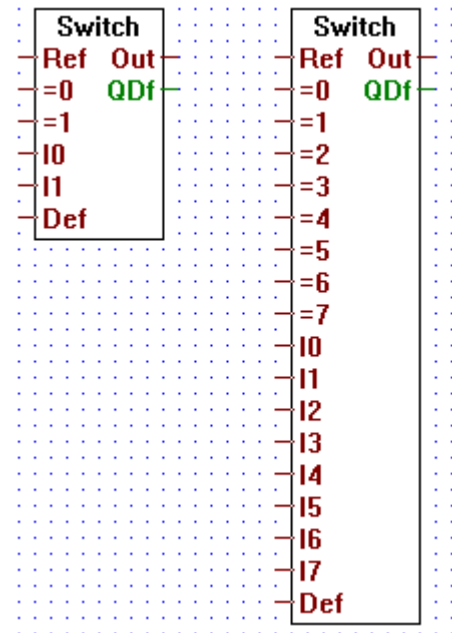
---

## 4.4.6.25

## Commute



[\_iswitch]



Entrées: = toutes → format entier

Sorties: = Out: format entier  
– QDf format binaire

Quand l'entrée "Ref" est égale à l'une des entrées "=n", la valeur de "In" associée est transférée dans la sortie. Quand aucune entrée "=n" n'est égale à "Ref", alors l'entrée "Def" est copiée dans "Out" et la sortie "Qdf" est mise "Haut".

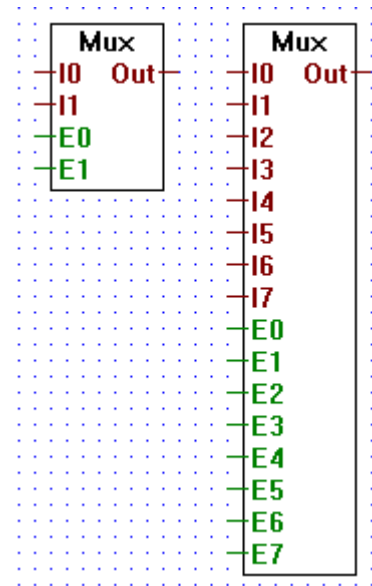
La boîte est étirable de I1 à I7.

L'entrée I0 est prioritaire par rapport à I7. Quand plus d'une entrée "=n" est égale à "Ref", c'est l'entrée "In" de plus haute priorité qui est transférée.

#### 4.4.6.26 Multiplexage avec présélection binaire



[\_imux]



Entrées: = I0:  
           |       format entier  
           = I7:  
           - E0:  
           |       format binaire  
           - E7:

Sortie: = Out: format entier

Cette fonction sort le contenu de "In" quand l'entrée correspondante "En" est au niveau "H".

La boîte est étirable de 2 à 8 entrées.

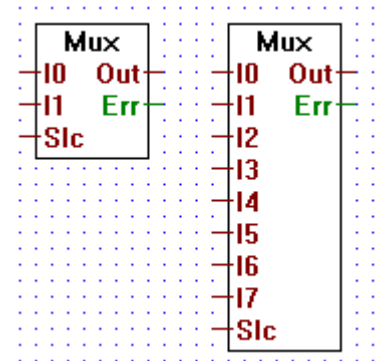
Si toutes les validations "En" sont "L", la valeur 0 est sortie.

L'entrée de validation E0 a la plus haute priorité et E7 la plus basse. Si plus d'une entrée sont actives en même temps, la valeur associée à celle de plus haute priorité est sortie.

#### 4.4.6.27 Multiplexage avec présélection numérique



[\_imux2]



Entrées: = I0:  
           |     format entier  
           = I7:  
           = Slc  format entier

Sortie: = Out: format entier  
        - Err: format binaire

Cette fonction transfère l'entrée "In" dans la sortie "Out". L'adresse de l'entrée est la valeur de "Slc".

(par ex.: si "Slc" = 0 transfère de I0, si "Slc" = 5 transfère de I5...).

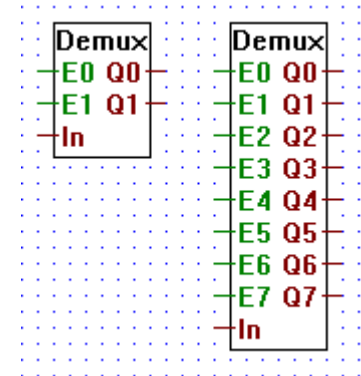
La boîte est étirable de 2 à 8 entrées "In".

Si "Slc" est en dehors de l'échelle (p.ex.: "Slc" < 0 ou "Slc" > n), "Out" vaut 0 et "Err" est mis "Haut".

#### 4.4.6.28 Démultiplexage avec présélection binaire



[\_idemux]



Entrées: – E0:  
           |       format binaire  
 – E7:  
 = In:   format entier

Sorties: = Q0:  
           |       format entier  
 = Q7:

Cette fonction permet le transfert de la valeur de "In" dans la sortie (Q0..Q7) quand la validation correspondantes (E0..E7) est active.

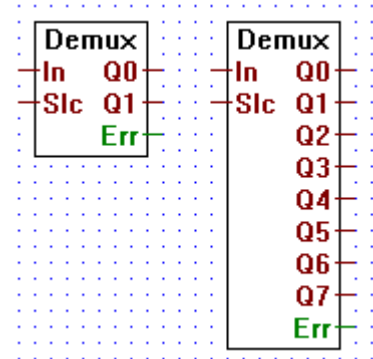
La boîte est étirable de 2 à 8 entrées.

Quand une ligne de validation est basse (E0..E7) la sortie correspondante (Q0...Q7) est mise à .

#### 4.4.6.29 Démultiplexage avec présélection numérique



[\_idemux2]



Entrées: = In: format entier  
 = Slc: format entier

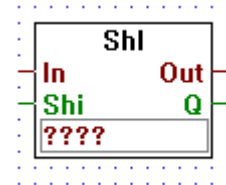
Sorties: = Q0 format entier  
 |  
 = Q7 format entier  
 – Err: format binaire

Cette fonction permet le transfert de la valeur de "In" dans la sortie (Q0..Qn) sélectionnée par "Slc". Par exemple, si "Slc" = 1 l'entrée "In" est transférée dans "Q1". Les sorties non sélectionnées sont mises à 0.

La boîte est étirable de 2 à 8 sorties.

"Err" devient haut quand "Slc" est en dehors de l'échelle, est négatif ou plus grand que "n".

#### 4.4.6.30 Décalage à gauche



[\_ishftl2]

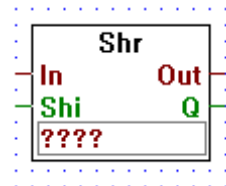
Entrées: = In: Input (format entier)  
 – Shi: Shift (format binaire)

Sorties: = Out: Output (format entier)  
 – Q: Output (format binaire)

La sortie "Out" contient la valeur entière de "In" après décalage à gauche du nombre de bits spécifié dans la fenêtre de paramétrage. L'état logique de "Q" correspond à l'état du dernier bit qui a été décalé. L'entrée "Shi" est l'état logique que doit prendre le bit 0.

---

#### 4.4.6.31 Décalage à droite



[\_ishftr2]

Entrées: = In: Input (format entier)  
 – Shi: Shift (format binaire)

Sorties: = Out: Output (format entier)  
 – Q: Output (format binaire)

La sortie "Out" contient la valeur entière de "In" après décalage à droite du nombre de bits spécifié dans la fenêtre de paramétrage. L'état logique de "Q" correspond à l'état du dernier bit qui a été décalé. L'entrée "Shi" est l'état logique que doit prendre le bit 31.

---



#### 4.4.6.32 Rotation à gauche

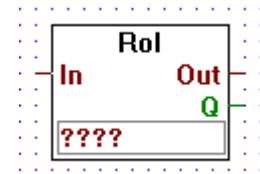


[\_irotlf2]

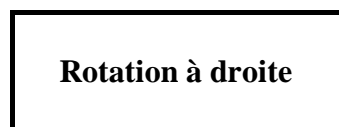
Entrées: = In: Input (format entier)

Sorties: = Out: Output (format entier)  
– Q: Output (format binaire)

La sortie "Out" contient la valeur entière de "In" après rotation à droite du nombre de bits spécifié dans le champ d'entrée. L'état logique de "Q" correspond à l'état du dernier bit qui a tourné.



#### 4.4.6.33 Rotation à droite

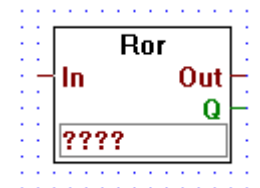


[\_irotri2]

Entrées: = In: Input (format entier)

Sorties: = Out: Output (format entier)  
– Q: Output (format binaire)

La sortie "Out" contient la valeur entière de "In" après rotation à droite du nombre de bits spécifié dans le champ d'entrée. L'état logique de "Q" correspond à l'état du dernier bit qui a tourné.



**4.4.6.34 Copie registre**

[\_iregin]

Entrées/Sorties: = toutes → format entier

Cette fonction permet de sortir le contenu du registre #.

Par ex.: si # = 5, alors la valeur du registre 5 est sortie.

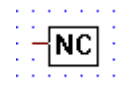
**4.4.6.35 Copie T/C**

[\_itcind]

Entrées/Sorties: = toutes → format entier

Cette fonction permet de sortir le contenu du compteur/tempo #.

Par ex.: si # = 5, alors la valeur du compteur/tempo 5 est sortie.

**4.4.6.36 Non connecté**

[\_inotcn]

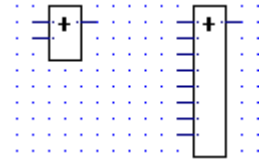
"Non connecté" est une fonction qui termine une sortie entière non raccordée.

## 4.4.7 Arithmétique en format flottant

**[sfupfloa]**

### 4.4.7.1

#### Addition



[\_fadd2]

Entrées/Sorties: – toutes → format flottant

Cette fonction réalise l'addition des entrées flottantes dont le résultat est obtenu en sortie.

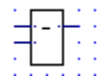
Cette opération va se faire par traitement des mots binaires en fonction des bits de même poids binaire.

La boîte est étirable de 2 à 8 entrées.

---

### 4.4.7.2

#### Soustraction



[\_fsub]

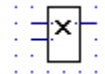
Entrées/Sorties: – toutes → format flottant

Cette fonction donne à la sortie le résultat de la soustraction de l'entrée supérieure et l'entrée inférieure.

---

### 4.4.7.3

#### Multiplication



[\_fmul]

Entrées/Sorties: – toutes → format flottant

Cette fonction donne en sortie le résultat de la multiplication des deux entrées.

---

## 4.4.7.4

**Division**

[\_fdiv]

Entrées/Sorties: – toutes → format flottant

Cette fonction donne le résultat de la division de la 1ère entrée par la seconde.

---

## 4.4.7.5

**Racine carrée**

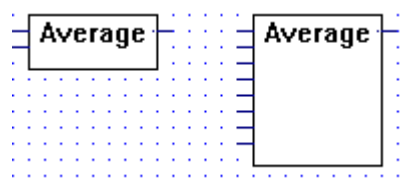
[\_fsqr]

Entrées/Sorties: – toutes → format flottant

Cette fonction donne la racine carrée de l'entrée flottante à la sortie.

---

## 4.4.7.6

**Moyenne**

[\_faverage]

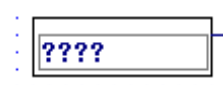
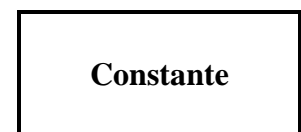
Entrées/Sorties: – toutes → format flottant

La sortie de cette fonction donne la moyenne des entrées flottantes.

La boite est étirable de 2 à 8 entrées.

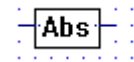
---

## 4.4.7.7

**Constante**

[\_fconst]

Cette fonction donne la valeur de la constante flottante dans la fenêtre de paramétrage à la sortie

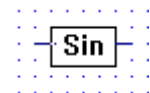
**4.4.7.8****Absolu**

[\_fabs ]

Entrées/Sorties: – toutes → format flottant

La sortie de cette fonction donne la valeur absolue (conversion de la valeur flottante en positif si négatif) de l'entrée flottante.

---

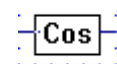
**4.4.7.9****Sinus**

[\_fsin]

Entrées/Sorties: – toutes → format flottant

Cette fonction donne le sinus de l'angle entré en radian.

---

**4.4.7.10****Cosinus**

[\_fcos]

Entrées/Sorties: – toutes → format flottant

Cette fonction donne le cosinus de l'angle (valeur flottante) entré dont l'unité sera exprimée en radian.

---

**4.4.7.11 ARC tangente**

**ARC tangente**

**Atan**

[\_fatan]

Entrées/Sorties: – toutes → format flottant

La sortie de cette fonction donne l'arc-tangente de l'angle (en valeur flottante) entré dont l'unité sera exprimée en radian.

---

**4.4.7.12 Exponentiel**

**Exponentiel**

**Exp**

[\_fexp]

Entrées/Sorties: – toutes → format flottant

Cette fonction donne en sortie la valeur de l'exponentiel de x ( $e^x$ ) ou x est la valeur en puissance de l'entrée .

---

**4.4.7.13 Log népérien**

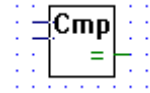
**Log népérien**

**Ln**

[\_fln]

Entrées/Sorties: – toutes → format flottant

Cette fonction donne le logarithme népérien de l'entrée.

**4.4.7.14**      **Egal à**

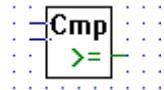
[\_fcmpeq]

Entrées: – toutes →    format flottant

Sortie: –                    format binaire

La sortie de cette fonction est mise "H" lorsque les deux entrées sont égales, sinon elle est "L".

---

**4.4.7.15**      **Plus grand ou égal à**

[\_fcmpge]

Entrées: – toutes →    format flottant

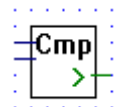
Sortie: –                    format binaire

Cette fonction compare deux nombres flottants:

Si les deux entrées sont supérieures ou égales, l'état logique binaire de la sortie est mise "H".

Si elles sont inférieures et différentes, l'état logique binaire de la sortie est mise "L".

---

**4.4.7.16**      **Plus grand que**

[\_fcmpgt]

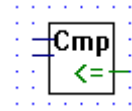
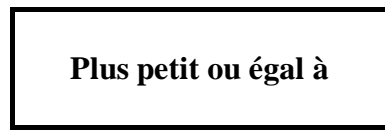
Entrées: – toutes →    format flottant

Sortie: –                    format binaire

Cette fonction compare deux nombres flottants:

La sortie binaire est à "H" si l'entrée du haut est strictement plus grande que celle du bas, sinon la sortie est à "L".

---

**4.4.7.17 Plus petit ou égal à**

[\_fcmpse]

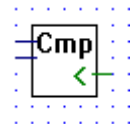
Entrées: – toutes → format flottant

Sortie: – format binaire

Cette fonction compare deux nombres flottants:

La sortie binaire est à "H" si l'entrée du haut est plus petite ou égale à celle du bas, sinon la sortie est à "L".

---

**4.4.7.18 Plus petit que**

[\_fcmpst]

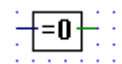
Entrées: – toutes → format flottant

Sortie: – format binaire

Cette fonction compare deux nombres flottants:

La sortie binaire est à "H" si l'entrée du haut est strictement plus petite que celle du bas, sinon la sortie est à "L".

---

**4.4.7.19 Egal à zéro**

[\_fiszero]

Entrée: – format flottant

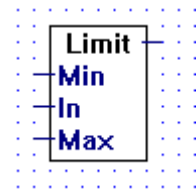
Sortie: – format binaire

La sortie de cette fonction est à "H" si l'entrée flottante est égale à zéro, sinon elle garde un état "L" lorsqu'il n'y a pas de présence du nombre flottant (0.0).

---



## 4.4.7.20

**Limite**

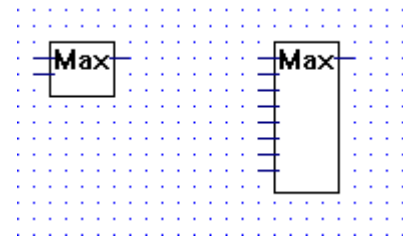
[\_flimit]

Entrées/Sorties: – toutes → format flottant

Cette fonction sort la valeur flottante de "In" si celle-ci est dans les limites "Max" et "Min".

---

## 4.4.7.21

**Maximum**

[\_fmax2]

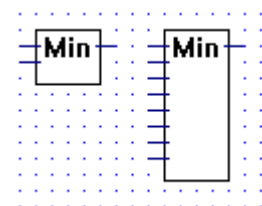
Entrées/Sorties: – toutes → format flottant

Cette fonction donne la valeur de la plus grande des entrées.

La boîte est étirable de 2 à 8 entrées.

---

## 4.4.7.22

**Minimum**

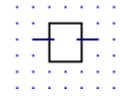
[\_fmin2]

Entrées/Sorties: – toutes → format flottant

Cette fonction donne la valeur de la plus petite des entrées.

La boîte est étirable de 2 à 8 entrées.

---

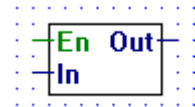
**4.4.7.23 Mouve**

[\_fmove]

Entrées/Sorties: – toutes → format flottant

La sortie de cette fonction est égale à l'entrée. Elle est utilisée pour établir une liaison entre deux étiquettes de flottants.

---

**4.4.7.24 Mouve valide**

[\_fmovee]

Entrées: – En: Enable (format binaire)  
– In: Input (format flottant)

Sortie: – Out: Output (format flottant)

La sortie de cette fonction est égale à l'entrée quand la ligne de validation est "Haut". Si la ligne de validation est "Bas", la sortie est 0.0.

---

## 4.4.7.25

**Mouve sauve**

[\_fmoves]

Entrées: > Sto: Store (format binaire)  
– In: Input (format flottant)

Sortie: – Out: Output (format flottant)

Cette fonction sauvegarde l'entrée dans un registre interne quand "Sto" passe de "Bas" à "Haut". La sortie contient toujours le contenu de ce registre interne.

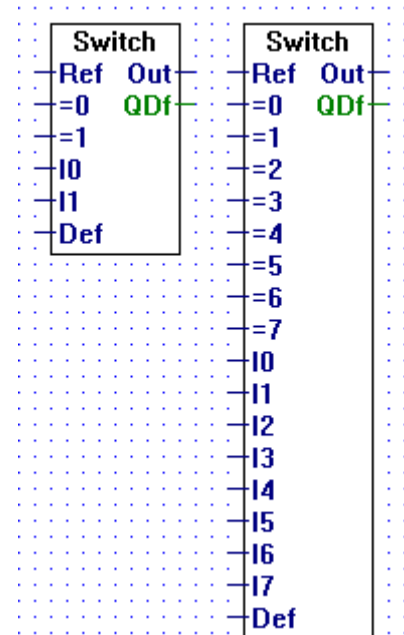
---

## 4.4.7.26

## Commute



[\_fswitch]



Entrées: – →            format flottant

Sorties: – Out:            format flottant  
          – QDf:            format binaire

Quand l'entrée "Ref" est égale à l'une des entrées "=n", la valeur de "In" associée est transférée dans la sortie. Quand aucune entrée "=n" n'est égale à "Ref", alors l'entrée "Def" est copiée dans "Out" et la sortie "Qdf" est mise "Haut".

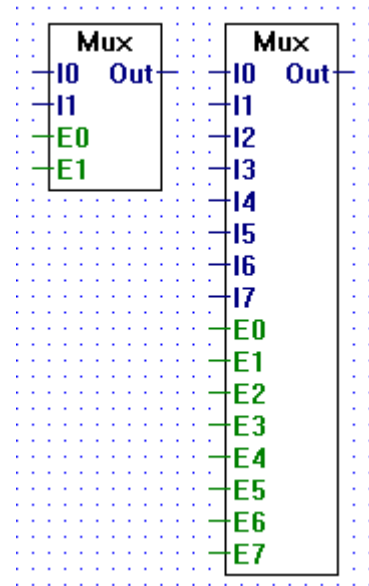
La boîte est étirable de I1 à I7.

L'entrée I0 est prioritaire par rapport à I7. Quand plus d'une entrée "=n" est égale à "Ref", c'est l'entrée "In" de plus haute priorité qui est transférée.

#### 4.4.7.27 Multiplexage avec présélection binaire



[\_fmux]



Entrées: – I0:  
           |       format flottant  
 – I7:  
 – E0:  
           |       format binaire  
 – E7:

Sortie: – Out: format flottant

Cette fonction transfère l'entrée (I0..I7) dans la sortie (Q0..Q7) quand la validation correspondante (E0..E7) est "Haut".

La boîte est étirable de 2 à 8 entrées.

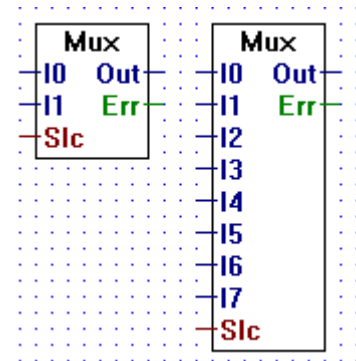
Si toutes les validation "En" sont "Bas", alors la sortie vaut 0.0.

L'entrée E0 est la plus haute priorité et E7 la plus basse: En conséquence, si plus d'une ligne de validation est "Haut", c'est l'entree associée a la plus haute priorité qui est transférée dans la sortie.

#### 4.4.7.28 Multiplexage avec présélection numérique



[\_fmux2]



Entrées: – I0:  
           |       format flottant  
 – I7:  
 = Slc: format entier

Sorties: – Out: format flottant  
 – Err: format binaire

Cette fonction transfère l'entrée "In" dans la sortie "Out". L'adresse de l'entrée est la valeur de "Slc".

(par ex.: si "Slc" = 0 transfère de I0, si "Slc" = 5 transfère de I5...).

La boîte est étirable de 2 à 8 entrées "In".

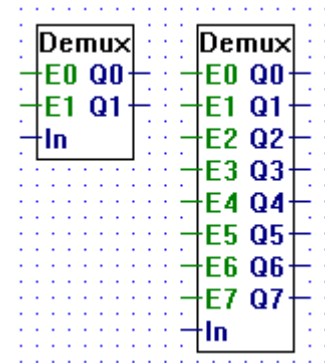
Si "Slc" est en dehors de l'échelle (p.ex.: "Slc" < 0 ou "Slc" > n), "Out" vaut 0.0 et "Err" est mis "Haut".

## 4.4.7.29

## Démultiplexage avec présélection binaire



[\_fdemux]



Entrées: – E0:  
           |       format binaire  
           – E7:  
           = In:   format flottant

Sorties: – Q0:  
           |       format flottant  
           – Q7:

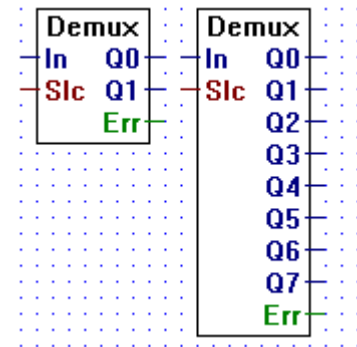
Cette fonction transfère la valeur de l'entrée "In" dans la sortie "Out" quand l'une des validations (E0..E7) associées est "Haut".

---

#### 4.4.7.30 Démultiplexage avec présélection numérique



[\_fdemux2]



Entrées: – In: format flottant  
= Slc: format entier

Sorties: – Q0:  
| format flottant  
– Q7:  
– Err: format binaire

Cette fonction transfère la valeur de "In" dans la sortie (Q0..Q7) sélectionnée par "Slc". Par exemple, si "Slc" = 1, l'entrée "In" est transférée dans "Q1". Les sorties non sélectionnées sont mises à 0.0.

La boîte est étirable de 2 à 8 sorties.

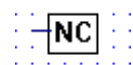
"Err" devient haut quand "Slc" est en dehors de l'échelle, est négatif ou plus grand que "n".

---

#### 4.4.7.31 Non connecté



[\_Fnotcn]



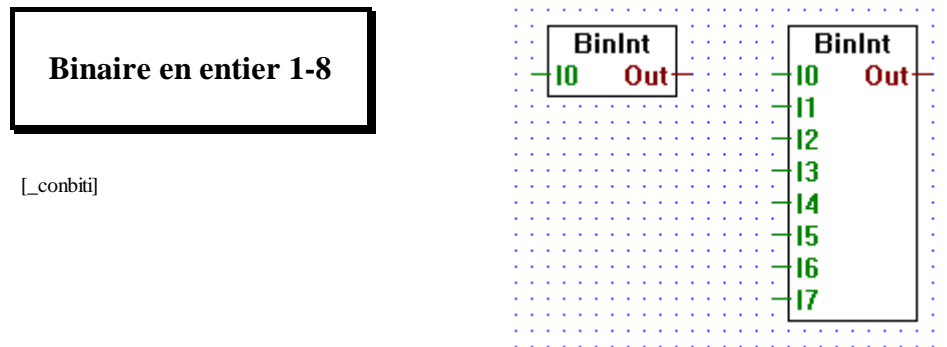
La boîte "Non connecté" termine une ligne flottante non raccordée.

---



## 4.4.8 Convertisseurs (binaire - entier - flottant) [sfupconv]

### 4.4.8.1 Binaire en entier à partir de 1 à 8 I/O/F



Entrées:       – I0  
                   |       format binaire  
                   – I7

Sortie:         = Out format entier

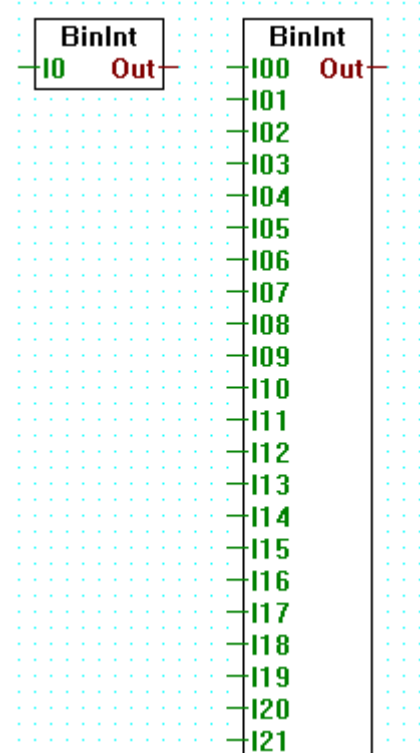
Cette fonction permet le transfert des entrées I0 à I7 dans les bits 0 à 7 de la sortie "Out". Tous les autres bits de la sortie sont effacés.

La boîte est étirable de 1 à 8 entrées.

**4.4.8.2 Binaire en entier à partir de 1 à 24 I/O/F**



[\_conbitim]



Entrées:       – I0:  
                  |       format binaire  
                  – I23

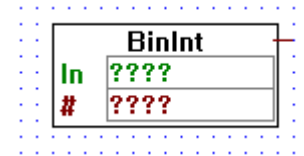
Output:         = Out format entier

Cette fonction permet le transfert des entrées I0 à I23 dans les bits 0 à 23 de la sortie Out". Tous les autres bits de la sortie sont effacés.

La boîte est étirable de 1 à 24 entrées.

Pour plus que 24 bits veuillez utiliser la fonction "Binaires en entier rapide".

#### 4.4.8.3 Binaire en entier rapide (format PCD)



[\_conbitiq]

Sortie: = format entier

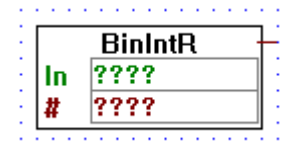
Cette fonction permet le transfert d'un groupe de bits dans un entier.

Le 1er paramètre indique l'adresse source (entrée, sorties ou variable) du groupe. Le second précise le nombre de bits à transférer.

L'adresse de poids faible du groupe sera transférée dans le bit de poids faible (LSB) de la sortie de la boite. Ceci est contraire au système PCA.

---

#### 4.4.8.4 Binaire en entier inversé rapide (format PCA)



[\_conbitiqr]

Sortie: = format entier

Cette fonction permet le transfert d'un groupe de bits dans un entier.

Le 1er paramètre indique l'adresse source (entrée, sorties ou variable) du groupe, le second précise le nombre de bits à transférer.

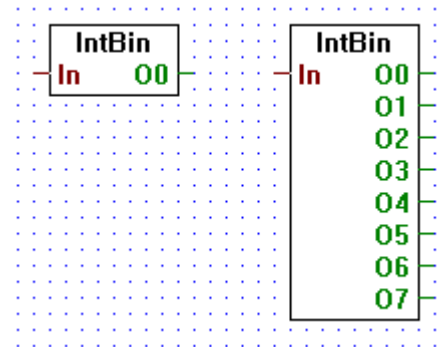
L'adresse de poids fort du groupe sera transférée dans le bit de poids faible (LSB) de la sortie de la boite. Ceci est conforme au système PCA.

---

**4.4.8.5 Entier en binaire sur 1 à 8 O/F**



[\_conbito]



Entrée: = In: format entier

Sorties: – O0:  
          | format binaire  
          – O7

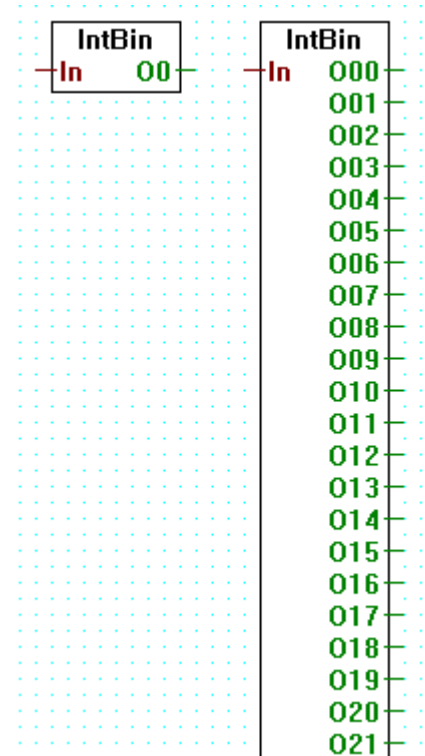
Cette fonction permet le transfert des bits 0..7 d'un entier sur les sorties binaires O0 à O7.

La boite est étirable de 1 à 8 bits.

**4.4.8.6 Entier en binaire sur 1 à 24 O/F**



[\_conbitom]



Entrée: = In: format entier

Sorties: – OO  
format binaire  
– 023

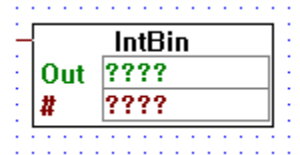
Cette fonction permet le transfert des bits 0..23 d'un entier sur les sorties binaires OO à 23.

La boîte est étirable de 1 à 24 bits.

Pour plus que 24 bits veuillez utiliser la fonction "Entier en binaire rapide".

#### 4.4.8.7 Entier en binaire rapide (format PCD)

**Ent en binaire rapide**



[\_conbitoq]

Entrée: = format entier

Cette fonction permet le transfert d'un entier dans un groupe de bits.

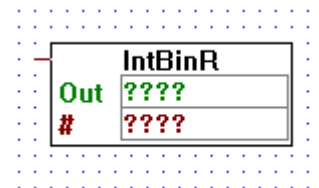
Le 1er paramètre indique l'adresse de destination (sortie ou variable) du groupe. Le second précise le nombre de bits à transférer.

Le bit de poids faible (LSB) de l'entier sera transféré dans l'adresse de poids faible du groupe. Ceci est contraire au système PCA.

---

#### 4.4.8.8 Entier en binaire inversé rapide (format PCA)

**Ent en bin. inv. rapide**



[\_conbitoqr]

Entrée: = format entier

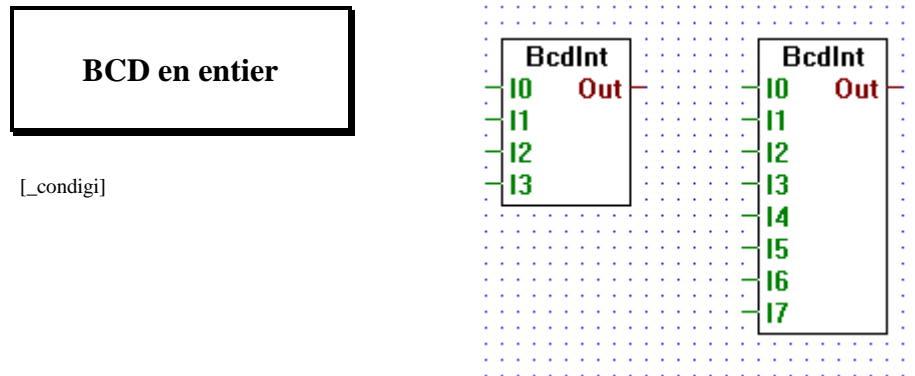
Cette fonction permet le transfert d'un entier dans un groupe de bits.

Le 1er paramètre indique l'adresse de destination (sortie ou variable) du groupe. Le second précise le nombre de bits de l'entier à transférer.

Le bit de poids le plus fort (MSB) sera transféré dans l'adresse de poids faible du groupe. Ceci est conforme au système PCA.

---

#### 4.4.8.9 BCD en entier à partir de 4 ou 8 I/O/F



Entrées: – I0

          |       format binaire  
 – I7

Sortie:       = Out format entier

Cette fonction permet la conversion de 1 à 2 décades BCD en entier positif dans la sortie "Out". Il est étirable de 4 à 8 entrées.

L'entrée I0 est l'entrée de poids faible ( $2^0$ ).

L'entrée I7 de poids fort ( $2^3 \cdot 10$ ).

---

#### 4.4.8.10 BCD en entier rapide (format PCD)



[\_condigi]

Sortie:       =       format entier

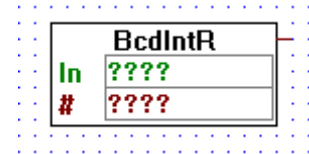
Cette fonction permet le transfert d'un groupe d'entrées codées BCD dans une sortie de la boîte en entier.

Le 1er paramètre est l'adresse de base du groupe d'entrées et le second le nombre de décade devant être lues.

L'entrée de poids faible est le bit unitaire de la décade des unités: il sera transféré dans le bit de poids faible de l'entier (LSB) ce qui est contraire à PCA.

---

#### 4.4.8.11 BCD en entier inversé rapide (format PCA)



[\_condigiqr]

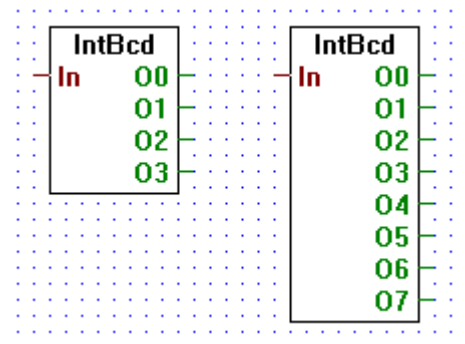
Sortie: = format entier

Cette fonction permet le transfert d'un groupe d'entrées codées BCD dans une sortie de la boîte en entier.

Le 1er paramètre est l'adresse de base du groupe d'entrées et le second le nombre de décade devant être lues.

L'entrée de poids fort est le bit unitaire de la décade des unités: il sera transféré dans le bit de poids faible de l'entier (LSB) ce qui est conforme à PCA.

#### 4.4.8.12 Entier en BCD sur 4 ou 8 O/F



[\_condigo]

Entrée: = In: format entier

Sorties: – O0:  
           | format binaire  
           – O7:

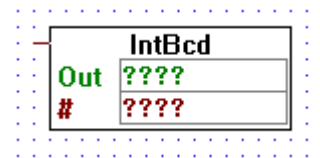
Cette fonction permet la conversion d'un entier à l'entrée "In" en 4 à 8 bits BCD. Il est étirable de 4 à 8 sorties.

La sortie O0 est la sortie de poids faible ( $2^0$ ).

La sortie O7 de poids fort ( $2^3 * 10$ ).



#### 4.4.8.13 Entier en BCD rapide (format PCD)



[\_condigoq]

Entrée: =        format entier

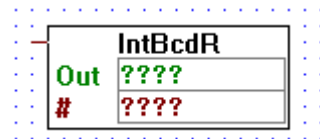
Cette fonction transcode un entier en BCD et positionne un groupe de sorties ou de variables.

Le 1er paramètre est l'adresse de base du groupe de sorties et le second le nombre de décade devant être écrites.

La sortie de poids faible est le bit unitaire de la décade des unités: elle provient du bit de poids faible de l'entier (LSB) ce qui est contraire à PCA.

---

#### 4.4.8.14 Entier en BCD inversé rapide (format PCA)



[\_condigoqr]

Entrée:        =        format entier

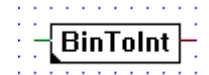
Cette fonction permet de transcoder un entier en BCD et le positionner en un groupe de sorties ou de variables.

Le 1er paramètre est l'adresse de base du groupe de sorties et le second le nombre de décade devant être écrites.

La sortie de poids fort est le bit unitaire de la décade des unités: elle provient du bit de poids faible de l'entier (LSB) ce qui est conforme à PCA.

---

#### 4.4.8.15 1-bit en entier avec shift



[\_conbinto]

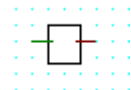
Entrée: – format binaire  
Sortie: = format entier

Cette fonction copie l'état logique de l'entrée binaire dans le bit désigné du registre de l'entier. Tous les autres bits sont mis à zéro.

Le bit choisi est spécifié par "l'adresse du bit" de la boîte de dialogue. (La boîte de dialogue est obtenu par deux clic de la souris sur la boîte réalisant cette fonction).



#### 4.4.8.16 1-bit en entier LSB

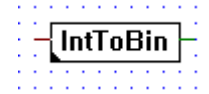


[\_conbimove]

Entrée: – format binaire  
Sortie: = format entier

Cette fonction copie l'état logique binaire de l'entrée en entier: si l'entrée est "Bas", l'entier vaut 0, si l'entrée est "Haut", l'entier vaut 1.

#### 4.4.8.17 Entier en 1 bit avec shift



[\_conitobin]

Entrée: = format entier  
Sortie: - format binaire

Cette fonction copie l'état du bit de l'entier désigné dans la sortie.

Le bit choisi est spécifié par "l'adresse du bit" de la boîte de dialogue.



#### 4.4.8.18 Entier LSB en 1-bit

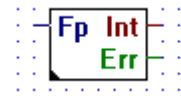


[\_conibmove]

Entrée: = format entier  
Sortie: - format binaire

Cette fonction positionne à l'état logique "Bas" d'une sortie si l'entier est pair (la fonction considère le bit du poids faible), sinon elle positionne cette sortie à l'état logique "Haut" si l'entier est impair (la fonction considère le bit du poids faible)

#### 4.4.8.19 Format flottant en format entier



[\_confip2]

Entrée: – Fp: format flottant

Sorties: = Int: format entier  
– Err: format binaire

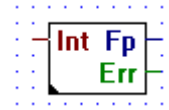
Cette fonction donne la représentation entière "Int" d'une valeur flottante "Fp". Elle convertit la valeur en virgule flottante en format entier.

Le résultat est multiplié par le facteur exprimé en puissance de 10 dans la fenêtre d'ajustage. Par ex.: si le paramètre est -2 et la valeur 1234.56 resp. 1.234560E+3, la valeur en entier sera 12.

"Err" est mis haut en cas de dépassement de capacité et la conversion n'est pas effectuée.



#### 4.4.8.20 Format entier en format flottant



[\_conifp2]

Entrée: = Int: format entier

Sortie: – Fp: format flottant  
– Err: format binaire

Cette fonction donne la représentation flottante "Fp" d'une valeur entière "Int". Elle réalise la conversion d'une valeur entière en format virgule flottante.

Le résultat est multiplié par le facteur exprimé en puissance de 10 dans la fenêtre d'ajustage. Par ex.: si le paramètre est 3 et la valeur 12, la valeur en flottant sera 12000.00 resp. 1.200000E+4.



**Notes personnelles :**

#### 4.4.9 Adressage indirect

**[sfupindi]**

Cette famille de fonctions sert à l'adressage indirect d'éléments individuels et consécutifs. Les applications principales sont:

- Traitement de fonctions répétitives avec des adresses différées
- Choix de paramètres dans des grandes tables
- Partout où la flexibilité dans l'adressage est demandée

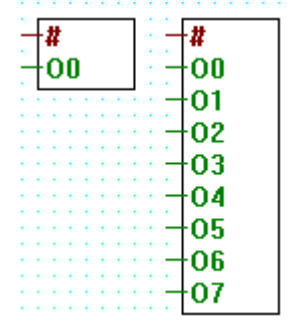
L'utilisation de cette famille est très universelle mais demande une certaine discipline à l'application. Une utilisation sans précaution peut provoquer des erreurs d'adressage difficiles à localiser.

- Mauvaise valeur à l'entrée #, surtout pour les fonctions "copie..."
- Fautes d'adressage lorsque la valeur est introduite par un appareil externe (supervision, terminaux, entrées BCD). Ceci peut être évité en introduisant un FBox "Limit" et un signal d'alarme.
- Conflits avec les plages d'adresses des variables dynamiques.
- Répartition des Temporisateur/Compteurs (Temporisateur deviennent Compteurs)
- Equipement Entrées/Sorties

#### 4.4.9.1 Copie indirecte de variables binaires sur des sorties consécutives.



[\_indioo]



Entrées:       = #:       Adresse de la 1ère sortie (0 - 8191)  
               - 00       Variable binaire  
               ⋮  
               - 07       Variable binaire

Cette fonction sert à copier 1 à 8 variables binaires sur des sorties consécutives de façon indirecte. L'adresse de la première sortie est donnée par la variable à l'entrée numérique #.

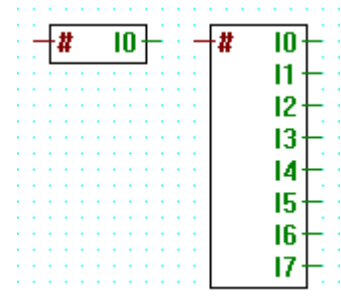
Consultez également les informations en début de chapitre 4.4.9.

---

#### 4.4.9.2 Lecture indirecte d'entrées consécutives.



[\_indioi]



Entrée:       = #       Adresse de la 1ère entrée (0 - 8191)

Sorties:      - I0       Variable binaire  
               ⋮  
               - I7       Variable binaires

Cette fonction sert à lire 1 à 8 entrées consécutives de façon indirecte. L'adresse de la première entrée est donnée par la variable à l'entrée numérique #.

Consultez également les informations en début de chapitre 4.4.9.

---



### 4.4.9.3 Copie indirecte de variables binaires sur des indicateurs consécutifs.



Entrées:       = #    Adresse du 1er indicateur (0 - 8191)  
               - F0    Variable binaire  
               |  
               - F7    Variable binaire

Cette fonction sert à copier 1 à 8 variables binaires sur des indicateurs consécutifs de façon indirecte. L'adresse du premier indicateur est donnée par la variable à l'entrée numérique #.

Consultez également les informations en début de chapitre 4.4.9.

### 4.4.9.4 Lecture indirecte d'indicateurs consécutifs.



Entrée:       = #    Adresse du 1er indicateur (0 - 8191)

Sorties:      - F0    Variable binaire  
               |  
               - F7    Variable binaire

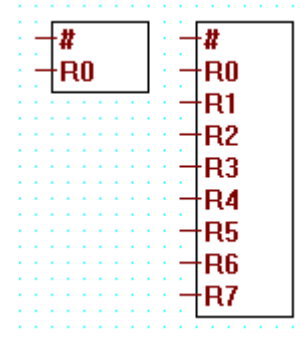
Cette fonction sert à lire 1 à 8 indicateurs consécutifs de façon indirecte. L'adresse du premier indicateur est donnée par la variable à l'entrée numérique #.

Consultez également les informations en début de chapitre 4.4.9.

#### 4.4.9.5 Copie indirecte de valeurs entières sur des registres consécutifs.



[\_indrego]



Entrées:       = #    Adresse du 1er registre (0 - 4095)  
               = R0   format entier  
               |  
               = R7   format entier

Cette fonction sert à copier 1 à 8 valeurs entières sur des registres consécutifs de façon indirecte. L'adresse du premier registre est donnée par la variable à l'entrée numérique #.

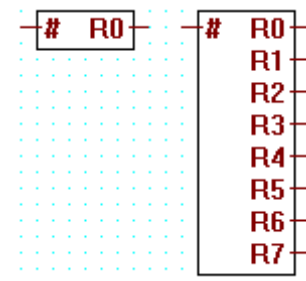
Consultez également les informations en début de chapitre 4.4.9.

---

#### 4.4.9.6 Lecture indirecte des registres consécutifs en format entier.



[\_indregi]



Entrée:       = #    Adresse du 1er registre (0 - 4095)

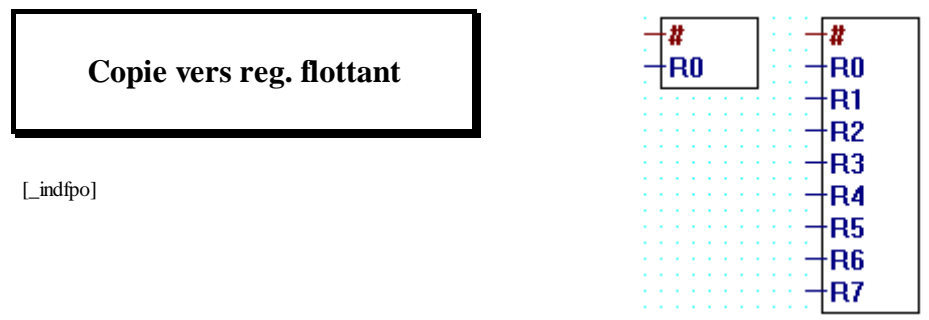
Sorties:      = R0   format entier  
               |  
               = R7   format entier

Cette fonction sert à lire 1 à 8 registres consécutifs en format entier de façon indirecte. L'adresse du premier registre est donnée par la variable à l'entrée numérique #.

Consultez également les informations en début de chapitre 4.4.9.

---

#### 4.4.9.7 Copie indirecte de valeurs flottantes sur des registres consécutifs.



Entrées:

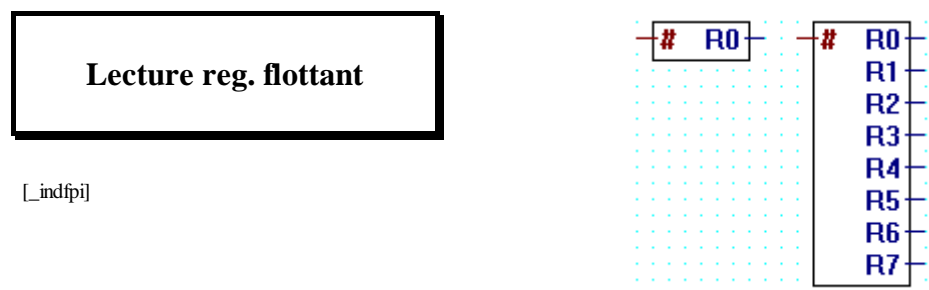
- = # Adresse du 1er registre (0 - 4095)
- = R0 format flottant
- ⋮
- = R7 format flottant

Cette fonction sert à copier 1 à 8 valeurs flottantes sur des registres consécutifs de façon indirecte. L'adresse du premier registre est donnée par la variable à l'entrée numérique #.

Consultez également les informations en début de chapitre 4.4.9.

---

#### 4.4.9.8 Lecture indirecte des registres consécutifs en format flottant.



Entrée: = # Adresse du 1er registre (0 - 4095)

Sorties:

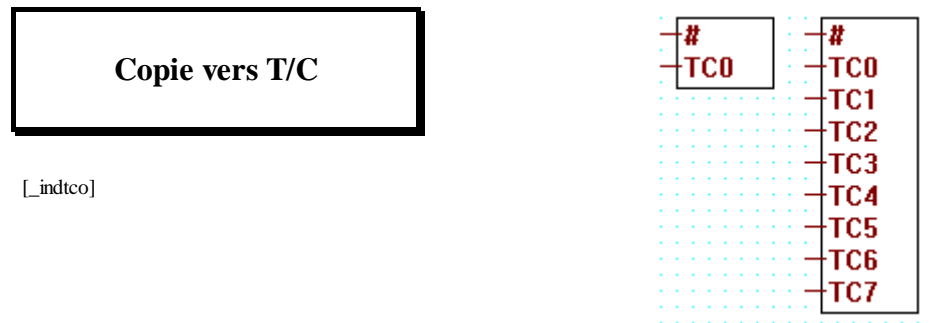
- = R0 format flottant
- ⋮
- = R7 format flottant

Cette fonction sert à lire 1 à 8 registres consécutifs en format flottant de façon indirecte. L'adresse du premier registre est donnée par la variable à l'entrée numérique #.

Consultez également les informations en début de chapitre 4.4.9.

---

#### 4.4.9.9 Copie indirecte de valeurs entières sur des T/C consécutifs.



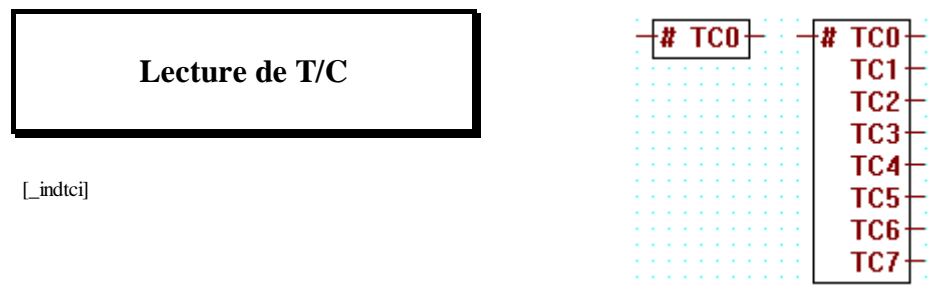
Entrées:       = #           Adresse du 1er T/C (0 - 1599)  
               = TC0       format entier  
               |  
               = TC7       format entier

Cette fonction sert à copier 1 à 8 valeurs entières sur des T/C consécutifs de façon indirecte. L'adresse du premier T/C est donnée par la variable à l'entrée numérique #.

Consultez également les informations en début de chapitre 4.4.9.

---

#### 4.4.9.10 Lecture indirecte des T/C consécutifs.



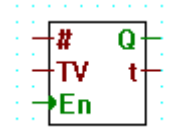
Entrée:       = #           Adresse du 1er T/C (0 - 1599)

Sorties:      = TC0       format entier  
               |  
               = TC0       format entier

Cette fonction sert à lire 1 à 8 T/C consécutifs de façon indirecte. L'adresse du premier T/C est donnée par la variable à l'entrée numérique #.

Consultez également les informations en début de chapitre 4.4.9.

---

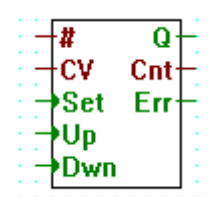
**4.4.9.11 Temporisateur avec adressage indirecte.**

[\_indtmr]

Entrées:	= #	Adresse du 1er Tempo	
	= TV	Valeur tempo en 1/10 de sec.	format entier
	> En	Enable (Départ du tempo)	format binaire
Sorties:	- Q	Sortie tempo	format binaire
	= t	Contenu tempo	format entier

Temporisateur avec adressage indirecte. L'adresse du Temporisateur est donnée par la variable à l'entrée numérique #.

---

**4.4.9.12 Compteur avec adressage indirecte.**

[\_indcnt]

Entrées:	= #	Adresse du 1er Compteur	
	= CV	Counter Value: Valeur initiale	format entier
	> Set	Présélection avec "CV"	format binaire
	> Up	Incrémenter (+1)	format binaire
> Dwn	Décrémenter (-1)	format binaire	
Sorties:	- Q	Sortie binaire (H, si C > 0)	format binaire
	= Cnt	Contenu du Compteur	format entier
	- Err	Error	format binaire

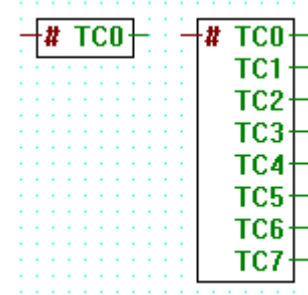
Compteur avec adressage indirecte. L'adresse du Compteur est donnée par la variable à l'entrée numérique #.

---

#### 4.4.9.13 Lecture des états logiques de Temporisateurs/Compteurs.



[\_indtcli]



Entrée: = # Adresse du 1er T/C (0 - 1599)

Sorties: – TC0 H/L T/C format binaire  
 |  
 – TC7 H/L T/C format binaire

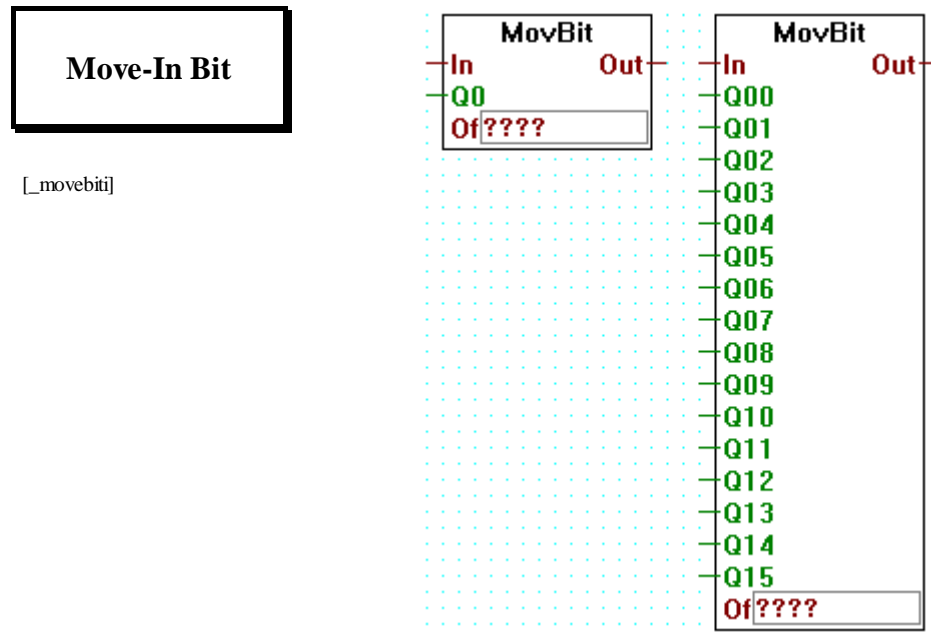
Lecture des états logiques de 1 à 8 Temporisateurs/Compteurs. L'adresse du premier T/C est donnée par la variable à l'entrée numérique #.

---

### 4.4.10 Move Data

[sfupmove]

#### 4.4.10.1 Move-In Bit



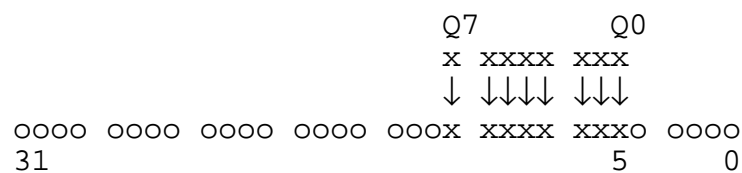
- Entrées:
- = In: Valeur initiale            format entier
  - Q00: Variable binaire (Q = Quantum) (0 ou 1)
  - |
  - Q15: Variable binaire
  - = Of: Offset, format entier (0 -31)

Sortie:            = Out: Valeur combinée            format entier

L'information numérique (registre, constante) qui est à combiner avec l'information binaire aux entrées "Q0"..."Q.." (max. "Q15") est mise à l'entrée numérique "In".

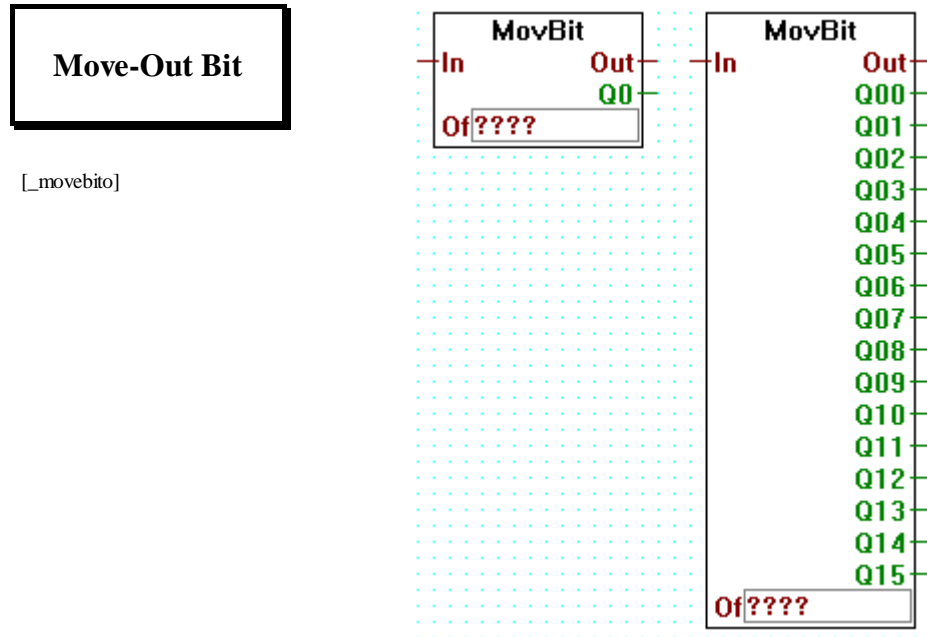
Le champ 'Of' (Offset) indique le numéro du premier bit introduit dans le registre. Si p.ex. <5> est donné, cela signifie que l'information aux entrées "Q0" à p.ex. "Q7" est placée sur les bits 5 à 12 dans le registre. Q0 est toujours le bit avec le poids le plus faible. Les autres bits du registre, pour cet exemple bit no. 0...4 et 13...31, ne sont pas changés.

"Move-In Bit" avec "Of" = 5 et entrées Q0 à Q7:



A la sortie numérique "Out" apparaît le contenu combiné du registre.

4.4.10.2 Move-Out Bit



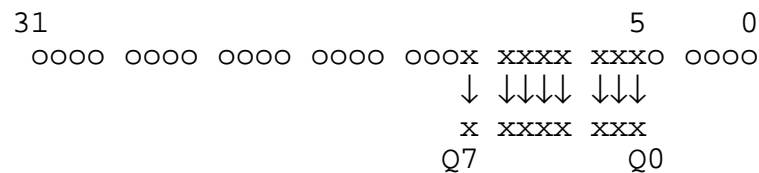
Entrées: = In: Valeur initiale format entier  
 = Of: Offset, format entier (0 -31)

Sorties: = Out: Valeur initiale format entier  
 - Q00: Variable binaire (Q = Quantum) (0 ou 1)  
 |  
 - Q15: Variable binaire

Copie une suite de bits d'un registre vers les sorties "Q0" à "Q.." (max. "Q15"). La valeur à transférer est mise à l'entrée numérique "In".

Le champ "Of" (Offset) indique le numéro du bit avec le poids le plus faible qui sort à la sortie Q0. Si p.ex. <5> est donné, cela signifie, que l'information dans le registre sur les bits 5 à p.ex. 12 est placée sur les sorties Q0 à Q7.

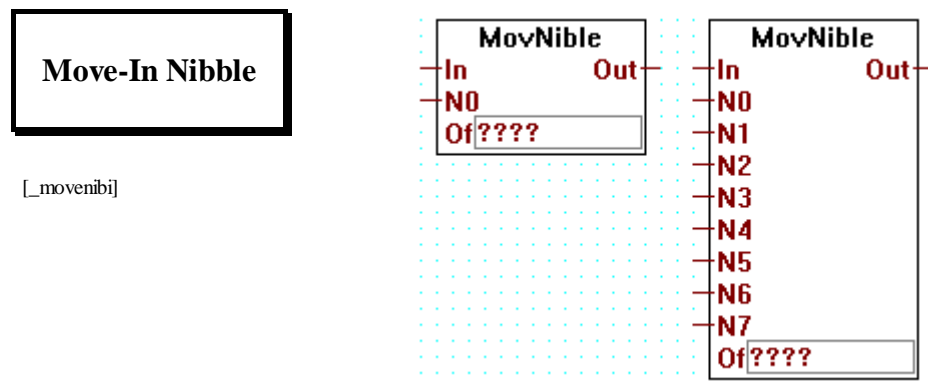
"Move-Out Bit" avec "Of" = 5 et sorties Q0 à Q7.



A la sortie numérique "Out" apparaît la valeur inchangée de l'entrée "In".



### 4.4.10.3 Move-In Nibble (4 Bit binaire)



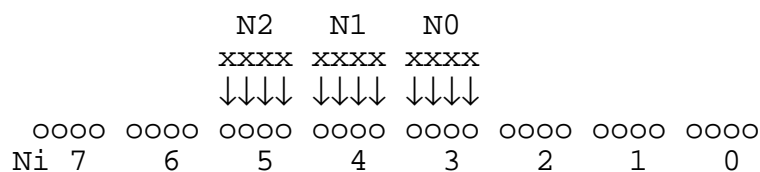
Entrées: = In: Valeur initiale format entier  
 = N0: Variable entière (0 - 15)  
 |  
 = N7: Variable entière  
 = Of: Offset, format entier(0 - 7)

Sortie: = Out: Valeur combinée format entier

L'information numérique (registre, constante) qui est à combiner avec l'information en format nibbles aux entrées "N0"... "N.." (max. "N7") est mise à l'entrée numérique "In".

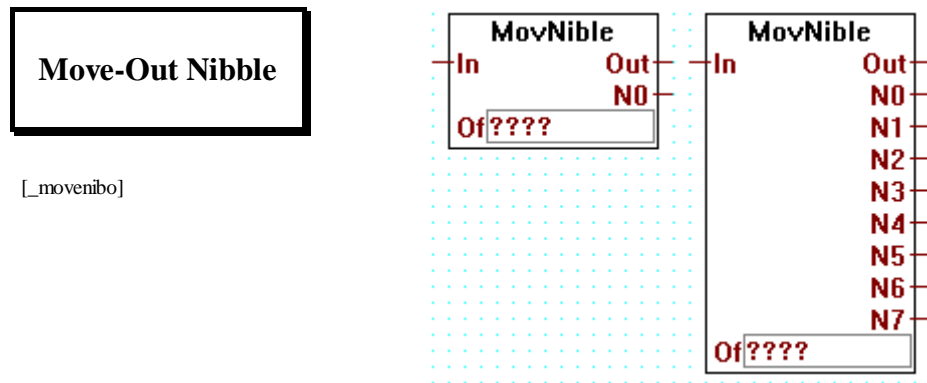
Le champ 'Of' (Offset) indique le numéro du premier nibble introduit dans le registre. Si p.ex. <3> est donné, cela signifie que l'information aux entrées "N0" à p.ex. "N2" est placée sur les nibbles 2 à 4 dans le registre. N0 est toujours le nibble avec le poids le plus faible. Les autres nibbles du registre, pour cet exemple bit no. 0...2 et 6...7, ne sont pas changés.

"Move-In Nibbles" avec "Of" = 3 et entrées N0 à N2.



A la sortie numérique "Out" apparaît le contenu combiné du registre.

**4.4.10.4 Move-Out Nibble (4 Bit binaire)**

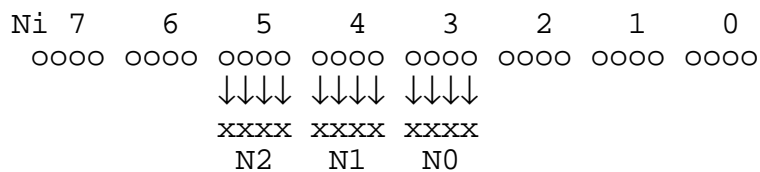


- Entrées:       = In:  Valeur initiale       format entier  
               = Of:  Offset, format entier(0 - 7)
- Sorties:       = Out: Valeur initiale       format entier  
               = N0:  Variable entière (0 - 15)  
               |  
               = N7:  Variable entière

Copie une suite de nibbles d'un registre vers les sorties "N0" à "N.." (max. "N15"). La valeur à transférer est mise à l'entrée numérique "In".

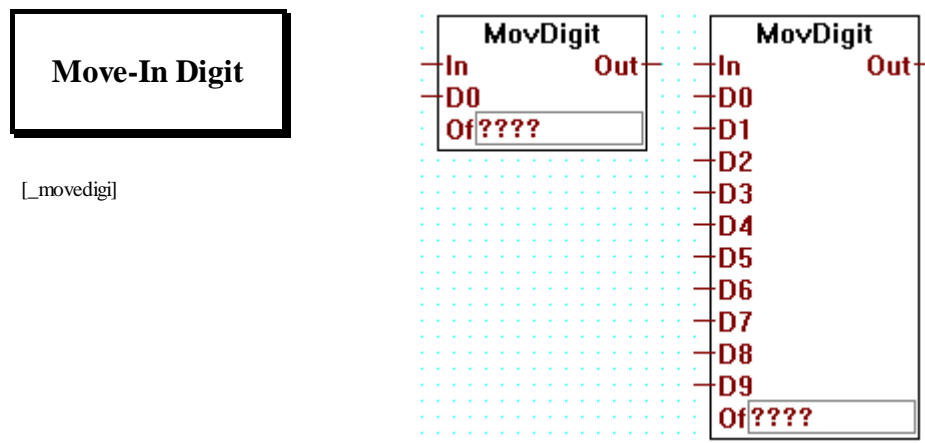
Le champ "Of" (Offset) indique le numéro du nibble avec le poids le plus faible qui sort à la sortie N0. Si p.ex. <3> est donné, cela signifie, que l'information dans le registre sur les nibbles 3 à p.ex. 5 est placée sur les sorties N0 à N2.

"Move-Out Nibble" avec "Of" = 3 et sorties N0 à N2.



A la sortie numérique "Out" apparaît la valeur inchangée de l'entrée "In".

### 4.4.10.5 Move-In Digit (4 Bit BCD)



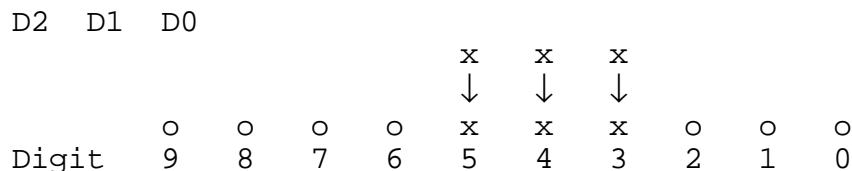
Entrées: = In: Valeur initiale format entier  
 = D0: Variable entière (0 - 9)  
 |  
 = D9: Variable entière  
 = Of: Offset, format entier (0 - 9)

Sortie: = Out: Valeur combinée format entier

L'information numérique (registre, constante) qui est à combiner avec l'information en format digit aux entrées "D0"..."D.." (max. "D9") est mise à l'entrée numérique "In".

Le champ 'Of' (Offset) indique le numéro du premier digit introduit dans le registre. Si p.ex. <3> est donné, cela signifie que l'information aux entrées "D0" à p.ex. "D2" est placée sur les digits 3 à 5 dans le registre. D0 est toujours le digit avec le poids le plus faible. Les autres digits du registre, pour cet exemple bit no. 0...2 et 6...9, ne sont pas changés.

"Move-In Digit" avec "Of" = 3 et entrées D0 à D2.



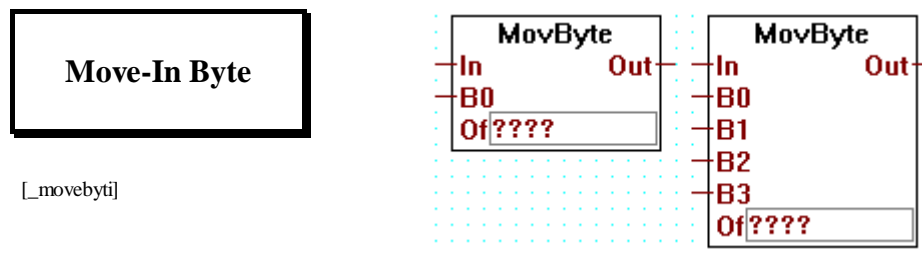
Seule des valeurs positives peuvent être traitées.

La valeur maximale pour digit 9 est 2.

A la sortie numérique "Out" apparaît le contenu combiné du registre.



### 4.4.10.7 Move-In Byte (8 Bit)



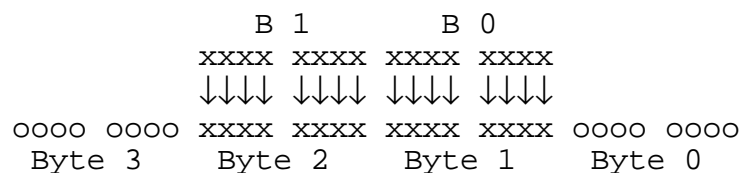
Entrées:           = In: Valeur initiale           format entier  
                   = B0: Variable entière (0 - 255)  
                   |  
                   = B3: Variable entière  
                   = Of: Offset, format entier (0 - 3)

Sortie:            = Out: Valeur combinée       format entier

L'information numérique (registre, constante) qui est à combiner avec l'information en format byte aux entrées "B0"..."B.." (max. "D3") est mise à l'entrée numérique "In".

Le champ 'Of' (Offset) indique le numéro du premier byte introduit dans le registre. Si p.ex. <1> est donné, cela signifie que l'information aux entrées "B0" à p.ex. "B1" est placée sur les bytes 1 à 2 dans le registre. B0 est toujours le byte avec le poids le plus faible. Les autres bytes du registre, pour cet exemple byte no. 0 et 3, ne sont pas changés.

"Move-In Byte" avec "Of" = 1 et entrées B0 et B1:

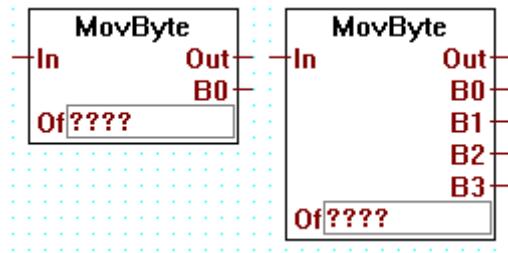


A la sortie numérique "Out" apparaît le contenu combiné du registre.

### 4.4.10.8 Move-Out Byte (8 Bit)



[\_movebyte]



Entrées:       = In:  Valeur initiale       format entier  
               = Of:  Offset, format entier (0 - 3)

Sorties:       = Out: Valeur initiale       format entier  
               = B0:  Variable entière (0 - 255)  
               |  
               = B3:  Variable entière

Copie une suite de bytes d'un registre vers les sorties "B0" à "B.." (max. "B3"). La valeur à transférer est mise à l'entrée numérique "In".

Le champ "Of" (Offset) indique le numéro du byte avec le poids le plus faible qui sort à la sortie B0. Si p.ex. <1> est donné, cela signifie que l'information dans le registre sur les bytes 1 à p.ex. 2 est placée sur les sorties B0 à B1.

"Move-Out Byte" avec "Of" = 1 et sorties B0 et B1.

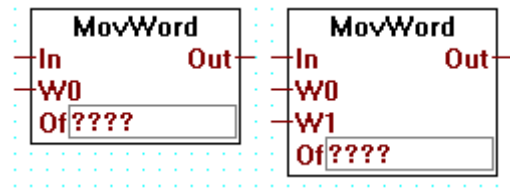
Byte 3	Byte 2	Byte 1	Byte 0	
0000	0000	xxxx	xxxx	xxxx
0000	xxxx	xxxx	xxxx	xxxx
	↓↓↓↓	↓↓↓↓	↓↓↓↓	↓↓↓↓
	xxxx	xxxx	xxxx	xxxx
	B 1	B 0		

A la sortie numérique "Out" apparaît la valeur inchangée de l'entrée "In".

**4.4.10.9 Move-In Word (16 Bit)**



[\_moveword]



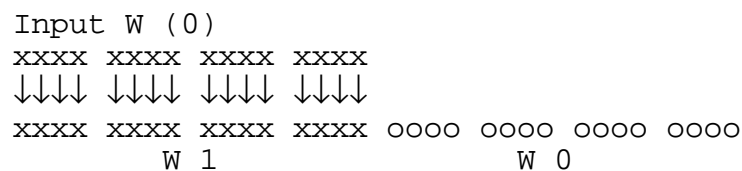
Entrées:           = In: Valeur initiale           format entier  
                   = W0: Variable entière (0 - 65535)  
                   |  
                   = W1: Variable entière  
                   = Of: Offset, format entier (0 - 1)

Sortie:            = Out: Valeur combinée       format entier

L'information numérique (registre, constante) qui est à combiner avec l'information en format word aux entrées "W0"..."W.." (max. "W1") est mise à l'entrée numérique "In".

Le champ 'Of' (Offset) indique le numéro du premier word introduit dans le registre. Si p.ex. <1> est donné, cela signifie que l'information aux entrées "W (0)" est placée sur le word 1 dans le registre. W0 est toujours le word avec le poids le plus faible. L'autre word du registre, pour cet exemple word no. 0, n'est pas changés.

"Move-In Word" avec "Of" = 1 et entrée W (0):

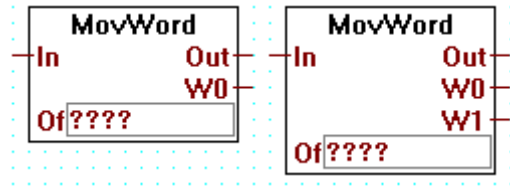


A la sortie numérique "Out" apparaît le contenu combiné du registre.

**4.4.7.10 Move-Out Word (16 Bit)**



[\_moveworo]



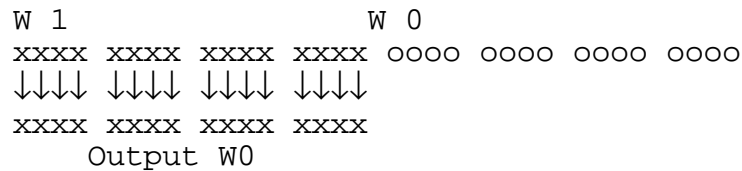
Entrées: = In: Valeur initiale format entier  
 = Of: Offset, format entier (0 - 1)

Sorties: = Out: Valeur initiale format entier  
 = W0: Variable entière (0 - 65535)  
 |  
 = W1: Variable entière

Copie une suite de words d'un registre vers les sorties "W0" à "W.." (max. "W1"). La valeur à transférer est mise à l'entrée numérique "In".

Le champ "Of" (Offset) indique le numéro du word avec le poids le plus faible qui sort à la sortie W0. Si p.ex. <1> est donné, cela signifie que l'information dans le registre sur le word 1 est placée sur la sortie W0.

"Move-Out Word" avec "Of" = 1 et la sortie W0.



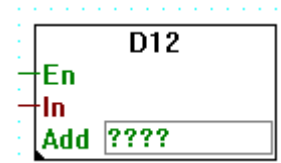
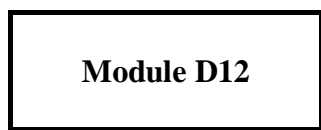
A la sortie numérique "Out" apparaît la valeur inchangée de l'entrée "In".



## 4.4.11 Afficheurs

[sfupdisp]

### 4.4.11.1 Module d'affichage PCA2.D12



[\_dispd12]

Entrées:       – En: Enable  
               = In:  Format entier

Cette fonction permet de sortir une valeur entière sur le module d'affichage PCA2.D12.

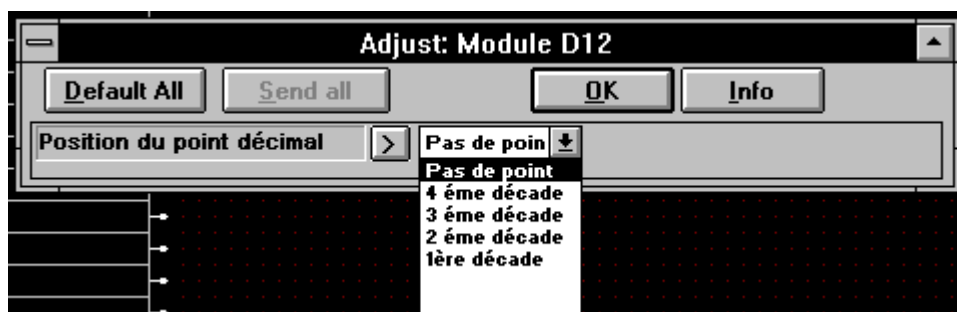
Quand la boîte de fonction est active, les 4 derniers chiffres de l'entier sont visibles.

Le D12 est activé par l'entrée "En". La fenêtre de paramétrage "Add" contient l'adresse de base de câblage du module. La boîte de fonction implique que "Clk", "D-IN" et "En" soient respectivement raccordées à Add+0, Add+1, Add+2.

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

L'affichage reste inchangé si la boîte de fonction est désactivée.

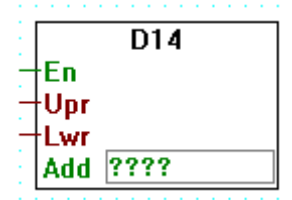
Un point décimal peut être ajouté manuellement. Celui-ci sera défini dans la boîte de dialogue accessible par un double clics sur le symbole. Le paramètre ne sera effectif qu'après recompilation du programme.



#### 4.4.11.2 Module d'affichage PCA2.D14



[\_dispd14]



- Entrées:
- En: Enable
  - = Upr: Affichage supérieur format entier
  - = Lwr: Affichage inférieur format entier

Cet afficheur permet de sortir deux valeurs entières sur le module d'affichage PCA2.D14.

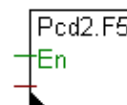
Quand la boîte de fonction est active, les 6 derniers chiffres de l'entier "Upr" sont visibles dans l'affichage supérieur et les 6 derniers chiffres de l'entier "Lwr" dans l'affichage du bas.

Le D14 est activé par l'entrée "En". La fenêtre de paramétrage "Add" contient l'adresse de base de câblage du module. La boîte de fonction implique que "Clk", "D-IN" et "En" soient respectivement raccordées à Add+0, Add+1, Add+2.

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

L'affichage reste inchangé si la boîte de fonction est désactivée.

### 4.4.11.3 Module d'affichage PCD2.F510 pour affichages numériques



[\_disppcd23]

Entrées: – Enable                                   format binaire  
           = Valeur à afficher                   format entier

Cette fonction permet de sortir sur le module d'affichage PCD2.F510 une valeur d'une capacité de 6 décades au maximum (de -99999 à 999999).

Il peut également afficher des valeurs numériques prédéfinies. Les différentes sélections sont à faire dans la fenêtre d'ajustage.

Si l'entrée Enable "En" = L, le dernier format défini reste actuel. Le format par défaut est "6 Décades".



Mode	Affichage du module
------	---------------------

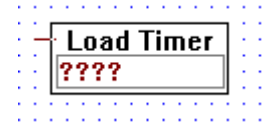
- 6 Décades    6 décades de la valeur d'entrée.  
Si cette valeur n'est pas comprise entre -99999 et 999999 l'affichage est effacé.
- Hlp nn       'HLP nn'       (nn = 2 chiffres 00-99)
- Err nn       'Err nn' suivi de 2 décades de la valeur.  
Si les 2 décades de la valeur ne sont pas comprises entre 0 et 99, l'affichage est effacé. (nn = 2 chiffres 00-99)
- 0 non sign.  6 décades y compris les zéros non significatifs.
- 2 Décades    2 décades. Si les 2 décades de la valeur ne sont pas comprises entre 0 et 99, l'affichage est effacé.
- Horloge      Affiche la valeur en prenant un format dateur.



## 4.4.12 Fonctions GRAFTEC

**[sfupgraf]**

### 4.4.12.1 Chargement d'un Temporisateur sans condition

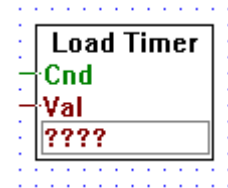


[\_grafldt]

Entrée: = Valeur de temp. - Registre ou Constante    format entier

Fonction pour charger et démarrer un Temporisateur.

#### 4.4.12.2 Chargement conditionnel d'un Temporisateur



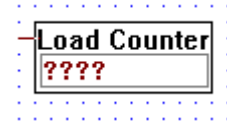
[\_grafcldt]

Entrées: – Cnd: Condition H/L format binaire  
= Valeur de temp. - Registre ou Constante format entier

Fonction pour le chargement conditionnel d'un Temporisateur.

---

### 4.4.12.3 Chargement d'un Compteur sans condition



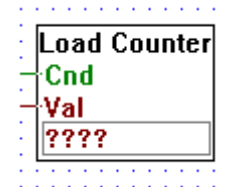
[\_grafldc]

Entrée: = Valeur de charge - Registre ou Constante    format entier

Fonction pour charger un Compteur.

---

### 4.4.12.4 Chargement conditionnel d'un Compteur



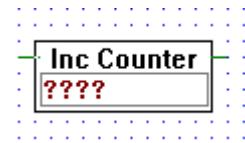
[\_grafldc]

Entrées: – Cnd:                    Condition H/L            format binaire  
           = Valeur de charge - Registre ou Constante    format entier

Fonction pour le chargement conditionnel d'un Compteur.

---

#### 4.4.12.5 Fonction pour incrémenter un Compteur



[\_grafinc]

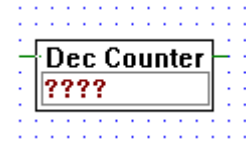
Entrée: – Condition pour incrémenter: H/L format binaire

Sortie: – Etat logique du Compteur: H/L format binaire

Fonction pour incrémenter (+1) un Compteur.

---



**4.4.12.6 Fonction pour décrémenter un Compteur**

[\_grafdecc]

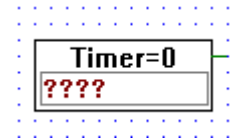
Entrée: – Condition pour décrémenter: H/L format binaire

Sortie: – Etat logique du Compteur: H/L format binaire

Fonction pour décrémenter (-1) un Compteur.

**4.4.12.7 Est-ce que Temporisateur est écoulé ?**

[\_grafcis]

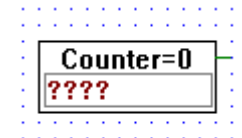


Sortie: – H, si temporisation est écoulée      format binaire

Fonction pour scruter l'état logique d'un Temporisateur.

**4.4.12.8 Est-ce que Compteur = 0 ?**

[\_grafcis]



Sortie: – H, si Compteur = 0.      format binaire

Fonction pour scruter l'état logique d'un Compteur. Si le Compteur = 0, la fonction est remplie

#### 4.4.12.9 Fin Transition



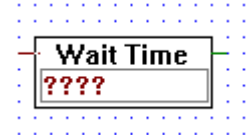
[\_grafetrsv]

Entrée: – format binaire

Fin d'une Transition (ETR).

---

#### 4.4.12.10 Charger un Temporisateur et attendre la fin de la temporisation



[\_grafwait]

Entrée: = Valeur de temp. - Registre ou Constante format entier

Sortie: – H, si temporisation est écoulée format binaire

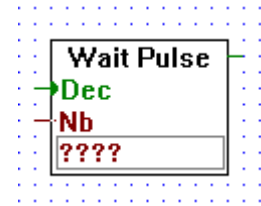
Fonction combinée pour programmer une pause dans une TR. Lors du premier traitement de la TR, le temporisateur est chargé avec la valeur d'entrée et démarré immédiatement. La sortie passe à 1 quand la temporisation est écoulée.

---

#### 4.4.12.11 Charger Compteur et attendre jusqu'à Compteur = 0.



[\_grafwpls]

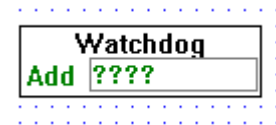


Entrées:            > Dec: signal de décrémentation            format binaire  
                       = Nb: Registre ou Constante            format entier

Sortie:             – H, si Compteur = 0.                        format binaire

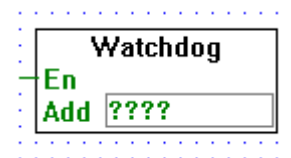
Fonction combinée pour programmer une pause dans une TR. La pause se termine après la réception d'un nombre d'impulsions présélectionné. Au premier traitement de la TR, le compteur est chargé avec la valeur d'entrée. Chaque impulsion à l'entrée "Dec" décrémente le compteur d'une unité. La sortie passe à H quand le compteur atteint zéro.

---

**4.4.13 Fonctions spéciales (Chien de garde)****[sfupspec]****4.4.13.1 Chien de garde (toujours actif)**

[\_watchdog]

Si cette fonction est appelée à une fréquence supérieure à 5 Hz, le circuit du chien de garde est actif. Ce dispositif contrôle le bon fonctionnement du système en permettant à l'utilisateur d'intervenir.

**4.4.13.2 Chien de garde activable**

[\_watchdoge]

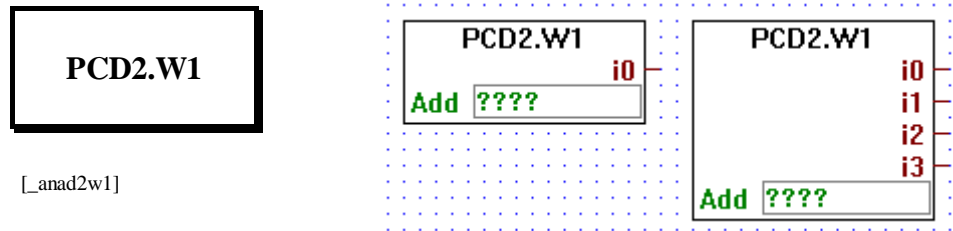
Entrée: – En Enable format binaire

Si cette fonction est appelé à une fréquence supérieure à 5 Hz, le circuit du chien de garde est actif lorsque l'entrée de validation "En" est au niveau "H" .

**Notes personnelles :**

## 4.4.14 Modules d'entrées/sorties analogiques [sfupanlg]

### 4.4.14.1 Module d'entrées analogiques PCD2.W1 (12 bits)



Entrées: = i0  
 | format entier  
 = i3

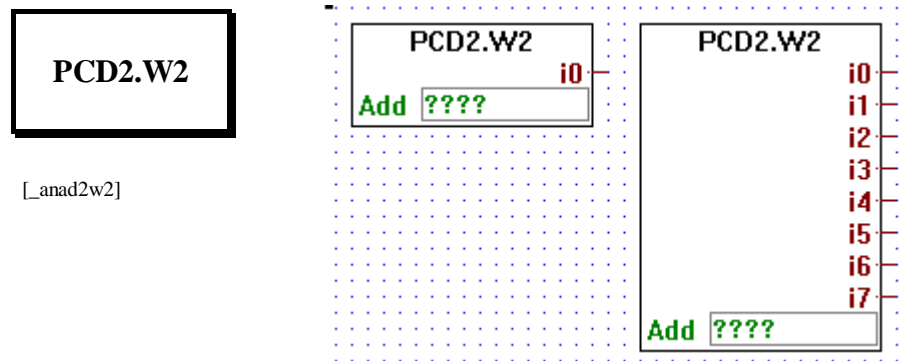
Cette boîte de fonction réalise la conversion des canaux d'entrées analogiques (1 à 4) d'une carte PCD2.W1.

A chaque exécution de cette fonction, un canal d'entrée est converti. Tant que la conversion n'est pas terminée, la sortie de la boîte reste inchangée. Si un canal n'a jamais subi une conversion depuis l'initialisation du système, la valeur de sortie sera zéro.

Une boîte de fonction doit être employée pour chaque carte PCD2.W1 utilisée.

Le paramètre "Add" est l'adresse de base du module, par ex: O 16. L'adresse 240 ne doit jamais être utilisée pour un module analogique (dysfonctionnement du Chien de Garde).

#### 4.4.14.2 Module d'entrées analogiques PCD2.W2 (10 bits)



Entrées: = i0  
 | format entier  
 = i7

Cette boîte de fonction réalise la conversion des canaux d'entrées analogiques (1 à 8) d'une carte PCD2.W2.

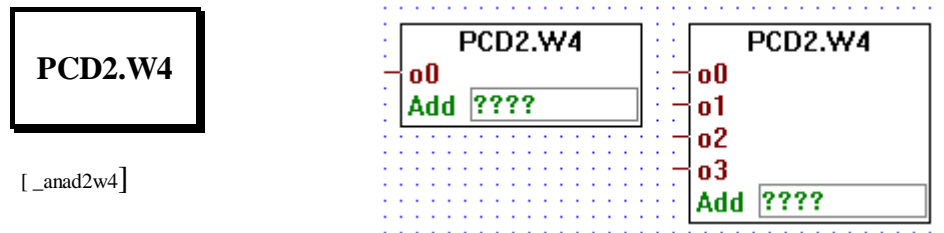
A chaque exécution de cette fonction, un canal d'entrée est converti. Tant que la conversion n'est pas terminée, la sortie de la boîte reste inchangée. Si un canal n'a jamais subi une conversion depuis l'initialisation du système, la valeur de sortie sera zéro.

Une boîte de fonction doit être employée pour chaque carte PCD2.W2 utilisée.

Le paramètre "Add" est l'adresse de base du module, par ex: O 16. L'adresse 240 ne doit jamais être utilisée pour un module analogique (dysfonctionnement du Chien de Garde).



#### 4.4.14.3 Module de sorties analogiques PCD2.W4 (8 bits)



Sorties: = o0  
 | format entier  
 = o3

Cette boîte de fonction réalise la conversion de valeurs numériques (éritable de 1 à 4 entrées) en sorties analogiques du module PCD2.W4 de conversion.

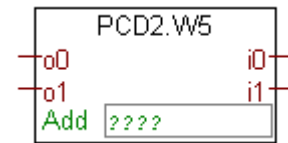
Une boîte de fonction doit être employée pour chaque carte PCD2.W4 utilisée.

Le paramètre "Add" est l'adresse de base du module, par ex: O 16.  
 L'adresse 240 ne doit jamais être utilisée pour un module analogique (dysfonctionnement du Chien de Garde).

#### 4.4.14.4 Module d'entrées/sorties analogiques PCD2.W5 (12 bits)



[\_anad2w5 ]



Entrées: = i0 format entier  
= i1 format entier

Sorties: = o12 format entier  
= o13 format entier

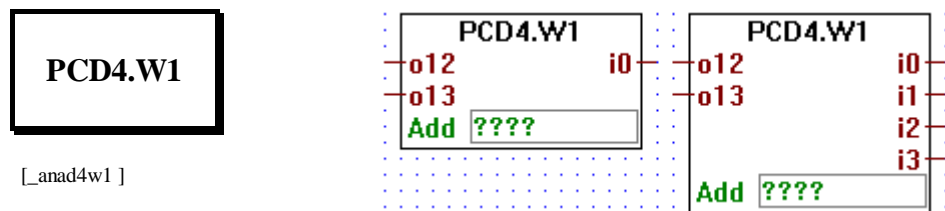
Cette boîte de fonction réalise la conversion des canaux d'entrées analogiques (i0 et i1) et la conversion de valeurs numériques en sortie analogiques (o0 et o1) d'un module PCD2.W5.

A chaque exécution de cette fonction, un canal d'entrée est converti. Tant que la conversion n'est pas terminée, la sortie de la boîte reste inchangée. Si un canal n'a jamais subi une conversion depuis l'initialisation du système, la valeur de sortie sera zéro.

Une boîte de fonction doit être employée pour chaque carte PCD2.W5 utilisée.

Le paramètre "Add" est l'adresse de base du module, par ex: O 16. L'adresse 240 ne doit jamais être utilisée pour un module analogique (dysfonctionnement du Chien de Garde).

#### 4.4.14.5 Module d'entrées/sorties analogiques PCD4.W1 (12 bits)



Entrées: = i0  
 | format entier  
 = i3

Sorties: = o12  
 | format entier  
 = o13

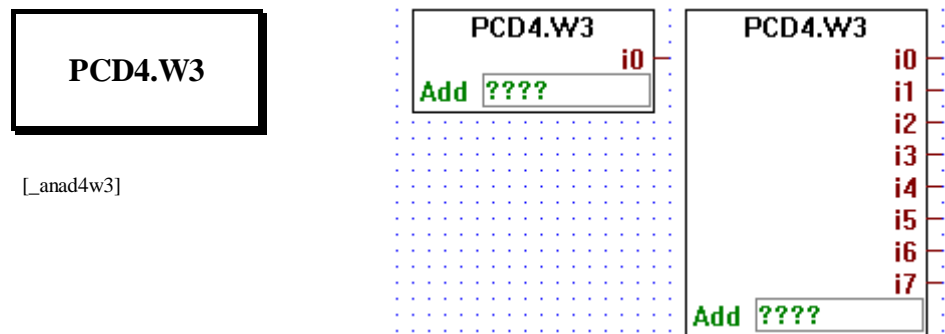
Cette boîte de fonction réalise les conversions analogiques/digitales des canaux d'entrées du module PCD4.W1. Cette carte pouvant également être équipée d'une ou deux sorties analogiques, les lignes "o12" et "o13" correspondent respectivement aux sorties relatives 12 et 13 de la carte. Les 4 sorties "i0" à "i3" du module correspondent aux 4 entrées relatives 0 à 3 de la carte.

A chaque exécution de cette fonction, un canal d'entrée est converti. Tant que la conversion n'est pas terminée, la sortie de la boîte reste inchangée. Si un canal n'a jamais subi une conversion depuis l'initialisation du système, la valeur de sortie sera zéro.

Une boîte de fonction doit être employée pour chaque carte PCD4.W1 utilisée.

Le paramètre "Add" est l'adresse de base du module, par ex: O 16. L'adresse 240 ne doit jamais être utilisée pour un module analogique (dysfonctionnement du Chien de Garde).

#### 4.4.14.6 Module d'entrées analogiques PCD4.W3 (12 bits et signe)



Entrées: = i0  
 | format entier  
 = i7

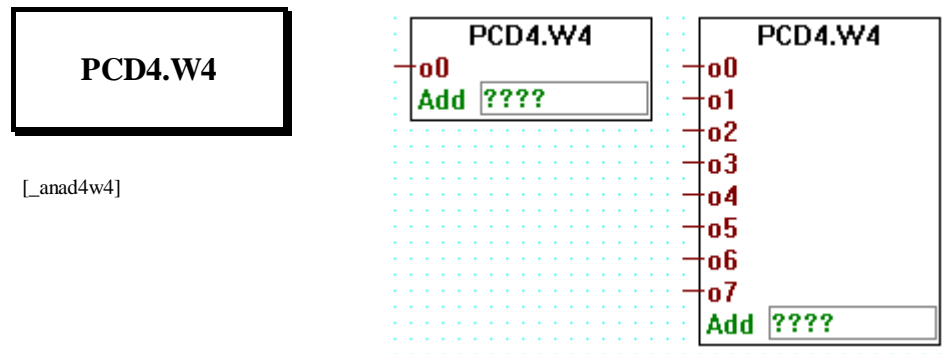
Cette boîte de fonction réalise la conversion des canaux d'entrées analogiques (étirable de 1 à 8 ) d'une carte PCD4.W3.

A chaque exécution de cette fonction, un canal d'entrée est converti. Tant que la conversion n'est pas terminée, la sortie de la boîte reste inchangée. Si un canal n'a jamais subi une conversion depuis l'initialisation du système, la valeur de sortie sera zéro.

Une boîte de fonction doit être employée pour chaque carte PCD4.W3 utilisée.

Le paramètre "Add" est l'adresse de base du module, par ex: O 16. L'adresse 240 ne doit jamais être utilisée pour un module analogique (dysfonctionnement du Chien de Garde).

#### 4.4.14.7 Module de sorties analogiques PCD4.W4 (8 bits)



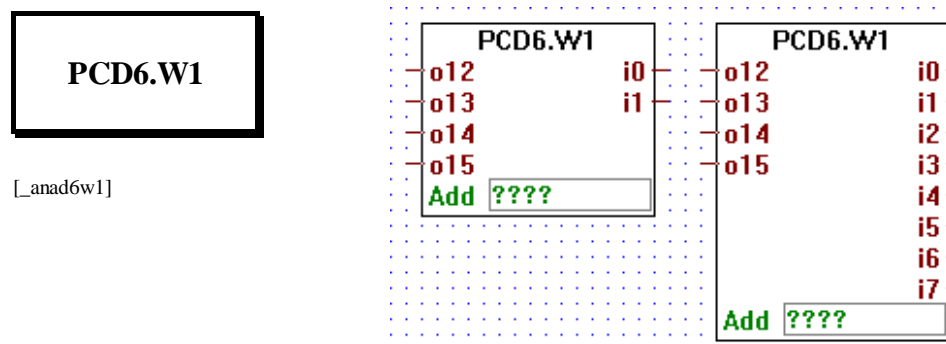
Sorties: = o0  
 | format entier  
 = o7

Cette boîte de fonction réalise la conversion de valeurs numériques (éritable de 1 à 8) en sorties analogiques du module PCD4.W4 de conversion.

Une boîte de fonction doit être employée pour chaque carte PCD4.W4 utilisée.

Le paramètre "Add" est l'adresse de base du module, par ex: O 16.  
 L'adresse 240 ne doit jamais être utilisée (généralité) pour un module analogique (dysfonctionnement du Chien de Garde).

#### 4.4.14.8 Module d'entrées/sorties analogiques PCD6.W1 (12 bits)



Sorties: = o12  
 | format entier  
 = o15

Entrées: = i0  
 | format entier  
 = i7

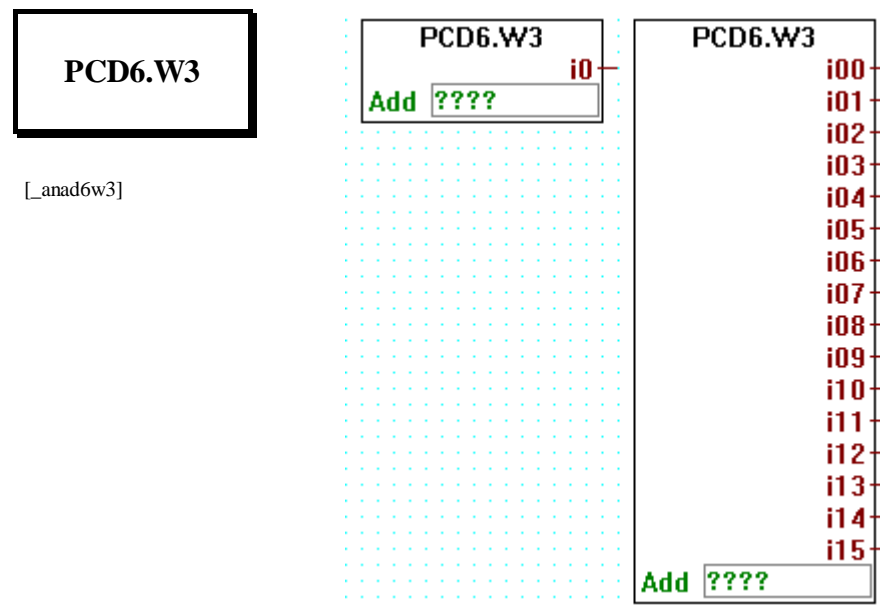
Cette boîte de fonction réalise les conversions analogiques/digitales des canaux d'entrées du module PCD6.W1. Cette carte pouvant également être équipée d'une à quatre sorties analogiques, les lignes "o12" à "o15" correspondent respectivement aux sorties relatives 12 à 15 de la carte. Les 8 sorties "i0" à "i7" du module correspondent aux 8 entrées relatives 0 à 7 de la carte.

A chaque exécution de cette fonction, un canal d'entrée est converti. Tant que la conversion n'est pas terminée, la sortie de la boîte reste inchangée. Si un canal n'a jamais subi une conversion depuis l'initialisation du système, la valeur de sortie sera zéro.

Une boîte de fonction doit être employée pour chaque carte PCD6.W1 utilisée.

Le paramètre "Add" est l'adresse de base du module, par ex: O 160. L'adresse tiroir +240 ne doit jamais être utilisée pour un module analogique (dysfonctionnement des Chiens de Garde).

#### 4.4.14.9 Module d'entrées analogiques PCD6.W3 (12 bits et signe)



Entrées: = i00  
 | format entier  
 = i15

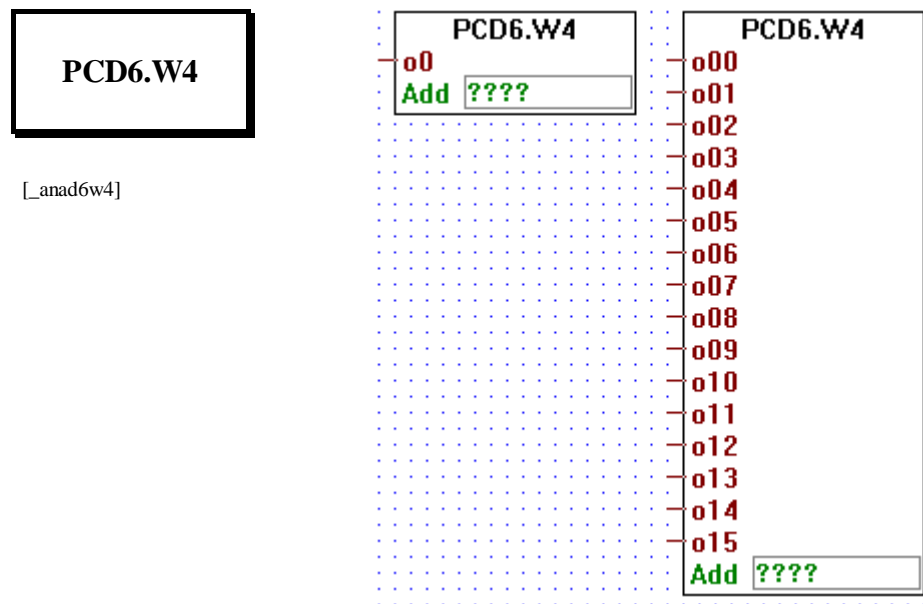
Cette boîte de fonction réalise la conversion des canaux d'entrées analogiques (étirable de 1 à 16) d'une carte PCD6.W3.

A chaque exécution de cette fonction, un canal d'entrée est converti. Tant que la conversion n'est pas terminée, la sortie de la boîte reste inchangée. Si un canal n'a jamais subi une conversion depuis l'initialisation du système, la valeur de sortie sera zéro.

Une boîte de fonction doit être employée pour chaque carte PCD6.W3 utilisée.

Le paramètre "Add" est l'adresse de base du module, par ex: O 160. L'adresse tiroir +240 ne doit jamais être utilisée pour un module analogique (dysfonctionnement des Chiens de Garde).

**4.4.14.10 Module de sorties analogiques PCD6.W4 (8 bits)**



Sorties: = o00  
           |       format entier  
           = o15

Cette boîte de fonction réalise la conversion des valeurs numériques (éritable de 1 à 16) en sorties analogiques du module PCD6.W4.

Une boîte de fonction doit être employée pour chaque carte PCD6.W4 utilisée.

Le paramètre "Add" est l'adresse de base du module, par ex: O 160. L'adresse tiroir +240 ne doit jamais être utilisée pour un module analogique (dysfonctionnement des Chiens de Garde).



## 4.4.15 Régulation

[sfupregu]

### 4.4.15.1 Régulateur PID



[\_regpid3]



Entrées:	= W:	consigne	format entier
	= X:	grandeur réglée	format entier
	= YS:	grandeur de réglage pour la réinitialisation	format entier
	- CS:	signal de réinitialisation	format binaire
Sortie:	= Y:	grandeur de réglage	format entier

Cette description admet que l'utilisateur dispose de bonnes connaissances de la technique de régulation. L'instruction PID est décrite plus en détail dans le manuel PCD "Guide des Instructions" (26/733).

#### Redémarrage et mode manuel:

YS est la valeur prise comme grandeur de réglage lors de la réinitialisation. Si le signal CS est dynamisé, le réglage reprend normalement après la réinitialisation. Si le signal CS reste à l'état 1 sur une plus longue période, la valeur YS est transmise à la sortie Y à chaque échantillonnage. Le régulateur est alors hors service et peut être piloté en 'mode manuel'.

Dans la fenêtre d'ajustage, les 6 paramètres suivants peuvent être définis.

L'échantillonnage, le facteur proportionnel, le facteur intégral, la dérivée, la zone morte ainsi que le nombre de bit de résolution requis.

**Echantillonnage:** Défini le temps entre deux acquisitions. L'unité de temps utilisée est la même que celle qui a été définie pour les temporisations. Voir également l'instruction DEFTB.

**Facteur proportionnel Fp:** Ce facteur P détermine la part proportionnelle (amplification) du régulateur. Dans le calcul, seuls les 16 bits de poids faibles sont utilisés (0..65535).

Ce facteur est déterminé de la façon suivante:

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

avec  $X_p$ : Bande proportionnelle.

**Facteur intégral Fi:** Ce facteur I détermine l'incidence intégrale du régulateur. Dans le calcul, seuls les 16 bits de poids faibles sont utilisés (0..65535).

Ce facteur est déterminé de la façon suivante:

$$F_i = (T_0/T_i) * 256$$

avec  $T_0$ : Temps d'échantillonnage de la fonction PID  
 $T_i$ : Temps d'intégration.

**Facteur dérivé Fd:** Ce facteur D détermine la dérivée du régulateur. Dans le calcul, seuls les 16 bits de poids faibles sont utilisés (0..65535).

Ce facteur est déterminé de la façon suivante:

$$F_d = (T_d/T_0) * 256$$

avec  $T_0$ : Temps d'échantillonnage de la fonction PID  
 $T_d$ : Temps de dérivée.

**Zone morte Dr:** La zone morte correspond à une bande à l'intérieure, de laquelle les variations de calcul ne modifient pas la valeur de la correction MV.

**Départ à froids Ys:** Cette valeur est utilisée comme valeur initiale pour  $Y_n$ . Dès que l'utilisateur met une valeur autre que zéro dans ce paramètre, cette valeur est utilisée par un algorithme spécial dit de "départ à froids" dans lequel il intervient en temps que  $Y_n$ .

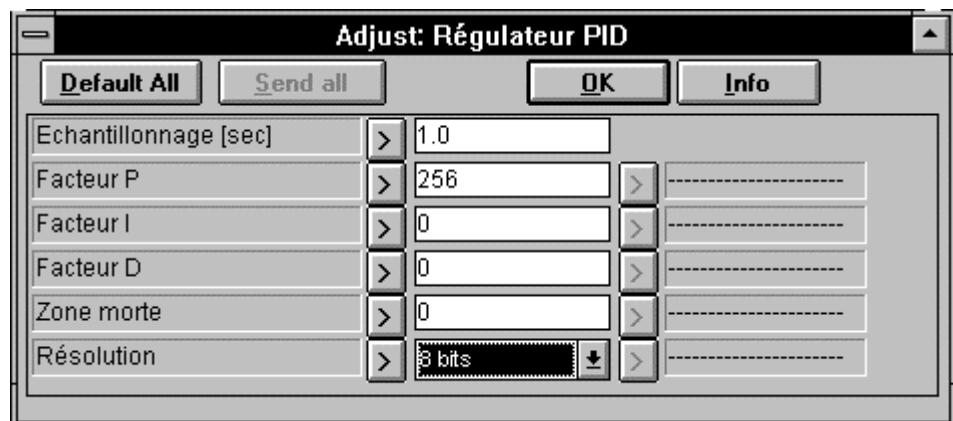
Note: Pour étouffer la pointe au démarrage,  $Y_s$  peut être chargé avec la valeur actuelle AV. Dès que le traitement "départ a froids" est terminé,  $Y_s$  est remis a zéro et algorithme standard est utilisé.

**Résolution:** Les variations maximums sont déterminées par la résolution.

8 bits: 0...255; 12 bits: 0...4095; 16 bits: 0...65'535.

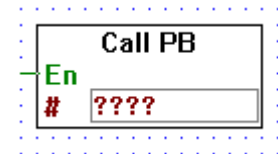
La résolution est principalement définie par le module analogique utilisant le résultat du calcul.

Fenêtre d'ajustage:



**4.4.16 Fonctions utilisateur****[sfupuser]**

Un outil pour créer des boîtes de fonctions est en préparation.

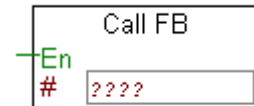
**4.4.16.1 Appel d'un bloc de programme (PB)**

[\_call\_pb]

Entrées:           – En: Enable                   format binaire  
                  #    Adresse du PB           format entier

Appel le PB numéro # quand l'entrée "En" est "Haut".

---

**4.4.16.2 Appel d'un bloc de fonction (FB)**

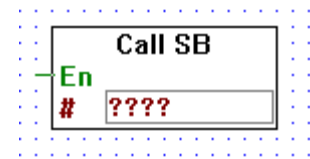
[\_call\_fb]

Entrées:           – En: Enable                   format binaire  
                  #    Adresse du FB           format entier

Appel le FB numéro # quand l'entrée "En" est "Haut".

---

### 4.4.16.3 Appel d'un bloc séquentiel (SB)



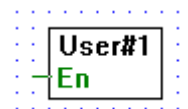
[\_call\_sb]

Entrées:      – En: Enable                   format binaire  
              #     Adresse du SB         format entier

Appel le SB numéro # (GRAFTEC) quand l'entrée "En" est "Haut".

---

#### 4.4.16.4 Bloc utilisateur 1



[\_user\_1]

Entrée: – En: Enable           format binaire

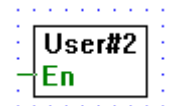
Cette fonction procure un chemin pour inclure un langage d'instruction de l'utilisateur.

Quand l'entrée "En" est active (état haut), le segment de programme est exécuté. Le morceau de programme doit être écrit dans un fichier portant le nom "user\_1.h".

Cette fonction n'est pas parfaite et ne permet pas de "passer" des paramètres au programme utilisateur. Nous vous conseillons de ne pas trop utiliser cette fonction.

---

#### 4.4.16.5 Bloc utilisateur 2



[\_user\_2]

Entrée: – En: Enable           format binaire

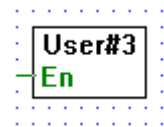
Cette fonction procure un chemin pour inclure un langage d'instruction de l'utilisateur.

Quand l'entrée "En" est active (état haut), le segment de programme est exécuté. Le morceau de programme doit être écrit dans un fichier portant le nom "user\_2.h".

Cette fonction n'est pas parfaite et ne permet pas de "passer" des paramètres au programme utilisateur. Nous vous conseillons de ne pas trop utiliser cette fonction.

---

#### 4.4.16.6 Bloc utilisateur 3



[\_user\_3]

Entrée: – En: Enable           format binaire

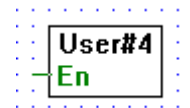
Cette fonction procure un chemin pour inclure un langage d'instruction de l'utilisateur.

Quand l'entrée "En" est active (état haut), le segment de programme est exécuté. Le morceau de programme doit être écrit dans un fichier portant le nom "user\_3.h".

Cette fonction n'est pas parfaite et ne permet pas de "passer" des paramètres au programme utilisateur. Nous vous conseillons de ne pas trop utiliser cette fonction.

---

#### 4.4.16.7 Bloc utilisateur 4



[\_user\_4]

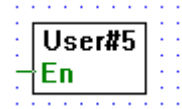
Entrée: – En: Enable           format binaire

Cette fonction procure un chemin pour inclure un langage d'instruction de l'utilisateur.

Quand l'entrée "En" est active (état haut), le segment de programme est exécuté. Le morceau de programme doit être écrit dans un fichier portant le nom "user\_4.h".

Cette fonction n'est pas parfaite et ne permet pas de "passer" des paramètres au programme utilisateur. Nous vous conseillons de ne pas trop utiliser cette fonction.

---

**4.4.16.8 Bloc utilisateur 5**

[\_user\_5]

Entrée: – En: Enable            format binaire

Cette fonction procure un chemin pour inclure un langage d'instruction de l'utilisateur.

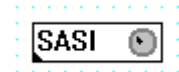
Quand l'entrée "En" est active (état haut), le segment de programme est exécuté. Le morceau de programme doit être écrit dans un fichier portant le nom "user\_5.h".

Cette fonction n'est pas parfaite et ne permet pas de "passer" des paramètres au programme utilisateur. Nous vous conseillons de ne pas trop utiliser cette fonction.

---

**Notes personnelles :**



**4.4.17 Communication série****[sfupcomm]****4.4.17.1 Paramètres d'interface série: SASI**

[\_comsasi]

**Initialisation de l'interface**

Lors du démarrage du PCD, la ligne série est assignée avec le mode de communication défini. Chaque canal du PCD utilisé doit être assigné au moyen d'une fonction SASI (une seule par canal). Après une coupure de tension ou après un "Départ à froid", l'assignement est active après avoir mis le PCD une fois en RUN (exécution de la routine d'initialisation: XOB 16).

**Paramètres:**

No de Texte: Chaque fonction SASI utilise un texte différent. A choisir dans la plage 0..3999. Attention aux conflits avec des textes éventuellement utilisés dans d'autres fichiers à linker. Vérifiez que le texte n'est pas utilisé par ailleurs dans le programme ou dans tout autre module lié à ce programme.

Canal: No de la ligne série (interface) à assigner du No 0 à 3. Sur PCD2, PCD4 et PCD6.M540, le canal No 0 est destiné à l'outil de programmation (PGU). Aussi longtemps qu'il est assigné par le SASI, il ne peut plus être utilisé pour la programmation, la liaison PGU ne fonctionnera plus.

**Mode:**

Le mode 'Aucun' permet de supprimer l'assignement sans devoir perdre les paramètres déjà définis.

Le mode "Texte" est utilisé pour effectuer un assignement en mode C. Les textes et le programme qui les utilisent doivent être écrit avec l'utilitaire PG3. Les notions de "Maître" et d' "Esclave" ne sont pas significatives. Un assignement est toujours traité en mode MC0.

Les modes D, S-BUS 0 et S-BUS 1 peuvent être utilisés efficacement avec FUPLA grâce aux fonctions de transmission et réception binaires et entiers de la famille 'Communication'.

Avec le mode D (liaison point à point), une station doit être assignée en "Maître" (client) et la deuxième en "Esclave" (serveur). Les fonctions de transmission et réception doivent être programmées exclusivement sur la station "Maître".

Avec les modes S-BUS (liaison multipoint), une seule station doit être assignée en "Maître" et toutes les autres en mode "Esclaves". Les fonctions de transmission et réception doivent être exclusivement utilisées sur la station "Maître". Lorsque le "Maître" est un système de supervision, il joue le rôle de "Maître" et toutes les autres stations PCD sont assignées en "Esclaves". Les fonctions de transmission et réception ne doivent pas être programmés dans des "Esclaves".

Pour plus de détails sur les modes de communication, voir les manuels PCD (guide d'utilisation).

Vitesse de transmission:

Elle doit être la même sur toutes les stations d'un réseau. La vitesse est à choisir selon la qualité du bus et les performances possibles des appareils connectés (pilote PCD, Répéteur, Modem, câble, etc...)

Nombre de bits parité et de bits stop: Le choix doit être identique sur toutes les stations connectées. Dans le mode S-BUS, ces trois paramètres ne sont pas utilisés.

Variables et Registre de diagnostic:

Voir manuel PCD. (Guide des instructions)

Le paramètre online SASI (OK ou Erreur!) indique si, au démarrage du PCD, l'instruction SASI à pu être exécutée correctement ou non. En cas d'erreur, aucune communication ne pourra fonctionner par ce canal.

Les source d'erreur peuvent être:

- Le firmware ne supporte pas ce mode (principalement S-BUS)
- Le CPU n'a pas accès à ce canal
- Le CPU ne supporte pas la (haute) vitesse de transmission principalement avec 38400 bds)
- Le port a été assigné plusieurs fois (éventuellement par un autre programme de lien)
- Une erreur de syntaxe du texte d'assignement. Cette erreur peut arriver avec ce logiciel après modification manuelle du texte du SASI avec le debugger.

### **Debug:**

Les fonctions Debug permettent, dans la station "Maître", de surveiller le fonctionnement de la communication. Lorsque la communication est en Run, les stations et les médias successivement adressés avec leurs diagnostics sont affichés cycliquement à l'écran. Pour un meilleur contrôle des diagnostics et pour analyser un problème de liaison, la communication peut être interrompue par une première pression sur le bouton "Step". Chaque pression supplémentaire déclenchera l'exécution de l'ordre de transmission ou réception suivant. La communication peut être remise normalement en route avec le bouton Run. Les informations 'Station, Média et Adresse' permettent d'identifier la fonction de transmission ou de réception qui est en cours.

**Adjust: Paramètres (SASI)**

Default All Send all OK Info

No de texte	>	0
Canal	>	Canal 1
Mode de communication	>	S-BUS 1
Type de station	>	Esclave
Vitesse de transmission	>	9600 bps
Nombre de bits	>	7 bits
Parité	>	Paire
Bit de stop	>	1 bit

SASI

Buffer de réception

Préselection en réception

Diagnostic en réception

Buffer d'émission

Préselection en émission

Diagnostic en émission

Sortie de texte

Exécution

Debug, station

Debug, élément

Debug, adresse

Debug

Debug

Debug

-----[ Diagnostic ]-----

Registre de diagnostic

0 Débordement

1 Parité

2 Cadrage

3 Signal Break

4 Signature

6 Transmission

7 Dépassement

8 Longueur

9 Format

10 Adresse

12 Echelle

13 Valeur

15 Programme

Run

Step

Vide

Canal 1

Canal 0

Canal 1

Canal 2

Canal 3

S-BUS 1

Aucun

Texte

Mode D

S-BUS 0

S-BUS 1

S-BUS 1/422

Gateway 0

Gateway 1

Esclave

Esclave

Maître

9600 bps

110 bps

300 bps

600 bps

1200 bps

2400 bps

4800 bps

9600 bps

19.2 kbps

38.4 kbps

7 bits

7 bits

8 bits

Paire

Paire

Impaire

Aucune

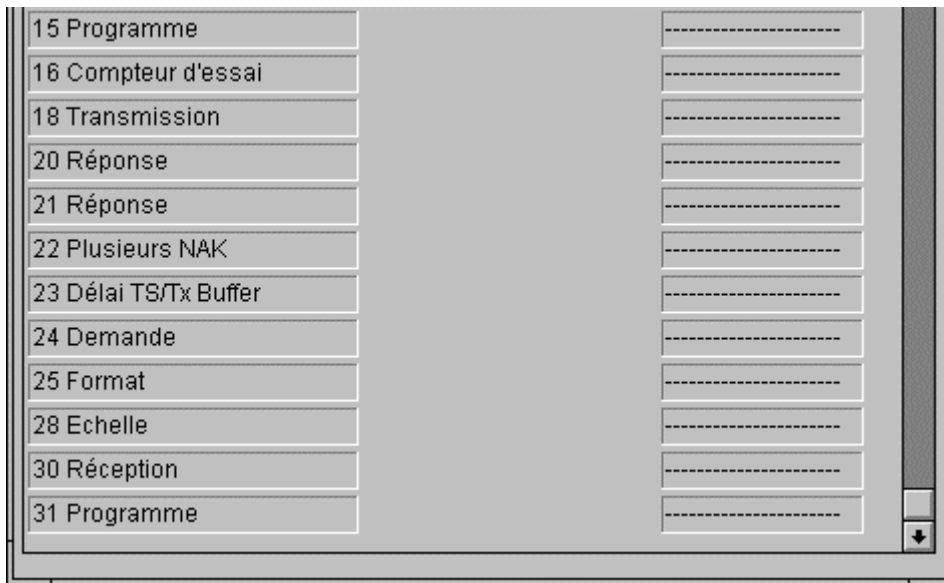
1 bit

1 bit

2 bits

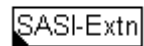
Suite de la fenêtre d'ajustage à la prochaine page.

Suite de la fenêtre d'ajustage



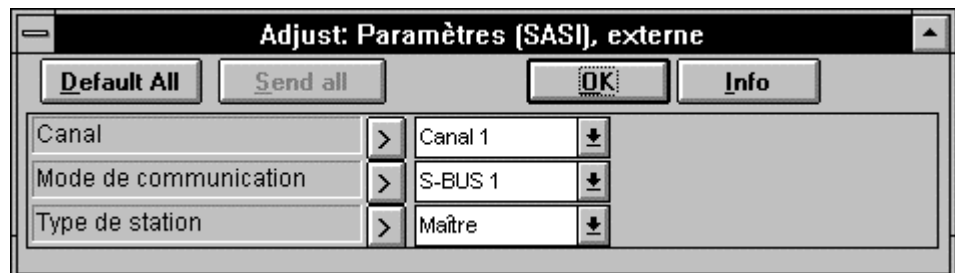
4.4.17.2

Paramètres d'interface série pour des fichiers externe

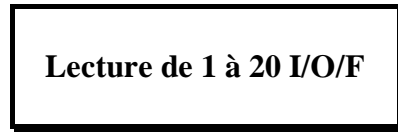


[\_comsase]

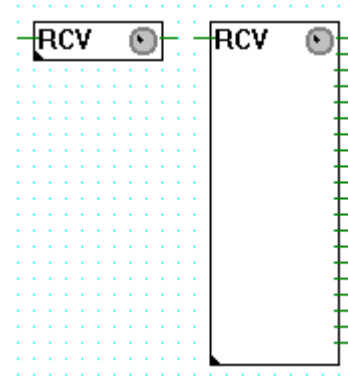
Si l'on désire avoir accès à la même interface série dans deux ou plusieurs fichiers FUPLA, une assignation "normale", dans un fichier (principal), doit être exécutée au moyen de la FBox "Paramètres (SASI)". Dans les autres fichiers, la FBox ci-dessous "Paramètres (SASI) externe" doit être ajoutée et complétée avec les mêmes paramètres que ceux du fichier principal.



**4.4.17.3 Lecture de 1 à 20 I/O/F**



[\_comrxb]



Entrée :        –        Enable

Sorties:        – →     format binaire

Si l'entrée de validation = H (connexion binaire à gauche de la boîte de fonction) alors les valeurs des éléments (Entrée, Sortie ou Variable) de la station distante sont lues. Les éléments de cette station sont définis par les indications à droite de la boîte de fonction.

La fonction permet la lecture de 1 à 20 éléments.

Les informations complémentaires, tels que le numéro du canal, le type des éléments à lire etc. sont précisées dans la boîte de dialogue appropriée.



Lecture de 1 à 20 I/O/F (description détaillée)

Cette fonction ne peut être exécutée que par un maître PCD qui a été assigné par la fonction SASI en mode D ou S-BUS.

Cette fonction fait la lecture de 1 à 20 éléments numériques (I/O/F) d'une station esclave. La réception s'exécute de manière cyclique à la plus grande vitesse aussi longtemps que l'entrée de validation ("connexion à gauche de la boîte de fonction") est haute. S'il s'agit seulement d'un front montant plus court que le temps de l'envoi, une seule requête est faite.

Cette fonction ne doit être utilisée que sur un PCD et une ligne série assignée en mode "Maître" D ou S-BUS, avec la fonction d'assignement de la famille 'Communication'.

Elle permet de recevoir 1 à 20 états binaires d'une station Esclave. La transmission se fait cycliquement le plus rapidement possible lorsque le signal d'activation (première entrée binaire de la fonction) est à 1. Au moins une transmission est faite lors d'un flanc positif sur ce signal, même si l'impulsion est plus courte que le temps de cycle de la transmission sur ce canal.

Principales sources d'erreur :

Manque d'assignement: Le canal sélectionné a été mal assigné (aucune fonction SASI ou SASI invalide). Un assignement par un autre moyen est incompatibles avec la fonction.

Pas "Maître": le canal n'est pas assigné en mode "Maître"

STXM: Une erreur a été détectée lors de l'exécution de l'instruction STXM. Ceci ne devrait pas arriver avec cette fonction sauf si un autre programme accède également à ce canal.

Diagnostic: Une erreur de communication a été détectée. Elle peut être analysée en détail grâce aux diagnostics de la fonction SASI et au moyen du mode Debug.

Step: Le canal a été mis en Step au moyen du mode Debug de la fonction SASI. Aucune transmission n'est exécutée tant que ce mode n'est pas remis en Run.

Paramètres:

Initialisation: Cette option permet d'activer une transmission lors du démarrage du PCD, même si le signal binaire d'activation est à 0. Ceci permet d'initialiser les variables de la station "Maître" après un arrêt de celle-ci.

Canal: Nombre de canaux utilisés.

Station: Nombre de station Esclave de laquelle les données seront reçues. Aucune signification en mode D.

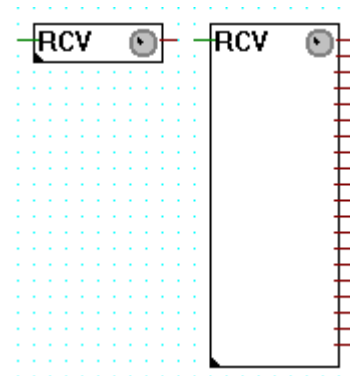
Type et adresse: Élément de base de la station Esclave ou les valeurs à recevoir sont lues. Lorsque l'on transmet plus d'une variable, les éléments suivants sont automatiquement utilisés.

Le bouton 'Ordre' permet de provoquer manuellement une transmission même si le signal d'activation est à 0.

#### 4.4.17.4 Lecture de 1 à 20 R/T/C/Horloge



[\_comrxi]



Entrée: – Enable

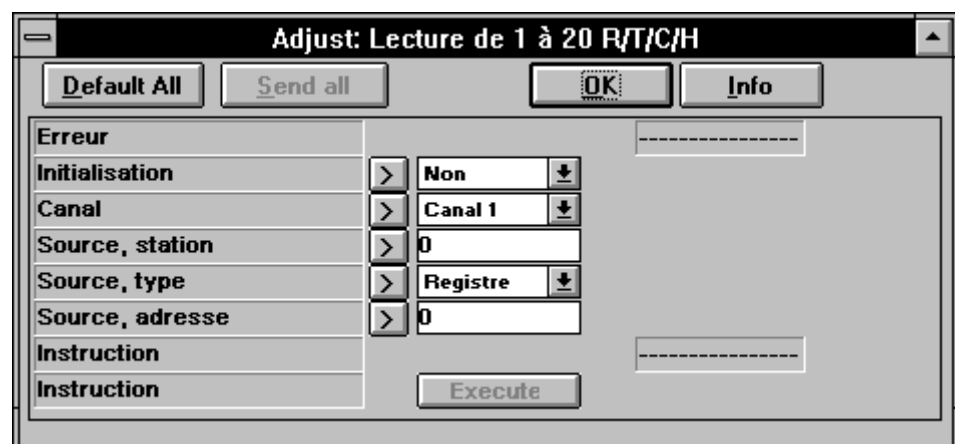
Sorties : = → format entier

Lecture de 1 à 20 R/T/C ou Horloge

Si l'entrée de validation = H (connexion binaire à gauche de la boîte de fonction) alors les valeurs des éléments (Registre, Temporisateur, Compteur ou Horloge) de la station distante sont lues. Les éléments de cette station sont définis par les indications à droite de la boîte de fonction.

La fonction permet la lecture de 1 à 20 éléments.

Les informations complémentaires, tels que le numéro du canal, le type des éléments à lire etc. sont précisées dans la boîte de dialogue appropriée.



Lecture de 1 à 20 R/T/C ou Horloge (description détaillée)

Cette fonction ne peut être exécutée que par un "Maître" PCD qui a été assigné par la fonction SASI en mode D ou S-BUS.

Cette fonction fait la lecture de 1 à 20 éléments numériques (R/T/C ou horloge) d'une station esclave. La réception s'exécute de manière cyclique à la plus grande vitesse aussi longtemps que l'entrée de validation ("connexion à gauche de la boîte de fonction") est haute. S'il s'agit seulement d'un front montant plus court que le temps de l'envoi, une seule requête est faite.

Principales sources d'erreur :

Manque d'assignement: Le canal sélectionné a été mal assigné ou pas du tout (pas de SASI ou SASI invalide). L'assignement par un autre moyen est incompatible avec la fonction.

Pas Maître: Le canal n'est pas assigné en mode "Maître"

STXM: Une erreur a été détectée lors de l'exécution de l'instruction STXM. Ceci ne devrait pas arriver avec cette fonction sauf si un autre programme accède également à ce canal.

Diagnostics: Une erreur de communication a été détectée. Elle peut être analysée en détail grâce aux diagnostics de la fonction SASI et au moyen du mode Debug.

Step: Le canal a été mis en Step au moyen du mode Debug de la fonction SASI. Aucune transmission n'est exécutée tant que ce mode n'est pas remis en Run.

Paramètres:

Initialisation: Cette option permet d'activer une transmission lors du démarrage du PCD, même si le signal binaire d'activation est à 0. Ceci permet d'initialiser les variables de la station "Maître" après un arrêt de celle-ci.

Canal: Nombre de canaux utilisés.

Station: Nombre de stations dont les éléments sont contrôlés. Aucune signification en mode D.

Type et adresse: Élément de base de la station esclave ou les valeurs à recevoir sont lues. Lorsque l'on transmet plus d'une variable, les éléments suivants sont automatiquement utilisés. Pour la lecture de l'horodateur, deux registres sont toujours utilisés. La boîte doit être dimensionnée en conséquence.

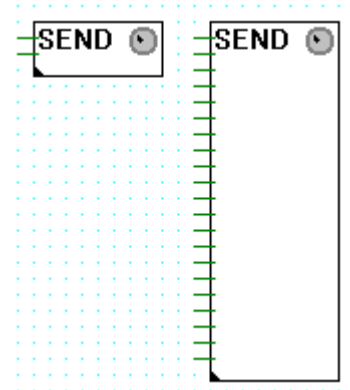
Le bouton 'Ordre' permet de provoquer manuellement une transmission, même si le signal d'activation est à 0.



**4.4.17.5 Ecriture de 1 à 20 I/O/F**



[\_comtxb]



- Entrées:            –        Enable  
                       – →     format binaire

Si l'entrée de validation = H (connexion binaire en haut à gauche de la boîte de fonction), alors les valeurs des éléments (Entrées, Sorties ou Variables) sont transférées depuis cette station à la station distante. Les éléments de cette dernière sont définis par les connexions à droite de la boîte de fonction (2ème à 21ème connexion).

La fonction peut être étendue de 1 à 20 éléments.

Les informations complémentaires, tels que le numéro du canal, le type des éléments destinataires etc. sont précisées dans la boîte de dialogue appropriée.



Ecriture de 1 à 20 I/O/F (description détaillée)

Cette fonction ne doit être utilisée que sur un PCD et une ligne série assignée en mode "Maître" D ou S-BUS, avec la fonction d'assignement de la famille 'Communication'.

Elle permet de transmettre 1 à 20 états binaires vers une station Esclave. La transmission se fait cycliquement le plus rapidement possible lorsque le signal d'activation (première entrée binaire de la fonction) est à 1. Au moins une transmission est faite lors d'un front positif sur ce signal, même si l'impulsion est plus courte que le temps de cycle de la transmission sur ce canal.

Principales sources d'erreur:

Manque d'assignement: Le canal sélectionné a été mal assigné ou pas du tout. (Pas de SASI ou SASI invalide). L'assignement est incompatible avec la fonction.

Pas Maître. Le canal n'a pas été assigné comme Maître.

STXM: Une erreur s'est produite pendant l'exécution de l'instruction STXM. Ceci ne peut pas se produire par l'usage de cette fonction sauf dans le cas où le canal est sollicité par une autre routine.

Diagnostics: Une erreur de communication a été identifiée. Celle-ci peut être détaillée par le diagnostic de la fonction SASI ainsi que par analyse dans le mode "Debug".

Step: Le canal a été mis en pas à pas par le mode "Debug" de la fonction SASI. L'échange ne deviendra automatique qu'après action sur le bouton "Run".

Paramètres :

Initialisation: Cette option permet d'activer une transmission lors du démarrage du PCD même si le signal binaire d'activation est à 0. Ceci permet d'initialiser les variables de la station esclave après un arrêt de la station "Maître".

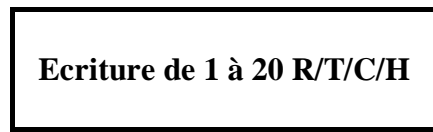
Canal: Nombre de canaux utilisés.

Station: Nombre de stations dont les éléments sont contrôlés. Non significatif en mode D.

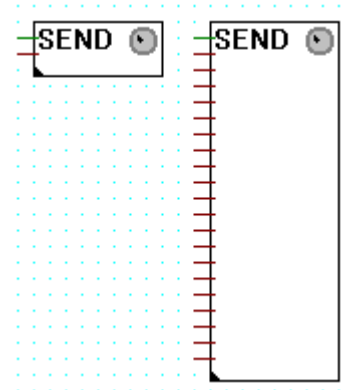
Type et adresse: Élément de base de la station "Esclave" ou les valeurs transmises sont déposées. Lorsque l'on transmet plus d'une variable, les éléments suivants sont automatiquement utilisés.

Le bouton 'Ordre' permet de provoquer manuellement une transmission, même si le signal d'activation est à 0.

#### 4.4.17.6 Ecriture de 1 à 20 R/T/C/Horloge



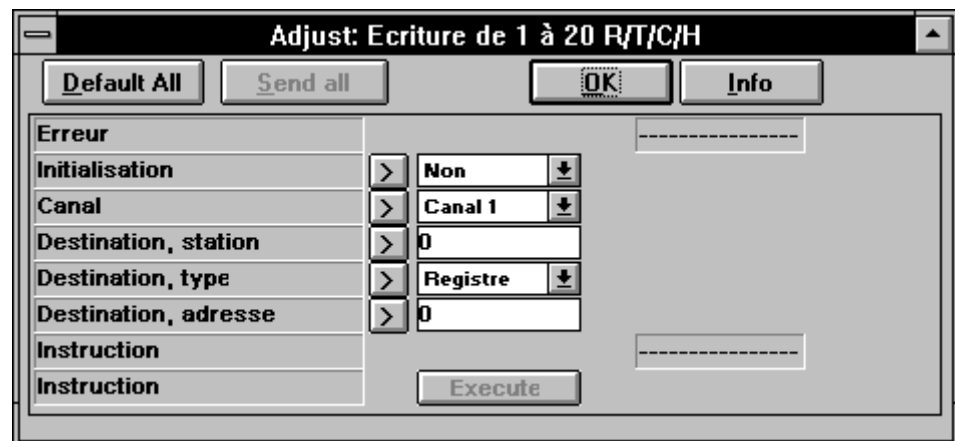
[\_comtxi]



Entrées :        –        Enable  
                   =        format entier

Si l'entrée de validation = H (connexion binaire en haut à gauche de la boîte de fonction), alors les valeurs des éléments (Registres, Temporisa- teurs, Compteurs ou Horloge) sont transférées depuis cette station dis- tante. Les éléments de cette dernière sont définis par les connexions à droite de la boîte de fonction (2ème à 21ème connexion).

La fonction peut être étendue de 1 à 20 éléments.



Ecriture de 1 à 20 R/T/C ou Horloge (description détaillée)

Cette fonction ne peut être exécutée que par un maître PCD qui a été as- signé par la fonction SASI en mode D ou S-BUS.

Cette fonction envoi de 1 à 20 éléments numériques (R/T/C/Horloge) à une station "Esclave". Cet envoi s'exécute de manière cyclique à la plus grande vitesse aussi longtemps que l'entrée de validation (connexion binaire en haut à gauche de la boîte de fonction) est haute. S'il s'agit seulement d'un front montant plus court que le temps de l'envoi, un seul envoi est exécuté.

Principales sources d'erreur:

Manque d'assignement: Le canal sélectionné a été mal assigné ou pas du tout. (Pas de SASI ou SASI invalide). L'assignement est incompatible avec la fonction.

Pas Maître: Le canal n'a pas été assigné comme Maître.

STXM: Une erreur s'est produite pendant l'exécution de l'instruction STXM. Ceci ne peut pas se produire par l'usage de cette fonction sauf dans le cas où le canal est sollicité par une autre routine.

Diagnostics: Une erreur de communication a été identifiée. Celle-ci peut être détaillée par le diagnostic de la fonction SASI ainsi que par analyse dans le mode "Debug".

Step: Le canal a été mis en pas à pas par le mode "Debug" de la fonction SASI. L'échange ne deviendra automatique qu'après action sur le bouton "Run".

Paramètres:

Initialisation: Cette option permet d'activer une transmission lors du démarrage du PCD même si le signal binaire d'activation est à 0. Ceci permet d'initialiser les variables de la station esclave après un arrêt de la station "Maître".

Canal: Nombre de canaux utilisés.

Station: No de la station Esclave ou les données sont transmises. Aucune signification en mode D.

Type et adresse: Adresse de base des éléments de la station esclave à échanger. Si plus d'un élément sont transférés, ils sont stockés dans des adresses successives.

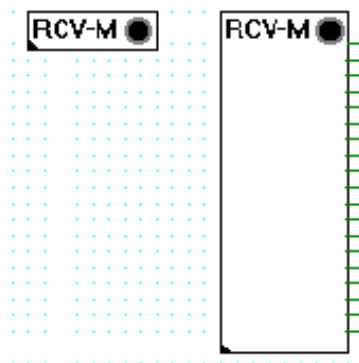
Deux registres sont nécessaires pour les échanges de l'horloge matérielle. La transmission doit tenir compte de cette particularité.

Le bouton 'Ordre' permet de provoquer manuellement une transmission même si le signal d'activation est à 0.

### 4.4.17.7 Lecture I/O/F de plusieurs stations



[\_comrxbm]



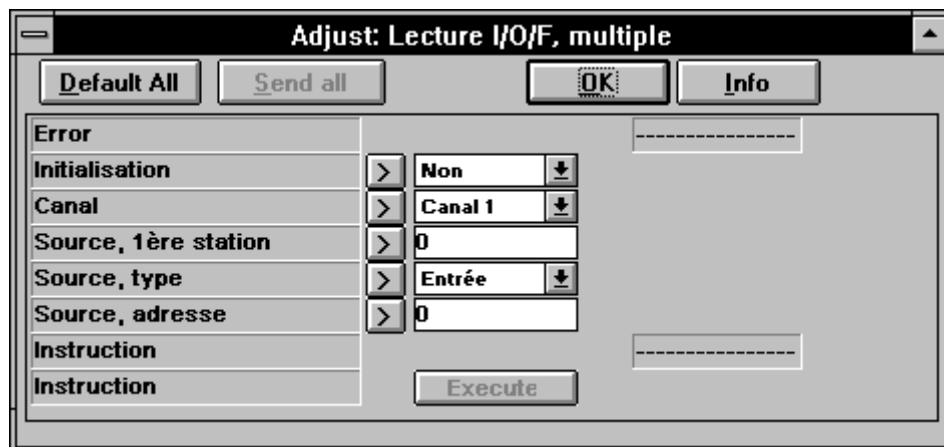
Entrée:            –     Enable

Sorties:           – →   format binaire

Si l'entrée Enable = H (entrée binaire de la Fbox), une valeur (élément binaire = entrée, sortie ou drapeau) est transférée de chaque station partenaire dans cette station.

La fonction peut être étendue de 1 à 20 éléments de sortie (une par station). La dimension de la Fbox définit le nombre de stations à lire. La première station est définie dans la fenêtre d'ajustage.

Les autres réglages, tels que numéro de canal, type d'élément source, etc., sont à exécuter dans la fenêtre d'ajustage.



Cette fonction peut uniquement être exécutée dans un PCD Master qui a été assigné en mode S-BUS Master au moyen de la fonction SASI.

La fonction permet de recevoir des éléments binaires (I/O/F) depuis un nombre (1 jusqu'à 20) de stations esclaves successives. La transmission s'opère de façon cyclique, à la vitesse maximale possible, aussi longtemps que le signal d'activation est haut (entrée binaire). Lorsque l'entrée reçoit un flanc positif, au moins une transmission par station a lieu, même si l'impulsion est plus courte qu'un cycle transmission.

Sources potentielles d'erreur:

L'assignation manque: Le canal sélectionné n'a pas reçu d'assignation, ou une assignation erronée (aucun SASI ou SASI non-valable). Une assignation autre que celle du SASI n'est pas permise.

Pas un master: le canal spécifié n'a pas été assigné comme master

Pas un S-BUS: le canal spécifié n'a pas été assigné en S-BUS. Seul le protocole S-BUS peut utiliser cette fonction.

SRXM: une erreur a été détectée pendant l'exécution de l'instruction SRXM. Ceci ne devrait pas survenir en utilisant cette fonction, à moins qu'une autre routine ait également accès à ce canal.

Diagnostic: Une erreur de communication a été détectée. Elle peut être analysée en utilisant les possibilités de diagnostic de la fonction SASI.

Trop grand: La Fbox a été agrandie de telle façon que le nombre de stations à être lues (première station plus l'index d'agrandissement) est plus grand que 254. Seules les stations 0 à 254 sont autorisées pour les S-BUS Slaves.

Paramètres:

Initialisation: Cette option permet la transmission pendant le démarrage du PCD, même si le signal binaire d'activation est bas. Ceci permet aux éléments d'une station slave d'être initialisés après un arrêt de la station master.

Canal: Numéro de canal à utiliser.

Station: Numéro de la première station slave à partir de laquelle les données sont reçues.

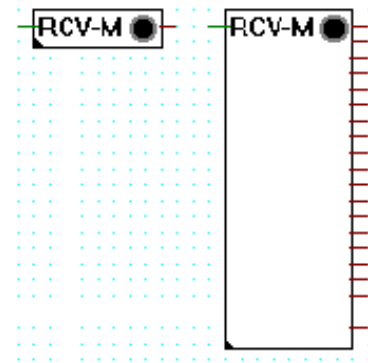
Type et adresse: Adresse de l'élément de la station slave où les éléments doivent être reçus. Un élément par station est transféré.

Le bouton "Instruction" permet d'exécuter une transmission même si le signal d'activation est bas.

#### 4.4.17.8 Lecture R/T/C de plusieurs stations



[\_comrxim]



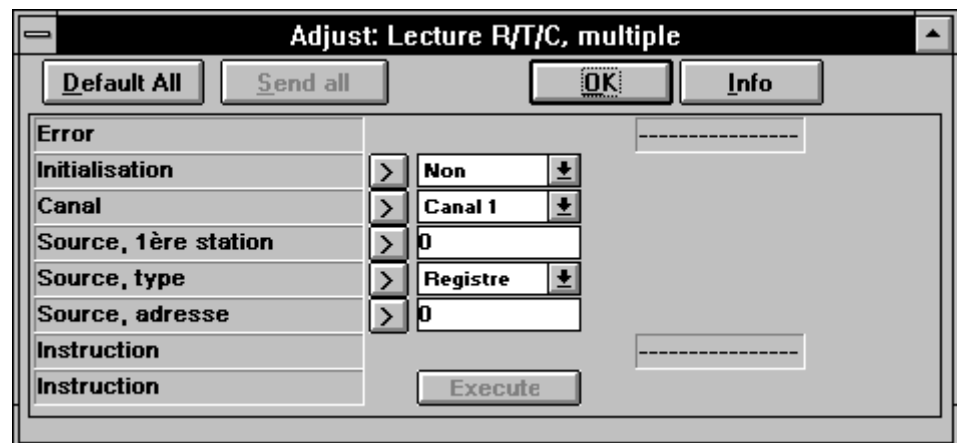
Entrée:            –        Enable

Sortie:            = →     format entier

Si l'entrée Enable = H (entrée binaire de la Fbox), une valeur (élément numérique = registre, tempo ou compteur) est transférée de chaque station partenaire dans cette station.

La fonction peut être étendue de 1 à 20 éléments de sortie (une par station). La dimension de la Fbox définit le nombre de stations à lire. La première station est définie dans la fenêtre d'ajustage.

Les autres réglages, tels que numéro de canal, type d'élément source, etc., sont à exécuter dans la fenêtre d'ajustage.



Cette fonction peut uniquement être exécutée dans un PCD Master qui a été assigné en mode S-BUS Master au moyen de la fonction SASI.

La fonction permet de recevoir des éléments numériques (R/T/C) depuis un nombre (1 jusqu'à 20) de stations esclaves successives. La transmission s'opère de façon cyclique, à la vitesse maximale possible, aussi longtemps que le signal d'activation est haut (entrée binaire). Lorsque l'entrée reçoit un flanc positif, au moins une transmission par station a lieu, même si l'impulsion est plus courte qu'un cycle transmission.

Sources potentielles d'erreur:

L'assignation manque: Le canal sélectionné n'a pas reçu d'assignation, ou une assignation erronée (aucun SASI ou SASI non-valable). Une assignation autre que celle du SASI n'est pas permise.

Pas un master: le canal spécifié n'a pas été assigné comme master

Pas un S-BUS: le canal spécifié n'a pas été assigné en S-BUS. Seul le protocole S-BUS peut utiliser cette fonction.

SRXM: une erreur a été détectée pendant l'exécution de l'instruction SRXM. Ceci ne devrait pas survenir en utilisant cette fonction, à moins qu'une autre routine ait également accès à ce canal.

Diagnostic: Une erreur de communication a été détectée. Elle peut être analysée en utilisant les possibilités de diagnostic de la fonction SASI.

Trop grand: La Fbox a été agrandie de telle façon que le nombre de stations à être lues (première station plus l'index d'agrandissement) est plus grand que 254. Seules les stations 0 à 254 sont autorisées pour les S-BUS Slaves.

Paramètres:

Initialisation: Cette option permet la transmission pendant le démarrage du PCD, même si le signal binaire d'activation est bas. Ceci permet aux éléments d'une station slave d'être initialisés après un arrêt de la station master.

Canal: Numéro de canal à utiliser.

Station: Numéro de la première station slave à partir de laquelle les données sont reçues.

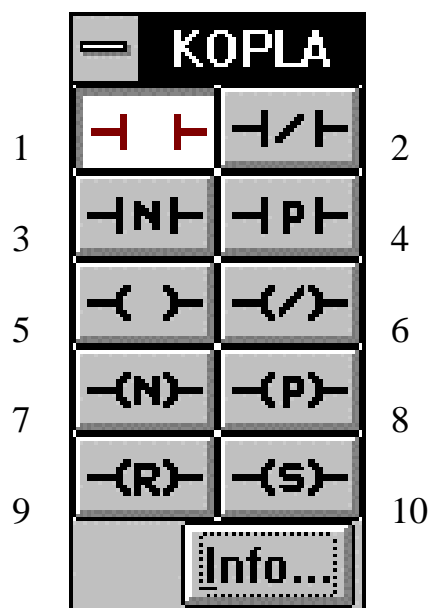
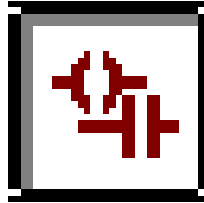
Type et adresse: Adresse de l'élément de la station slave où les éléments doivent être reçus. Un élément par station est transféré.

Le bouton "Instruction" permet d'exécuter une transmission même si le signal d'activation est bas.



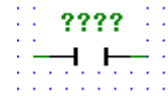
## 4.5 Les fonctions du KOPLA (Ladder Diagram: LD)

L'appel du KOPLA s'effectue à partir de la toolbox du FUPLA et ne pas à partir du menu "Fbox Selection".



Les fonctions du KOPLA sont les suivantes:

- 4.5.1 Contact NO (normalement ouvert)
- 4.5.2 Contact NF (normalement fermé)
- 4.5.3 Contact N (flanc négatif)
- 4.5.4 Contact P (flanc positif)
- 4.5.5 Relais
- 4.5.6 Relais inversé
- 4.5.7 Front N
- 4.5.8 Front P
- 4.5.9 Relais R
- 4.5.10 Relais S

**Plan de contact (Ladder diagram)****(sfupkopl)****4.5.1 Contact NO (normalement ouvert)**

[\_contact]

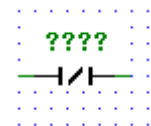
Contact NO: L'ouverture automatique des circuits permettant la commande des actionneurs sera obtenue par des contacts.

Un contact NO est une variable binaire. Il pourra être actionné (par convention, on dira NO = 1), ou non actionné (par convention, on dira NO = 0).

Les contacts seront appelés variables d'entrée. Elles caractérisent les capteurs et organes d'information dans un système automatisé.

L'état de l'armature gauche est copié dans le droite si la variable associée est EN. Dans les autres cas, l'armature droite est HORS.

---

**4.5.2 Contact NF (normalement fermé)**

[\_contact]

Contact NF: L'ouverture automatique des circuits permettant la commande des actionneurs sera obtenue par des contacts.

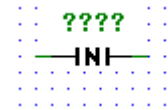
Un contact NF est une variable binaire. Il pourra être actionné (par convention, on dira NF = 1), ou non actionné (par convention, on dira NF = 0).

Les contacts seront appelés variables d'entrée. Elles caractérisent les capteurs et organes d'information dans un système automatisé.

L'état de l'armature gauche est copié dans le droite si la variable associée est HORS. Dans les autres cas, l'armature droite est HORS.

---

### 4.5.3 Contact N (flanc négatif)

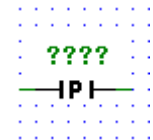


[\_contactnv]

Contact N (Contact de passage): L'état de l'armature droite devient EN quand une transition de EN à HORS est détectée dans la variable associée et que l'armature gauche est EN. L'armature gauche sera HORS dans tous les autres cas.

---

### 4.5.4 Contact P (flanc positif)

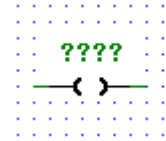


[\_contactps]

Contact P (Contact de passage): L'état de l'armature droite devient EN quand une transition de HORS à EN est détectée dans la variable associée et que l'armature gauche est EN. L'armature gauche sera HORS dans tous les autres cas.

---

## 4.5.5

**Relais**

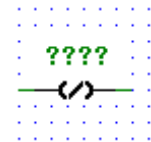
[\_coil]

Relais: Dispositif permettant la commande d'actionneur (variable associée).

L'état logique de l'armature gauche est copié dans l'armature droite. L'état de l'armature gauche est copié dans la variable logique associée.

---

## 4.5.6

**Relais inversé**

[\_coil]

Relais inversé: Dispositif permettant la commande de variable logique.

L'état logique de l'armature gauche est copié dans l'armature droite. L'état inverse de l'armature gauche est copié dans la variable logique associée.

Par ex.: si l'état à gauche est HORS, la variable associée est EN et vice versa.

---

4.5.7

**Front N**



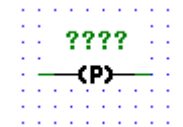
[\_coilnv]

Front N: La variable associée est mise EN entre deux "évaluations", si une transition de EN à HORS est détectée sur l'armature gauche.

L'état de l'armature gauche est toujours copié dans l'armature droite.

4.5.8

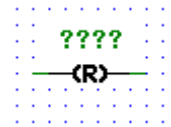
**Front P**



[\_coilps]

Front P: La variable associée est mise EN entre deux "évaluations", si une transition de HORS à EN est détectée sur l'armature gauche.

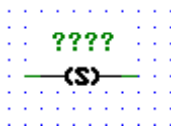
L'état de l'armature gauche est toujours copié dans l'armature droite.

**4.5.9****Relais R**

[\_coilreset]

Relais RESET (bistable): Dispositif permettant uniquement la mise HORS de la variable associée quand l'état de l'armature gauche est EN et remis EN par un relais SET.

---

**4.5.10****Relais S**

[\_coilset]

Relais SET (bistable): Dispositif permettant uniquement la mise EN de la variable logique associée quand l'état de l'armature gauche est EN et remis HORS par un relais RESET.

---

Vos coordonnées :

Société :

Service :

Nom :

Adresse :

Téléphone :

Date :

A renvoyer à :

SAIA-Burgess Electronics SA

Rue de la Gare 18

CH-3280 Morat (Suisse)

<http://www.saia-burgess.com>

DIV. : Electronic Controllers

Les fonctions du FUPLA et du KOPLA

PG4 - Version 1.3

Vos commentaires seront les bienvenus pour améliorer la qualité et le contenu de cette documentation SAIA<sup>®</sup> PCD. Nous vous remercions par avance de votre collaboration.

**Vos commentaires :**