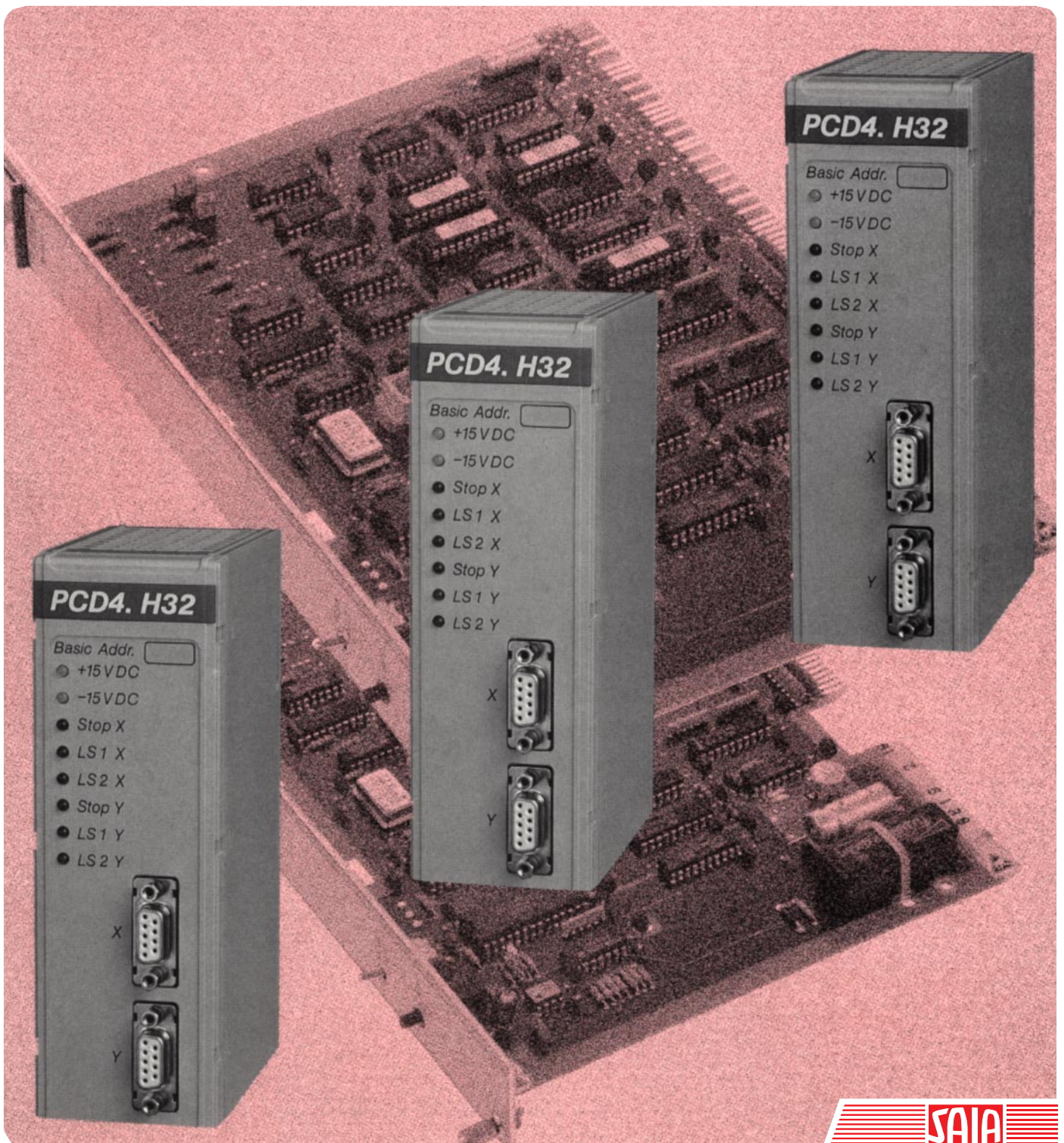


**SAIA®PCD**  
Process Control Devices

**Manuel**  
**Modules de positionnement**  
**pour servo-entraînements**  
**PCD4.H3..**



---

## 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 Opppegård ☎ 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**

# **Modules de positionnement pour servo-entraînements**

**PCD4.H3xx**

SAIA-Burgess Electronics SA 1991. Tous droits réservés  
Edition 26/729 F1 - 04.1991

Sous réserve de modifications

# Mise à jour

---

Manuel : Modules de positionnement pour servo-entraînements PCD4.H3xx - édition F1

Date	Chapitre	Page	Description

## Table des matières

<b>1.</b>	<b>Introduction</b>		
<b>2.</b>	<b>Caractéristiques techniques</b>		
<b>3.</b>	<b>Présentation</b>		
3.1	Circuit imprimé	page	3-1
3.2	Plaque frontale	page	3-3
<b>4.</b>	<b>Schéma-bloc</b>		
<b>5.</b>	<b>Connexions et adressage</b>		
5.1	Connexions	page	5-1
5.2	Adressage	page	5-3
<b>6.</b>	<b>Description du fonctionnement</b>		
6.1	Modes de fonctionnement	page	6-1
6.2	Générateur de profil de vitesse	page	6-2
6.3	Régulateur PID	page	6-4
6.4	Décodeur de position et circuit d'entrée	page	6-5
6.5	Convertisseur D/A (grandeur de réglage analogique)	page	6-11
6.6	Générateur PWM	page	6-12
<b>7.</b>	<b>Ecriture des programmes pour le module H3</b>		
7.1	Installation du logiciel	page	7-1
7.2	Blocs de fonctions principaux "AxInit" et "AxHndlg"	page	7-12
7.3	Vue d'ensemble sur les fonctions	page	7-20
7.4	Description des fonctions	page	7-23
<b>8.</b>	<b>Recherche et traitement des erreurs</b>		
<b>9.</b>	<b>Exemples didactiques d'application</b>		
9.1	Exemple 1	page	9-1
9.2	Exemple 2	page	9-10
9.3	Exemple 3	page	9-16
<b>10.</b>	<b>Vue d'ensemble des commandes et des symboles</b>		

**Notes personnelles :**



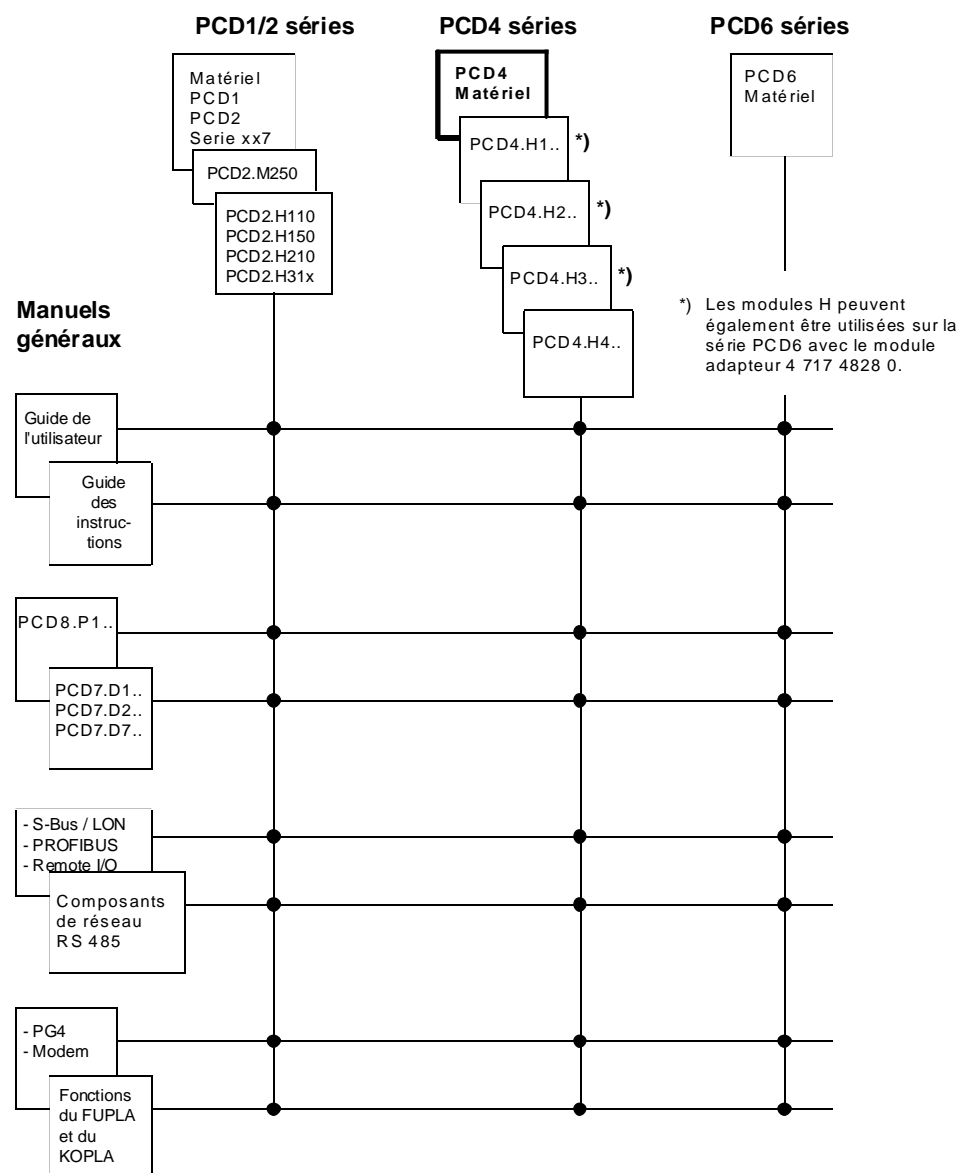
**Avis aux lecteurs :**

De nombreux manuels techniques précis et détaillés ont été élaborés par SAIA-Burgess Electronics SA afin de faciliter l'installation et l'exploitation de ses automates programmables ; ils s'adressent à un personnel qualifié ayant suivi au préalable nos stages de formation.

Pour optimiser les performances des appareils de commande de processus SAIA® PCD, nous vous conseillons de respecter scrupuleusement les consignes de montage, de câblage, de programmation et de mise en service figurant dans ces manuels. Cette démarche rigoureuse vous donnera l'assurance d'une satisfaction totale.

Toutefois, si vous souhaitez formuler des propositions ou des commentaires visant à améliorer la qualité et le contenu de nos documentations, nous vous serions reconnaissants de compléter le formulaire situé en dernière page de cette notice.

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



## **Fiabilité et sécurité des automates programmables**

---

Soucieux d'offrir à sa clientèle des automates programmables fiables et sûrs, SAIA-Burgess Electronics SA apporte le plus grand soin à la conception, au développement et à la fabrication de ses produits.

Parmi ces mesures, citons :

- Technologie de pointe,
- Conformité aux normes,
- Certification ISO 9001,
- Agrément de nombreux organismes internationaux (Germanischer Lloyd, UL, Det Norske Veritas, marquage CE...),
- Choix de composants de haute qualité,
- Contrôles qualité aux différents stades de fabrication,
- Essais en conditions réelles de fonctionnement,
- Déverminage à 85°C pendant 48 heures.

Malgré l'excellence et le grand soin apporté à sa production, SAIA-Burgess Electronics SA ne saurait être tenu responsable des défaillances naturelles d'un composant. A cet égard, les « Conditions générales de vente » exposent clairement les limites de garantie offertes par SAIA-Burgess Electronics SA.

Le responsable de production doit également s'assurer de la fiabilité de son installation ; il lui incombe en effet de se conformer aux spécifications techniques de l'automate sans jamais le soumettre à des conditions extrêmes d'utilisation (respect de la plage de températures, protection contre les surtensions, immunité aux parasites et tenue aux chocs).

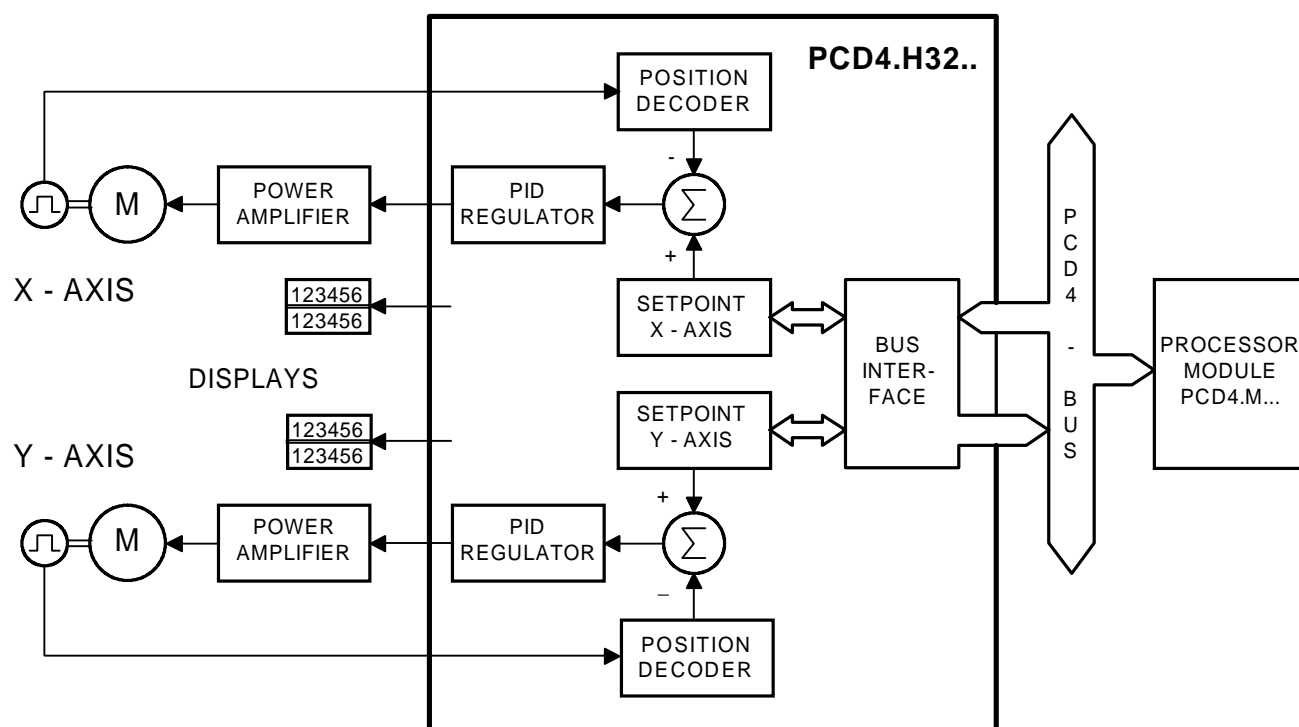
Il lui faut en outre veiller à l'application de toutes les règles de sécurité en vigueur afin de garantir qu'aucun produit défectueux ne risque de porter atteinte à la sécurité des biens et des personnes. Tout défaut générateur de danger doit donner lieu à des mesures complémentaires visant à l'identifier et à en prévenir les conséquences. Ainsi les sorties directement liées à la sécurité de fonctionnement du matériel doivent être raccordées aux entrées et surveillées par logiciel. Il convient enfin de faire systématiquement appel aux fonctions de diagnostic du PCD (chien de garde, blocs d'organisation des exceptions « XOB », instructions de test ou de recherche d'erreurs).

Exploitée dans les règles de l'art, la gamme SAIA® PCD intègre des constituants d'automatismes modernes, alliant sécurité et haute fiabilité, et capables d'assurer pendant des années les fonctions de contrôle-commande, de régulation et de surveillance de votre équipement.



# 1. Introduction

Schéma-bloc d'un servo-entraînement :



## Fonctionnement et application:

Le module de positionnement PCD4.H3.. sert au positionnement d'un ou de deux axes indépendants munis d'entraînements à vitesse de rotation réglable (servo-entraînements). Ces servo-moteurs peuvent être des moteurs réglables DC ou AC capables de déterminer le nombre de tour et la position via un étage de puissance et un détecteur incrémental de vitesse de rotation.

Le module communique avec le CPU du PCD4 via le bus PCD4. Chaque module occupe 16 adresses sur le bus, c'est-à-dire que le même système PCD4 peut théoriquement commander jusqu'à 16 modules de positionnement (32 axes).

Le module dispose pour chaque axe d'un processeur intégré, exécutant de façon autonome un mouvement en fonction des paramètres chargés (vitesse, accélération et destination). Les axes sont commandés indépendamment les uns des autres, c'est-à-dire qu'aucune interpolation n'est possible pour effectuer des trajets courbes. Il est par contre possible de programmer l'enchaînement de plusieurs axes (point-à-point) en mode quasi synchrone.

**Domaines d'application typiques:**

- Robots de manipulation.
- Automates d'équipement et de montage.
- Automates de palettisation.
- Machines d'emballage.
- Machines destinées au travail de la tôle.

**Programmation:**

Une bibliothèque contenant des modules logiciels est mise à la disposition de l'utilisateur. Ces modules fonctionnels se composent d'un programme utilisateur PCD transparent pour l'utilisateur. Le module de positionnement peut ainsi être programmé de manière simple, sans avoir à apprendre un code de commande compliqué.

**Caractéristiques essentielles:**

- Position et vitesse de rotation sont à régulation PID.
- Les paramètres vitesse, destination et PID peuvent être modifiés en cours de mouvement.
- Sortie analogique  $\pm 10$  V ou à modulation d'impulsion en largeur (PWM) pour la commande de l'étage de puissance du moteur.
- Entrées T.O.R. pour contact de référence et de fin de course sous 24 V DC (logique positive).
- Entrées de signaux d'encodage pour 24 V (logique positive ou négative) ou 5 V RS422 (Antivalent Line Driver).
- Sorties T.O.R. pour le raccordement de modules d'affichage PCA2.D14 avec 2 x 6 chiffres par axe.

**Vue d'ensemble des types:**

Désignation	Axes	Grandeur de réglage	Signaux d'encodage
PCD4.H310 <sup>1)</sup>	X	± 10V	24V
PCD4.H320 <sup>1)</sup>	X , Y	± 10V	24V
PCD4.H311 <sup>1)</sup>	X	± 10V	5V (RS422)
PCD4.H321 <sup>1)</sup>	X , Y	± 10V	5V (RS422)
PCD4.H316 <sup>2)</sup>	X	PWM	24V
PCD4.H326 <sup>2)</sup>	X , Y	PWM	24V
PCD4.H317 <sup>2)</sup>	X	PWM	5V (RS422)
PCD4.H327 <sup>2)</sup>	X , Y	PWM	5V (RS422)

1) Fourniture standard

2) Livraison sur demande

**Notes personnelles :**

## 2. Caractéristiques techniques

---

<b>Saisie du déplacement</b>	incrémentale : 2 repères d'impulsion et d'index décalés à 90° (signaux inversés)
Entrées 24 V	
Niveau de signalisation	low = 0.4V high = 19.32V
Courant d'entrée à 24V	10mA
Mode de fonctionnement	logique positive ou négative
Entrées 5 V	entrées antivalentes 5V selon RS422
Séparation galvanique	non
Fréquence de comptage	100kHz
<b>Entrées T.O.R.</b>	par axe : 2 interrupteurs de fin de course 1 interrupteur de référence 24 VDC
Niveau de signalisation	low = 0.4V high = 19.32V
Courant d'entrée à 24 V	10mA
Mode de fonctionnement	logique positive
Filtre d'entrée	100kHz
<b>Sorties T.O.R.</b>	pour la commande d'un module d'affichage PCA2.D14 (2*6 chiffres par axe)
2 sorties	Data et clock (communes pour les deux axes)
1 sortie par axe	Enable (logique inverse : actif bas)
Courant de sortie	1..100 mA (non protégée des courts-circuits) Résistance de charge min 240 ohm à 24 V DC

<b>Sortie de régulation</b>	pour la commande de l'amplificateur servo
Sortie analogique	± 10 V (résolution 12 bits plus signe) Résistance de charge min 3 kohm Protégée des courts-circuits
Sortie PWM	<b>Puls-Width-Modulated</b> Modulation d'impulsion en largeur (résolution 8 bits)  Signaux : SIGN et MAGNITUDE resp. adaptés à la commande directe d'un amplificateur en pont.  Sorties collecteur ouvert : I <sub>max</sub> = 500 mA U <sub>max</sub> = 50 V

**Mode de fonctionnement :** régulation de positionnement  
ou de vitesse de rotation

### Données de positionnement

Position	Unité : µm Plage : $[-2^{30} \dots + (2^{30} - 1)] / k * 10^{-3}$ k = f{résolution d'encodage et rapport de réduction}
Vitesse	Unité : µm/s Plage : $[-2^{30} \dots + (2^{30} - 1)] / k * 22348 * 10^{-6}$ k = f{résolution d'encodage et rapport de réduction}
Accélération	Unité : µm/s <sup>2</sup> Plage : $[-2^{30} \dots + (2^{30} - 1)] / k * 76206 * 10^{-9}$ k = f{résolution d'encodage et rapport de réduction}
Régulateur PID	Temps d'échantillonnage: 341µs Facteur proportionnel, intégral et dérivé programmable (Le temps du balayage est programmable séparément pour la partie différentielle)

**Programmation** Au moyen de modules de fonction reposant sur le programme utilisateur PCD.

### Alimentation

Externe (utilisateur) + 24V DC ( 19V.. 32V ) lissé  
ondulation 10%

Consommation de courant  
à partir d'une alimentation  
externe 24 V

pour H310, 320, 316 et 326  $I_{\max} = 150\text{mA} / \text{axe} + \text{alimentation encodeur}$

pour H311, 321, 317 et 327  $I_{\max} = 300\text{mA} / \text{axe}$   
(charge maximum de l'alimentation encodeur 5 V: 300 mA / axe)

Interne à partir du  
bus PCD4 + 5 V / 120 mA / axe  
 $\pm 15 \text{ V} / 5 \text{ mA}$  par axe (seulement  
pour H310, 311, 320 et 321)

### Conditions d'exploitation

Température ambiante 0°C.. +50°C  
sans ventilation forcée

Immunité aux parasites 1 kV avec couplage capacitif  
selon IEC 801-4

Résistance mécanique selon IEC 65A

Conditions de stockage température : -20°C.. +85°C  
humidité de l'air : 0.. 95 %

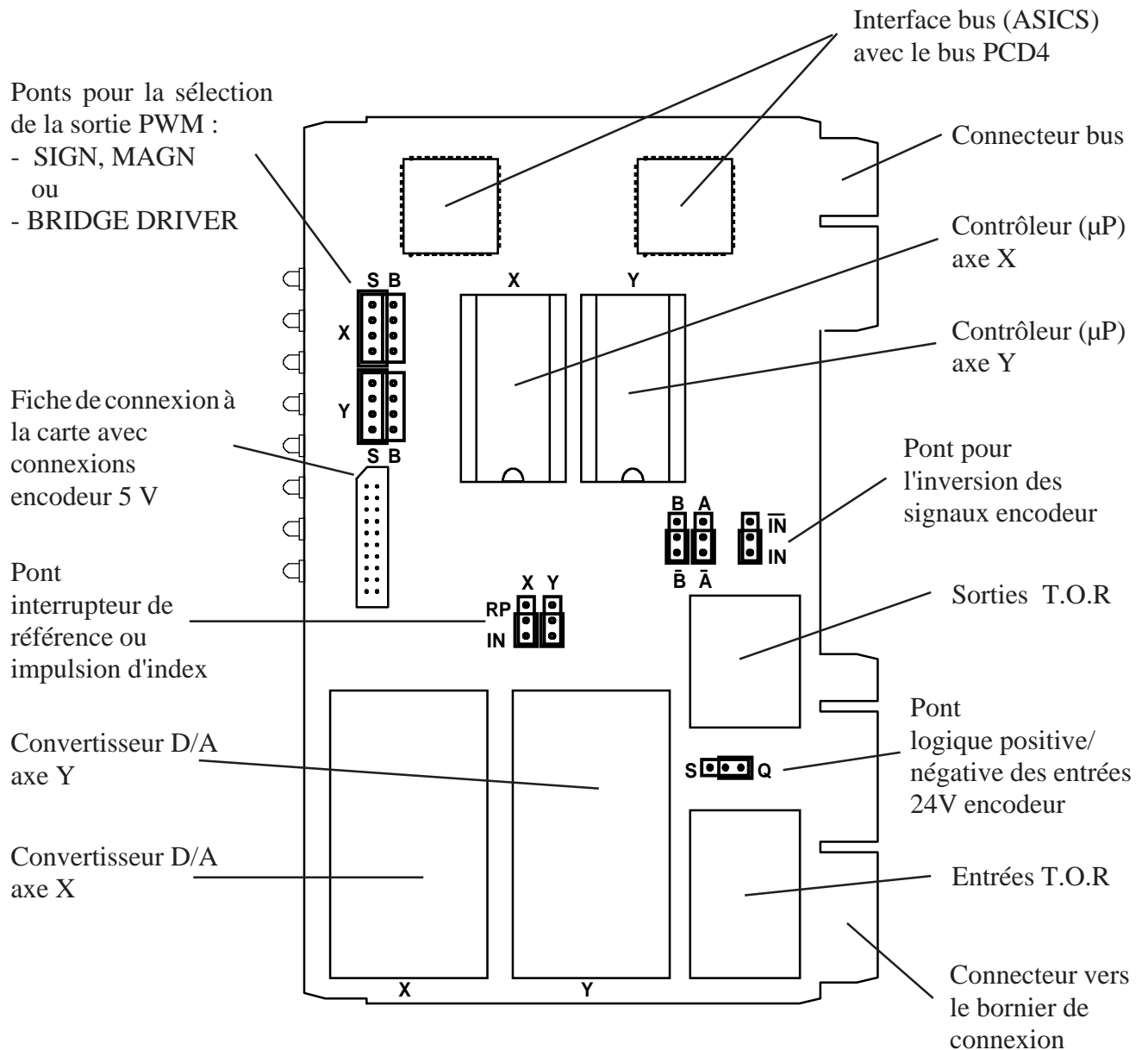
**Notes personnelles :**



## 3. Présentation

### 3.1 Carte et circuits

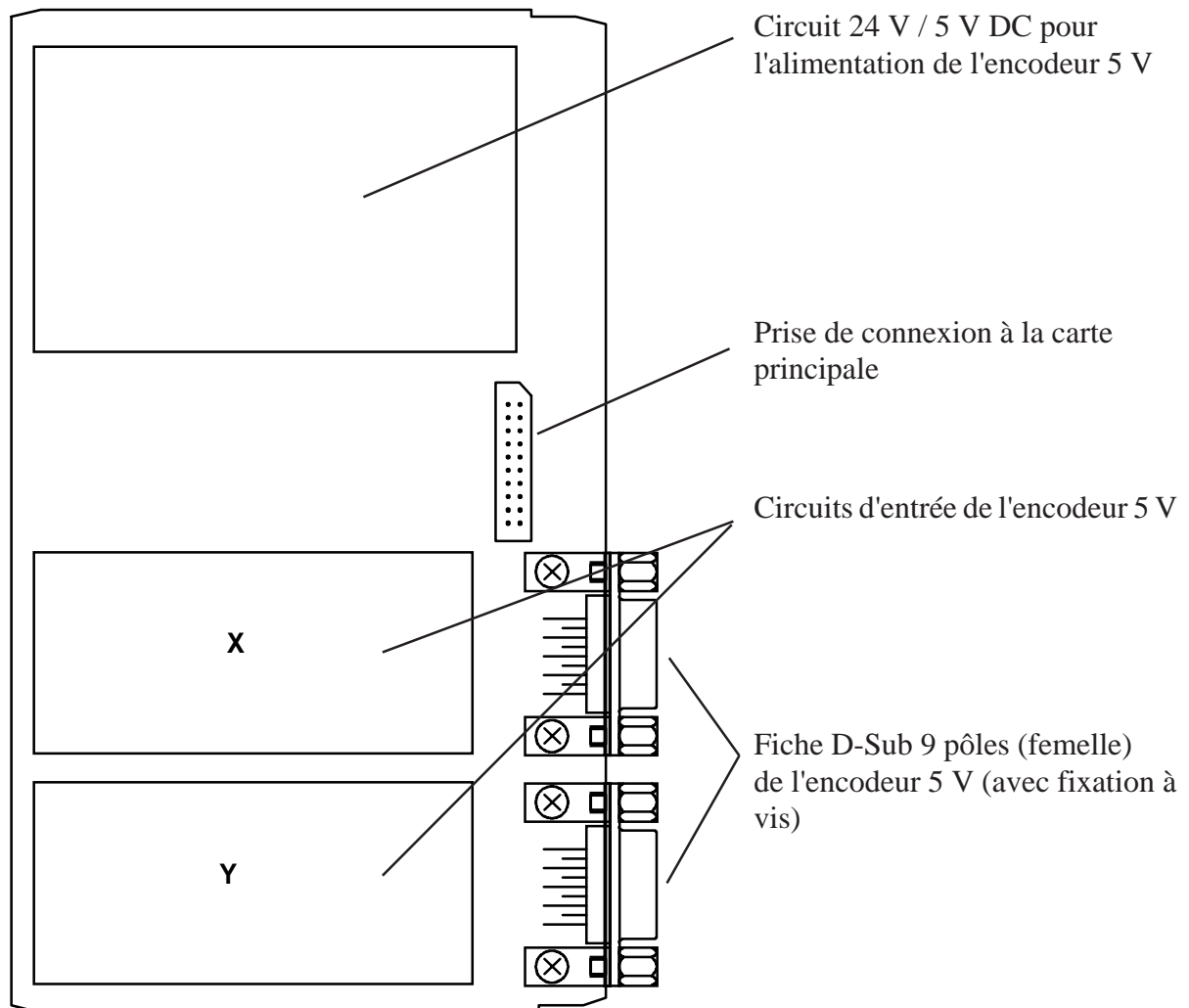
#### 3.1.1 Carte principale (2 axes)



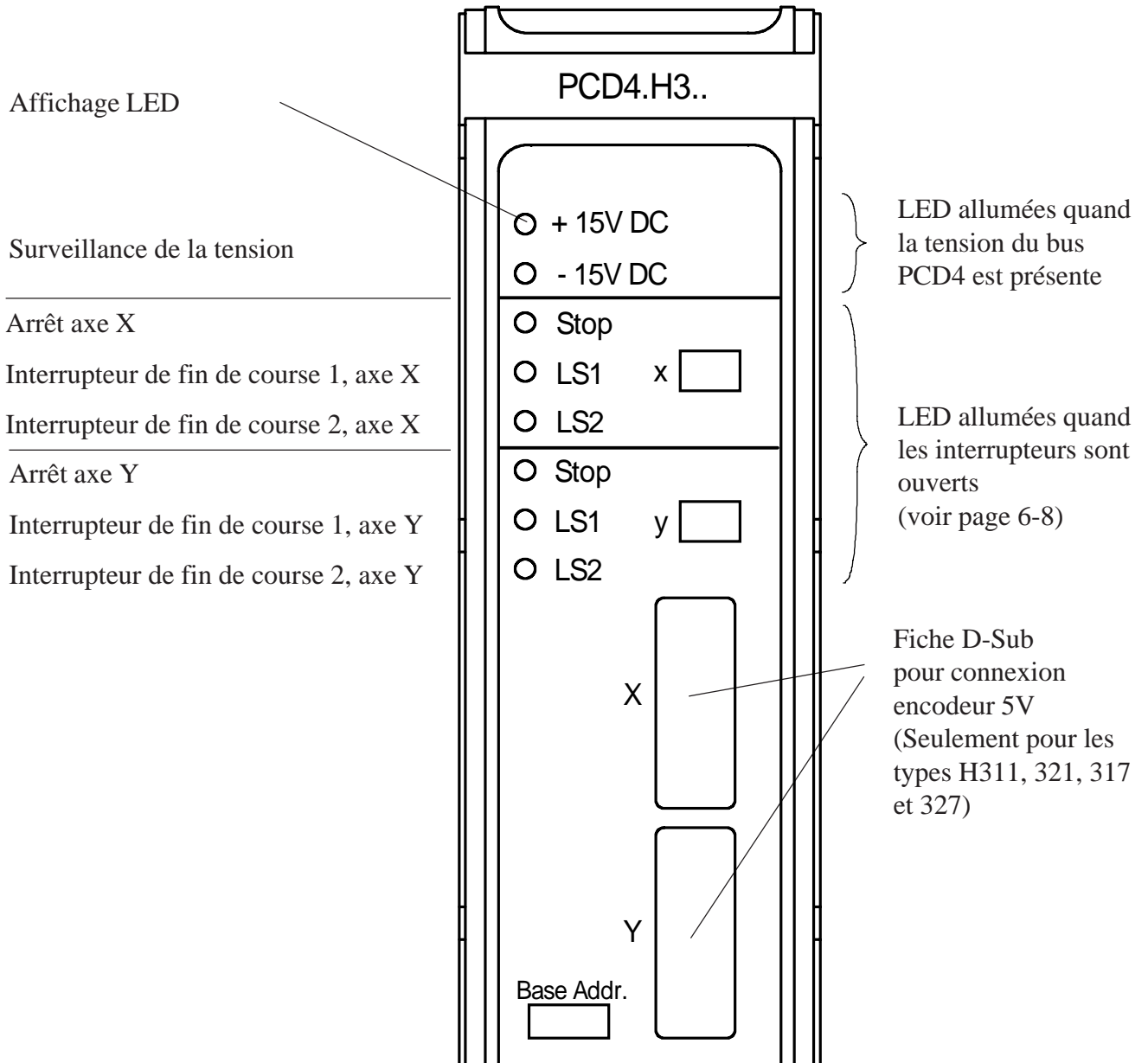
- Les modules convertisseur D/A n'équipent que les types H310, 320, 311 et 321.

- Le pont pour la sélection de la sortie PWM n'équipe que les types H316, 317, 326 et 327.

### 3.1.2 Carte supplémentaire pour l'encodeur 5 V (2 axes)



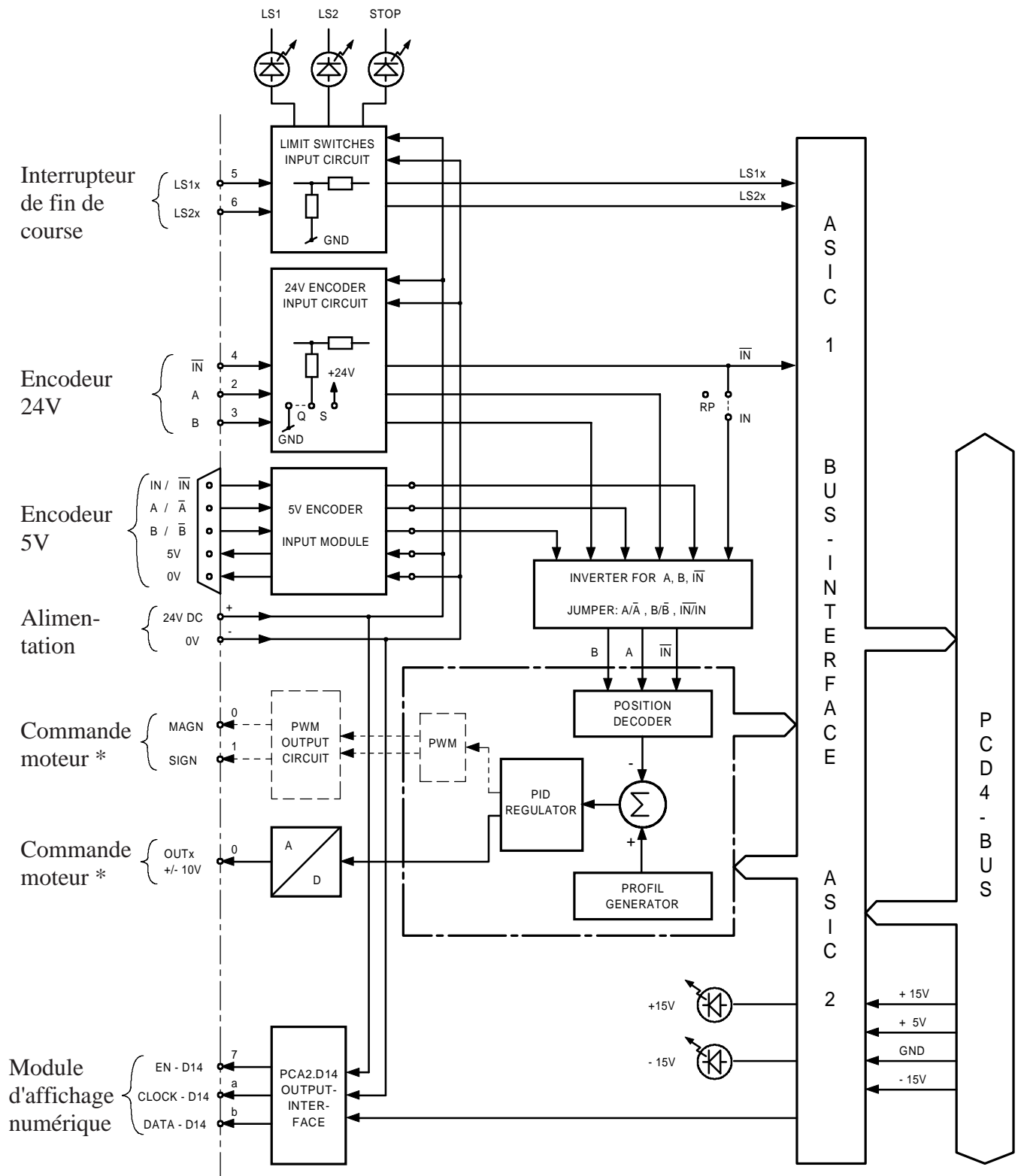
### 3.2 Plaque frontale



**Notes personnelles :**

## 4. Schéma-bloc

Seul l'axe X est représenté

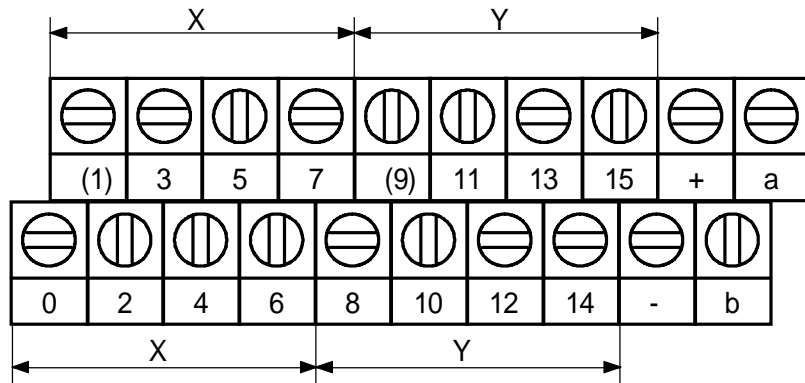


\*) Les sorties vers la commande moteur PWM ou analogique  $\pm 10$  V sont alternatives (voir la vue d'ensemble des types).

**Notes personnelles :**

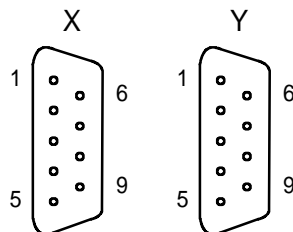
## 5. Connexions et adressage

### 5.1 Connexions



Bornes de connexion (sur le module bus)

0	Out-x	(PWM-MAG <sub>x</sub> )	8	Out-y	(PWM-MAG <sub>y</sub> )
1		(PWM-SIGN <sub>x</sub> )	9		(PWM-SIGN <sub>y</sub> )
2	Phase-Ax		10	Phase-Ay	
3	Phase-Bx		11	Phase-By	
4	/IN <sub>x</sub> (Ref. x)		12	/IN <sub>y</sub> (Ref. y)	
5	LS1 <sub>x</sub>		13	LS1 <sub>y</sub>	
6	LS2 <sub>x</sub>		14	LS2 <sub>y</sub>	
7	EN-D14 <sub>x</sub>		15	EN-D14 <sub>y</sub>	
-	GND		+	+24V	
a	CLK-D14		b	DATA-D14	



Fiche D-Sub (femelle) :

1	PGND	PGND
2	A <sub>x</sub>	A <sub>y</sub>
3	/A <sub>x</sub>	/A <sub>y</sub>
4	B <sub>x</sub>	B <sub>y</sub>
5	GND	GND
6	/B <sub>x</sub>	/B <sub>y</sub>
7	IN <sub>x</sub>	IN <sub>y</sub>
8	/IN <sub>x</sub>	/IN <sub>y</sub>
9	+5V	+5V

## Légende :

Out-x, y	Sortie de la grandeur de réglage $\pm 10$ V resp. PWM-MAGN
PWM-SIGN	Sortie
Phase-A	Entrée encodeur pour la phase A (24V)
Phase-B	Entrée encodeur pour la phase B (24V)
$\overline{\text{IN}}$ (Ref.)	Entrée encodeur pour repère d'index resp. interrupteur de référence
LS1	Entrée pour interrupteur de fin de course (24V)
LS2	Entrée pour interrupteur de fin de course (24V)
EN-D14	Sortie ENABLE pour le module d'affichage PCA2.D14
CLK-D14	Sortie CLOCK pour le module d'affichage PCA2.D14
DATA-D14	Sortie DATA pour le module d'affichage PCA2.D14
+24V	Alimentation +24V DC
+5V	Sortie 5V pour l'alimentation de l'encodeur RS422
GND	Connexion moins pour les alimentations 24 V DC et 5 V DC (masse)
PGND	Connexion de terre de protection pour l'encodeur 5 V. Ne doit pas être raccordée étant donné que le blindage du câble est relié à la terre de protection via le boîtier métallique de la fiche D-Sub et la fixation à vis. Voir aussi le chapitre 6.4.



## 5.2 Adressage

Le module occupe 16 adresses sur le bus PCD4. Etant donné que le module dispose de 2 interfaces de bus (ASIC), les 16 adresses peuvent être utilisées en entrée et sortie.

Signification des 16 adresses :

	Data-In (lecture)	Data-Out (écriture)
	0 Bus données (LSB)	0 Bus données (LSB)
	1 " "	1 " "
	2 " "	2 " "
	3 " "	3 " "
	4 " "	4 " "
	5 " "	5 " "
	6 " "	6 " "
	7 Bus données (MSB)	7 Bus données (MSB)
X	8 —	8 Write (WR)
	9* Limit switch (LS1x)	9 Read (RD)
	10* Limit switch (LS2x)	10 Port select (PS)
	11* In/Ref. switch X-Axis	11* Chip select (1/0=X/Y-Axis)
Y	12 —	12 Clock (PCA2.D14)
	13* Limit switch (LS1y)	13 Data ( " )
	14* Limit switch (LS2y)	14* Enable X axis (PCA2.D14)
	15* In/Ref. switch Y-Axis	15* Enable Y axis (PCA2.D14)

Toutes les adresses indiquées sont des adresses relatives.

Adresse absolue = adresse de base du module + adresse relative

Les adresses marquées \*) présentent un intérêt particulier pour l'utilisateur. Toutes les autres adresses sont exclusivement utilisées par les blocs de fonction.

Le signal Enable pour l'affichage sur le PCA2.D14 est actif bas. Un inverseur pour ces sorties est situé sur le module de positionnement.

**Notes personnelles :**

## 6. Description du fonctionnement

---

### 6.1 Modes de fonctionnement

---

On distingue essentiellement deux modes de fonctionnement :

- MODE DE FONCTIONNEMENT EN REGULATION DE POSITION
- MODE DE FONCTIONNEMENT EN REGULATION DE VITESSE DE ROTATION

#### **Mode de fonctionnement en régulation de position**

Le positionnement s'effectue selon le schéma de commande suivant :

1. Introduction de la position et des paramètres pour le profil de vitesse
2. Démarrage du positionnement
3. Attente du signal "Position cible atteinte"

En mode de positionnement, il y a déplacement régulé vers une position cible après introduction des différents paramètres (facteurs PID, accélération, vitesse, etc.). La vitesse, les facteurs PID et la position cible peuvent être modifiés pendant le mouvement.

#### **Mode de fonctionnement en régulation de la vitesse de rotation**

Schéma de commande :

1. Introduction des paramètres pour le profil de vitesse
2. Démarrage du mouvement
3. Interruption du mouvement en donnant une commande d'arrêt

En mode de régulation de la vitesse de rotation, on accélère jusqu'à la vitesse de consigne à l'accélération donnée. Le déplacement régulé se fait à cette vitesse jusqu'à l'intervention d'une commande d'arrêt. La vitesse de consigne peut être modifiée pendant le mouvement.

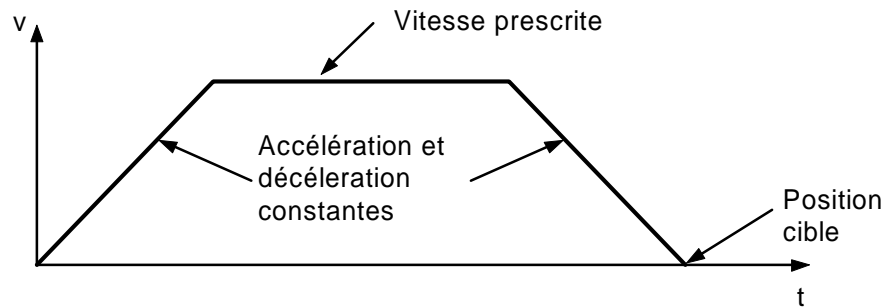
#### **Unités fonctionnelles**

Le schéma bloc montre que le module de positionnement comporte pour l'essentiel les unités fonctionnelles suivantes :

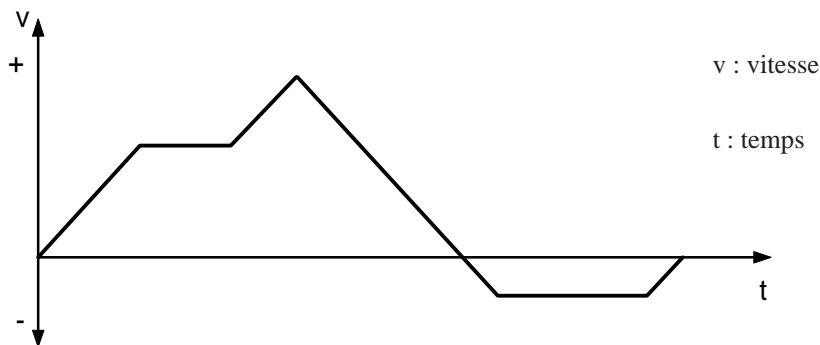
- Générateur de profil de vitesse
- Régulateur PID
- Décodeur de position et circuit d'entrée
- Interface de bus (ASIC) pour le bus PCD
- Convertisseur D/A pour la grandeur de réglage analogique ou générateur pour la modulation d'impulsion en largeur (PWM)

## 6.2 Générateur de profil de vitesse

En fonction de l'accélération et de la vitesse prédéterminées, le générateur de profil calcule la vitesse de consigne en fonction du temps, tant en mode de positionnement qu'en mode de vitesse de rotation. En mode de positionnement, la différence entre la position de consigne et la position actuelle est transmise au régulateur PID. On parvient ainsi à un positionnement très précis du moteur.



Profil de vitesse standard



Profil de vitesse avec consigne de vitesse modifiée et position cible changée pendant le mouvement.

La vitesse et la position cible peuvent être modifiées à volonté pendant le mouvement; le contrôleur provoquera une accélération et décélération avec les paramètres prédéfinis.

En mode de régulation de la vitesse de rotation, le contrôleur accélère jusqu'à la vitesse de consigne définie par l'utilisateur, puis le mouvement se poursuit à vitesse constante jusqu'à l'arrivée d'une commande d'arrêt.

Principe de fonctionnement de la régulation de la vitesse de rotation : La position de consigne est constamment agrandie (en fonction de la vitesse demandée). La différence entre la position de consigne et la position actuelle (définie par l'encodeur) est donnée au régulateur PID. Celui-ci compense immédiatement les variations de vitesse provoquées par les influences perturbatrices en augmentant ou en diminuant la valeur de réglage.

Si le moteur n'atteint pas la vitesse de consigne (par exemple parce que le rotor est bloqué), la différence entre la position de consigne et la position actuelle sera très grande. Ceci provoque l'émission d'un message d'erreur de positionnement qui peut déclencher un arrêt automatique du moteur. Il est possible de programmer la valeur maximum pour l'erreur de positionnement admissible.

### 6.3 Régulateur PID

---

Le moteur peut être amené exactement à la position cible désirée au moyen du régulateur PID; il est maintenu dans cette position car le régulateur est actif tant qu'une commande d'arrêt n'a pas été donnée. Le régulateur utilise l'algorithme de régulation suivant :

$$U(n) = k_p * e(n) + k_i * \sum_{N=0}^n e(n) + k_d * [e(n') - e(n'-1)]$$

où :

- U(n) --> grandeur de réglage pour le moteur
- e(n) --> erreur de position au nème balayage
- n --> balayage partie integrale
- n' --> balayage partie différentielle
- k<sub>p</sub> --> facteur proportionnel
- k<sub>i</sub> --> facteur intégral
- k<sub>d</sub> --> facteur dérivé

Paramètres qui peuvent être programmés par l'utilisateur :

- facteur de régulation k<sub>p</sub>, k<sub>i</sub>, k<sub>d</sub>
- temps d'échantillonnage dérivé
- limite d'intégration (IL) pour la partie intégrale

**Les facteurs de régulation k<sub>p</sub>, k<sub>i</sub> et k<sub>d</sub>** peuvent être modifiés pendant l'accomplissement d'un mouvement. **Le temps d'échantillonnage pour la partie proportionnelle et intégrale** est de 341 µs. Ceci signifie que la grandeur de réglage est mise à jour à un intervalle de temps de 341µs ! **Le temps d'échantillonnage partie différentielle** peut être réglé par pas de 341 µs (au maximum 256 \* 341 µs). Il faut choisir un temps d'échantillonnage plus grand en fonctionnement avec de faibles vitesses.

**Limite d'intégration IL :**

la valeur de l'expression  $k_i * \sum_{N=0}^n e(n)$  est limitée.

## 6.4 Décodeur de position et circuit d'entrée

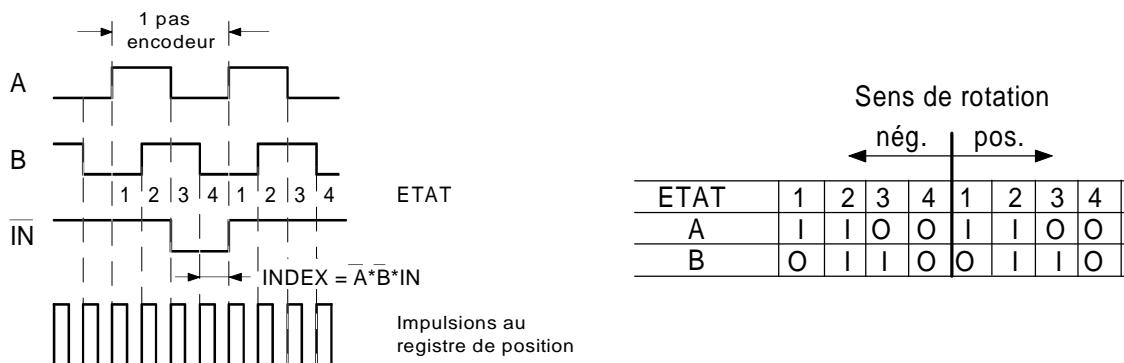
### Saisie de la position et de la vitesse

La position exacte et/ou la vitesse de rotation du moteur est saisie par un encodeur incrémental. Les signaux d'encodage suivant peuvent être raccordés :

PCD4 .H310	}	A,B,IN ou $\bar{A},\bar{B},\bar{IN}$ (connexions aux bornes) signaux 24 V en logique positive ou négative
" .H320		
" .H316		
" .H326		
PCD4 .H311	}	A, $\bar{A}$ ; B, $\bar{B}$ ; IN, $\bar{IN}$ ; (fiche frontale D-Sub) Entrées 5V RS 422 (Antivalent Line Driver)
" .H321		
" .H317		
" .H327		

### Entrées A, B, $\bar{IN}$ :

Diagramme d'état des signaux A,B, $\bar{IN}$  au décodeur de position :



### Entrées A, B :

Le registre de position interne est augmenté ou diminué de 1 lors de chaque changement d'état (0 -> 1 ou 1 -> 0) des signaux A et B. On obtient ainsi une quadruple résolution du pas encodeur. La donnée introduite pour la position cible doit en conséquence aussi être multipliée par quatre. Les signaux doivent se présenter au décodeur de position exactement dans l'ordre indiqué dans le schéma ci-dessus. Si l'encodeur émet d'autres signaux, ceux-ci doivent être inversés de façon appropriée au moyen des ponts (voir les pages suivantes).

**Entrée  $\overline{IN}$  :**

L'entrée  $\overline{IN}$  peut être utilisée avec les modules pour encodeur 24 V (types H310, 320, 316 et 326) en tant qu'entrée d'impulsion d'index (signal zéro en provenance de l'encodeur) ou en tant que point de référence.

- Utilisation en tant qu'entrée d'impulsion d'index :  
(Pont IN/RP en position IN)

Chaque fois que les 3 signaux d'encodage sont à l'état zéro et que le bloc fonctionnel "SetIP" (Set Index Position) a été préalablement appelé, la position absolue du moteur est écrite dans le registre de position d'index.

- Utilisation en tant qu'entrée de point de référence :  
(Pont IN/RP en position RP)

Un interrupteur de référence peut être raccordé par exemple pour définir la position zéro. Pour cette application, le jumper IN/RP doit se trouver en position RP afin que l'entrée ne soit plus active en combinaison avec celles du décodeur de position.

Dans le cas de l'utilisation des modules H310, 320, 316 ou 326 et de l'entrée  $\overline{IN}$  en tant que signal d'index, l'interrupteur de référence doit être connecté à une entrée d'un module d'entrée T.O.R (p. ex. E100).

**Modules pour le raccordement de signaux d'encodage 5 V**

Ils sont équipés d'une carte supplémentaire où est générée l'alimentation 5 V et où les connexions d'encodeur sont amenées sur la fiche D-Sub. Lors de l'utilisation des modules H311, 321, 317 ou 327, il est possible de raccorder aussi bien un interrupteur de référence (aux bornes) que le signal d'index (fiche D-Sub).

Comme on le voit sur le schéma d'entrée suivant, il faut utiliser un câble blindé pour établir la liaison avec l'encodeur 5 V.

- Longueur maximum du câble : 20m
- Section minimum du conducteur : 0.25mm<sup>2</sup>

Afin d'assurer la protection indiquée contre les perturbations, il faut utiliser une fiche D-Sub à boîtier métallique complet, ce qui assure la liaison directe avec la terre de protection (PGND) du système PCD4.



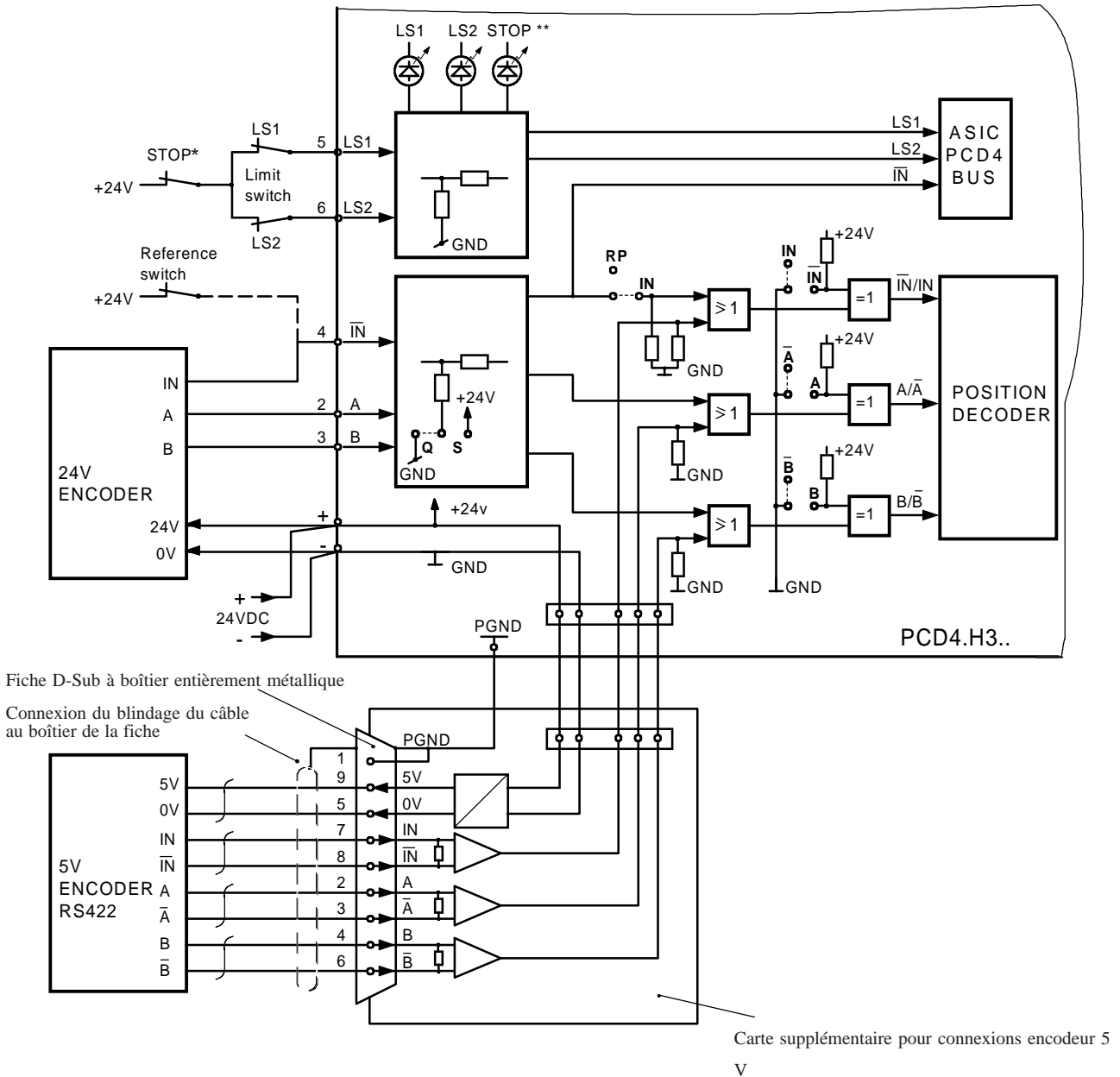
### **Interrupteur de fin de course, interrupteur de référence**

La connexion pour les interrupteurs de fin de course (entrées LS1 et LS2) et l' interrupteur de référence (entrée IN) se fait en logique positive sous 24 VDC. Les signaux de ces interrupteurs sont amenés directement au bus PCD4. Ceci signifie qu'ils doivent être surveillés par le programme utilisateur afin de déclencher les actions nécessaires.

Ces interrupteurs de fin de course (LS1/2) et un interrupteur d'arrêt éventuellement commuté en série **ne doivent assumer aucune fonction de déclenchement dans le sens des prescriptions de sécurité**. Pour ce faire, il faut prévoir des interrupteurs de fin de course de sécurité et des interrupteurs d'arrêt d'urgence supplémentaires qui interviennent directement sur le circuit principal des entraînements.

### Schéma d'entrée et connexions

(Un seul axe est représenté)



\*) Pour les aspects de sécurité, voir le paragraphe précédent "Interrupteurs de fin de course, interrupteur de référence"

\*\*\*) Tableau des états des LED (LS1, LS2 et Stop)

ENTREES		LED		
LS1	LS2	LS1	LS2	STOP
24V	24V	éteint	éteint	éteint
ouvert	24V	allumé	éteint	éteint
24V	ouvert	éteint	allumé	éteint
ouvert	ouvert	éteint	éteint	allumé

**Sélection des ponts** (voir aussi le chapitre 3.1, carte et circuits) :

Pont	Fonction	Position départ usine
Q/S <sup>1)</sup>	logique positive ou négative entrées A,B, $\overline{IN}$	logique positive (Q)
IN/RP <sup>2)</sup>	entrée pour impulsion d'index ou point de référence	impulsion d'index (IN)
A/ $\overline{A}$ <sup>3)</sup>	inversion signal de phase A	$\overline{A}$
B/ $\overline{B}$ <sup>3)</sup>	inversion signal de phase B	$\overline{B}$
IN/ $\overline{IN}$ <sup>3)</sup>	inversion impulsion d'index $\overline{IN}$	IN

} seulement pour encodeur 24V

} pour tous les encodeurs 24 V et 5 V

1) La commutation logique positive / logique négative des signaux 24 V se fait au moyen d'un seul pont pour les deux axes.

2) Pour les encodeurs 24 V, on peut raccorder aux bornes n°4 (axe X) et n°12 (axe Y) :

- signal d'index de l'encodeur ou
- interrupteur de référence

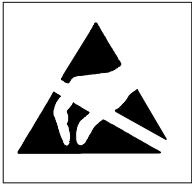
Dans le cas des encodeurs 5 V, les signaux d'index sont amenés via la fiche D-Sub, c'est-à-dire que les bornes 4 et 12 sont libres pour l' interrupteur de référence (pont en position "RP"). Le pont "IN/RP" situé sur la carte est toujours sélectionnable séparément pour chaque axe.

3) L'inversion de tous les signaux d'encodage (24 V et 5 V) se fait chaque fois avec un pont pour A, B et IN, également en commun pour les deux axes.

**Ouverture du boîtier du module pour le changement de place des ponts**

Afin de pouvoir modifier la position des ponts, il faut extraire la carte du boîtier du module. Ceci se fait en pressant les encliquetages latéraux du couvercle frontal. Il faut ensuite dévisser la vis de fixation de la carte qui se trouve sur le côté supérieur gauche du module; on peut alors extraire la carte.

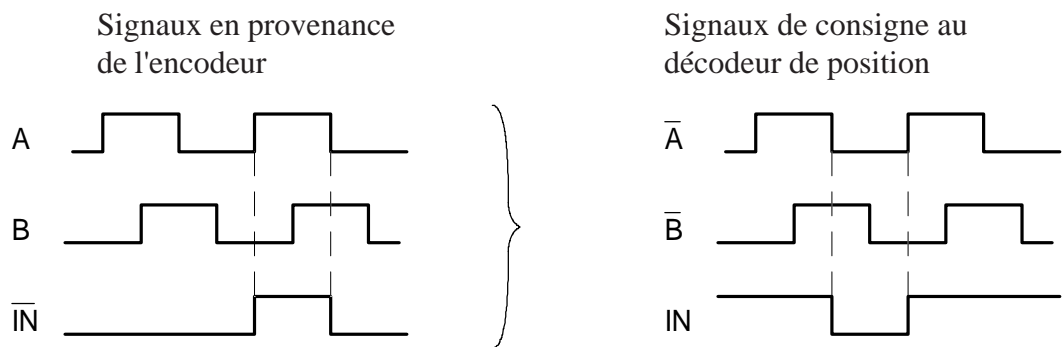
Après avoir correctement mis en place les ponts, refermer le boîtier et revisser la vis de fixation de la carte.



**Attention:** il y a des composants sensibles aux décharges électrostatiques tant sur la carte de base que sur les modules analogiques.

### Exemple d'application pour le choix des ponts :

Un encodeur fournit les signaux 24 V suivant en logique positive :



Afin que les signaux parviennent à l'entrée du décodeur de position dans l'ordre désiré, les trois signaux doivent être inversés.

Jumper :  $A/\bar{A} \rightarrow \bar{A}$   
 $B/\bar{B} \rightarrow \bar{B}$   
 $\bar{IN}/IN \rightarrow IN$  } ceci correspondant à la position ex usine

### Important :

Si l'on travaille (décodeurs 24 V) avec les **interrupteurs de référence** (jumper "IN/RP" en position RP) il faut aussi que le jumper  $\bar{IN}/IN$  soit mis en position "**IN**".

Motif : le signal d'index  $\bar{IN}$  ne doit pas rester à zéro en permanence à l'entrée du décodeur de position car ceci pourrait conduire à un mauvais fonctionnement du contrôleur dans le cas de vitesses de rotation élevées.

## 6.5 Convertisseur D/A (grandeur de réglage analogique)

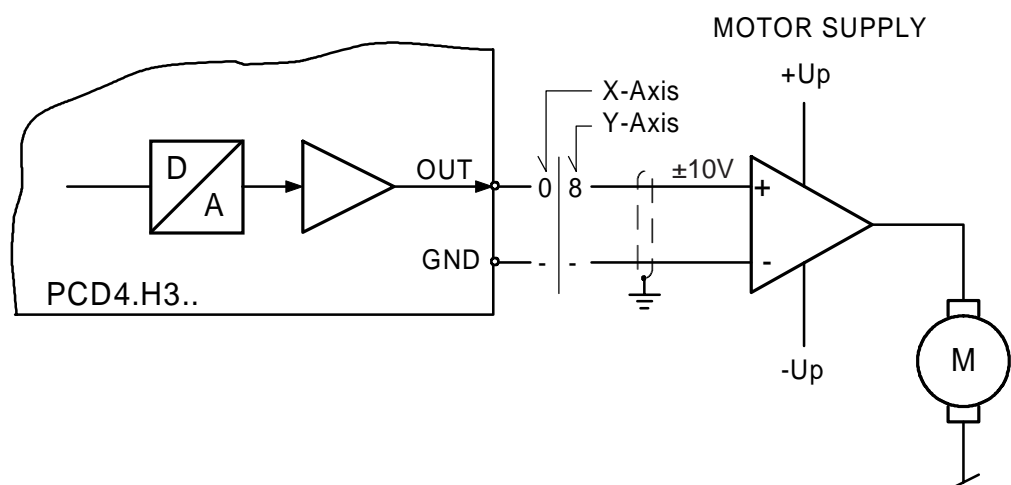
Modules: PCD4.H310  
" .H320  
" .H311  
" .H321

} disposent d'une sortie analogique pour la grandeur de réglage moteur

Un convertisseur D/A 12 bits est utilisé pour chaque axe.

### Raccordement de la sortie analogique :

(un seul axe est représenté)



## 6.6 Générateur PWM

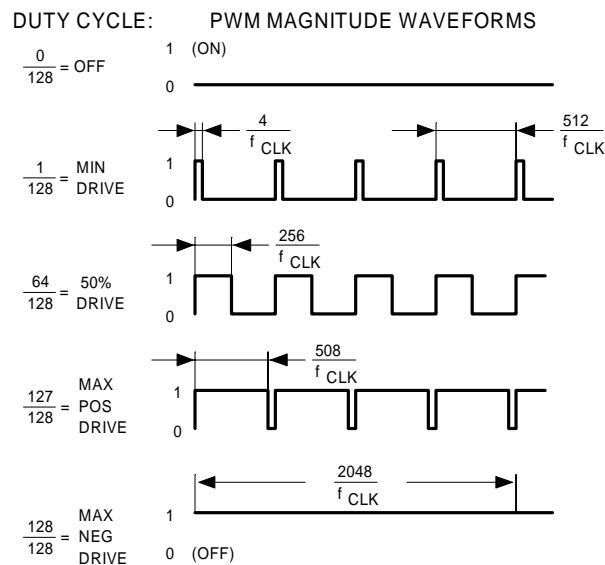
Modules: PCD4 .H316 } disposent d'une sortie PWM avec les  
 " .H326 } signaux SIGN et MAGNITUDE, ou  
 " .H317 } d'une logique de sortie pour la  
 " .H327 } commande d'un amplificateur en pont.

La logique ci-dessus mentionnée est intégrée dans beaucoup d'amplificateurs de puissance. C'est pourquoi on a la possibilité d'émettre directement ou via une logique les signaux SIGN et MAGNITUDE au moyen de la sélection des ponts.

### Signal PWM à la sortie du générateur PWM : (sans le signal SIGN)

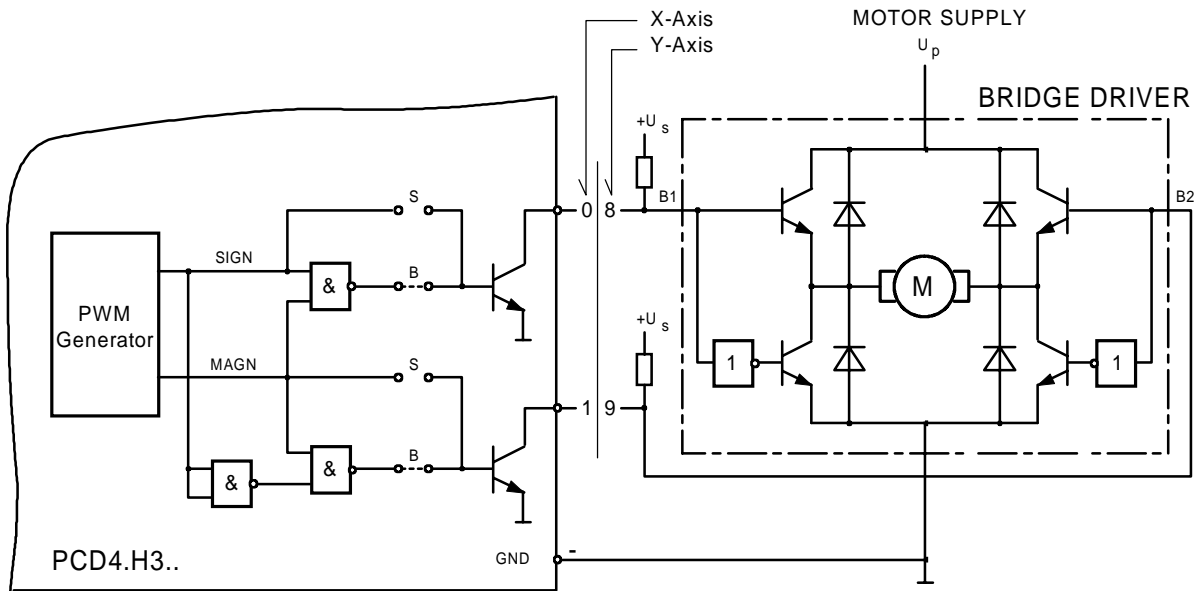
La Résolution du signal est de 8 bits.

Plage: -128 .... +127

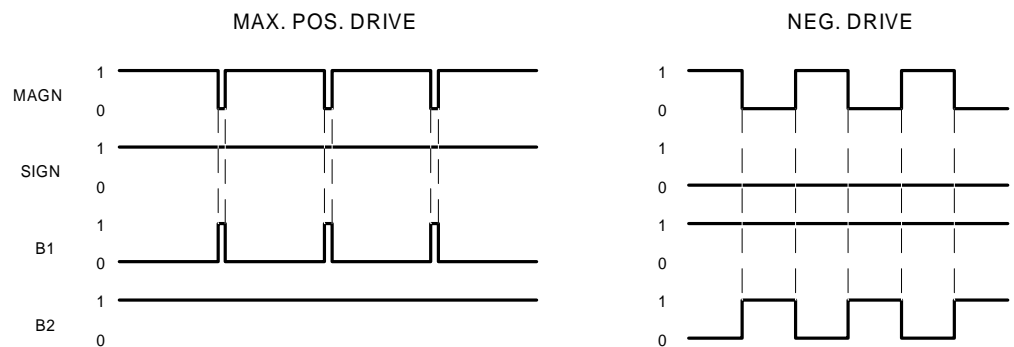


### Schéma-bloc de la sortie PWM

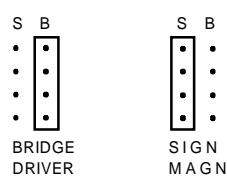
(un seul axe est représenté)



Le diagramme suivant montre la relation entre les signaux SIGN et MAGN et la sortie en vue de la commande directe d'un amplificateur en pont.



Sélection des ponts:



Les sorties pour les axes X et Y sont sélectionnables séparément

Position ex usine : SIGN, MAGN (pos. S)

**Notes personnelles :**



## 7. Ecriture des programmes pour le module H3

---

### 7.1 Installation du logiciel

---

#### 7.1.1 Le paquet logiciel PCD9.H3E1

(PCD9.H3E1 pour disquettes 5 1/4" ou PCD9.H3E6 pour disquettes 3 1/2"). L'utilisateur du module H3 dispose du paquet logiciel PCD9.H3E1 pour la préparation et l'utilisation du module. Ce paquet contient des blocs de fonction (FB) écrits en langage PCD utilisateur. En faisant appel à ces blocs de fonction dans le programme utilisateur, le module H3 est mis en fonction.

Le paquet contient les deux fichiers suivants :

**H3DEF.SRC** Ce fichier contient toutes les déclarations de symboles de l'ensemble du logiciel H3. L'installation H3 est configurée par ces déclarations..

**H3FB.SRC** Ce fichier contient les blocs de fonction destinés à l'utilisation du module H3.

Le paquet logiciel a la taille suivante :

- nombre de lignes de programme  $\leq 1250$
- niveaux d'imbrication max. des FBs 6

#### 7.1.2 Assemblage et linkage des fichiers

Il existe en principe deux possibilités d'assemblage et de linkage des fichiers H3. L'utilisateur travaille avec des définitions de symboles "externes", ou bien le fichier de définition des symboles est inclus au moyen de l'instruction assembleur "\$INCLUDE". Les fichiers H3 sont déjà préparés pour ces deux manières de travailler. En fonction de la méthode retenue, il suffit simplement de déterminer dans les fichiers H3 les instructions conditionnant l'assemblage.

Par défaut, les instructions sont préparées pour l'assemblage sans affectation externe de symboles.

### Travail sans affectation de symboles externes

Le fichier de définition des symboles H3DEF.SRC doit être inclus dans le programme utilisateur au moyen de "\$INCLUDE". Les déclarations dans les fichiers H3 doivent être définies (EQU) comme suit :

```
- H3DEF.SRC : PUBLSYM      EQU 0
- H3FB.SRC  : EXTNSYM     EQU 0
```

Inclusion du fichier H3 dans le programme utilisateur :

Fichier programme utilisateur (par ex. USER.SRC)

```

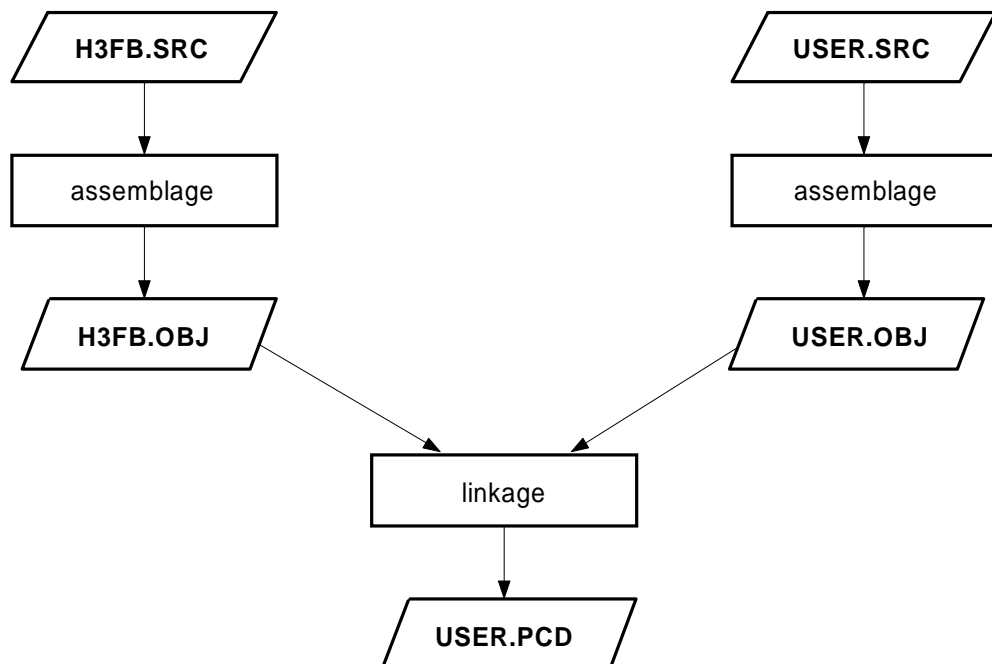
Debut du fichier (titre, en-tête, etc.)

$INCLUDE      H3DEF.SRC
$INCLUDE      autres fichiers à
              inclure
"
"
Programme utilisateur
"
"
Fin de fichier

```

Le fichier de déclaration des symboles H3 doit être inclus avant tous les autres fichiers utilisateur (qui peuvent utiliser les symboles contenus dans H3DEF.SRC).

Le diagramme ci-dessous montre comment les fichiers sont assemblés et liés.



### Travail avec des affectations de symboles externes

Tous les fichiers sont assemblés séparément, puis linkés. •

Les instructions dans les fichiers H3 doivent être définies comme suit :

- H3DEF.SRC : PUBLSYM EQU 1
- H3FB.SRC : EXTNSYM EQU 1

Si l'on choisit cette méthode d'assemblage, tous les symboles utilisés (à partir du fichier H3DEF.SRC) dans le programme utilisateur doivent être définis comme EXTERNAL. Les adresses de base des blocs registres et de champ d'indicateurs doivent être définies dans le fichier utilisateur, étant donné que l'assembleur n'autorise pas une addition de deux symboles définis extérieurement en tant qu'opérande symbolique (ex. SET F Fstart+FA1).

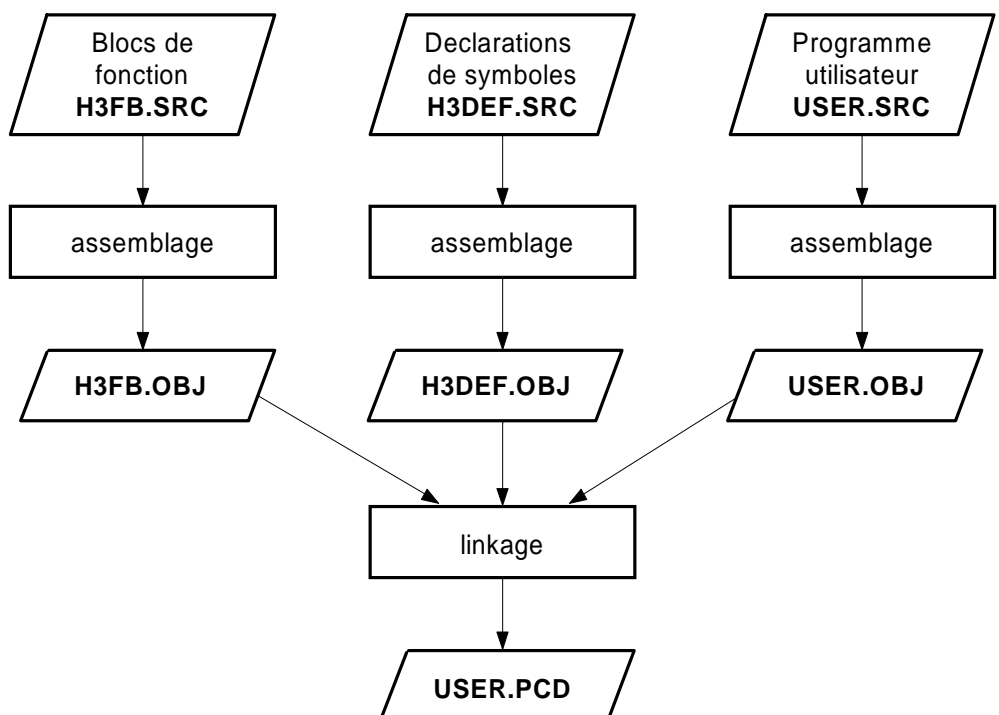
Définition de "RAi" et de "FAi" dans le programme utilisateur (voir également à ce sujet le chapitre 7.1.3).

```

RA1 EQU 0*NoRfeA
RA2 EQU 1*NoRfeA
|
FA1 EQU 0*NoFfeA
FA2 EQU 1*NoFfeA
|
    
```

Pour la raison indiquée ci-dessus, les valeurs absolues doivent être utilisées pour les symboles "NoRfeA" et "NoFfeA". Elles peuvent être extraites du fichier H3DEF.

Le diagramme ci-dessous indique comment doivent être assemblés et linkés les fichiers H3 et le programme utilisateur.



### 7.1.3 Configuration de l'installation dans le fichier H3DEF.SRC

L'installation H3 doit être configurée dans le fichier avant de commencer à programmer.

Il faut formuler des indications comme, par exemple, l'adresse de base du premier module H3, le nombre d'axes utilisés, les adresses de base des zones de mémorisation, etc. Toutes les définitions sont déjà mises à une valeur standard et peuvent être adaptées. Le tableau ci-dessous indique comment les indications ont été faites dans le fichier. Ce tableau se trouve au début du fichier.

Nom symbolique	Adresse	Commentaire
FMAH3 EQU	0	; First Module Address H3 ; Définit l'adresse de base du ; premier module H3. ; <u>Attention</u> : tous les modules H3 ; doivent être enfichés sans intervalle ; sur le bus PCD4.
IMode EQU	6	; Initialisation Mode ; Définit en fonction de la sortie valeur ; de consigne (Analogique ou PWM). ; Module H310,311,320 et 321: IMode = 6 ; Module H316,317,326 et 327: IMode = 5
MNA EQU	2	; Max. Number of Axes ; Définit le nombre des axes utilisés. ; Les zones mémoires nécessaires ; (Registres, indicateurs) sont réservées en ; fonction de cette indication.
BAF EQU	2000	; Base Address Flags ; Adresse de base des indicateurs utilisés
BAR EQU	2000	; Base Address Registers ; Adresse de base des registres utilisés
BAC EQU	1000	; Base Address Counters ; Adresse de base des compteurs utilisés
BAFB EQU	900	; Base Address FunctionBlocks ; Adresse de base des blocs de fonctions ; utilisés

Nom symbolique	Adresse	Commentaire
RA1	EQU 0*NoRfeA	; Registerblock constant Axis Nr. 1 ; Constante de bloc registres axe n° 1 ; Cette constante pointe sur le premier ; registre du bloc de registres utilisé ; par l'axe n° 1. ; La constante "NoRfeA" définit le ; nombre de registres utilisé par chaque ; axe. "NoRfeA" est déjà défini et ne doit ; pas être modifiée.
RA2	EQU 1*NoRfeA	; Registerblock constant Axis Nr. 2 ; Constante de bloc registres axe n° 2
FA1	EQU 0*NoFfeA	; Flagrange constant Axis Nr. 1 ; Constante de champ d'indicateurs axe n° 1. ; Cette constante pointe sur le premier ; indicateur du champ d'indicateur utilisé ; par chaque axe. "NoFfeA" définit le ; nombre d'indicateurs utilisés par chaque ; axe. "NoFfeA" est déjà définie et ne ; doit pas être modifiée.
FA2	EQU 1*NoFfeA	; Flagrange constant Axis Nr. 2 ; Constante de champ d'indicateurs axe n° 2

La liste contenant les constantes de bloc registres (RA1, RA2) et les constantes de champ d'indicateurs (FA1, FA2) doit être étendue si l'on utilise plus de deux axes.

Exemple :

Axe i	RAi	FAi
1	0*NoRfeA	0*NoFfeA
2	1*NoRfeA	1*NoFfeA
3	2*NoRfeA	2*NoFfeA
4	3*NoRfeA	3*NoFfeA
5	4*NoRfeA	4*NoFfeA
etc.		

Après ce tableau de configuration des symboles, suit une liste contenant toutes les affectations de symboles; elle ne doit pas être modifiée.

Numération des axes:

Chaque module H3 occupe 2 numéro d'axe. Ceci est aussi valable pour le module PCD4.H31.. qui a seulement 1 contrôleur d'axe.

Exemple: Le schéma ci-dessous représente une configuration des modules H3 sur le bus PCD4 avec la numération correspondante.

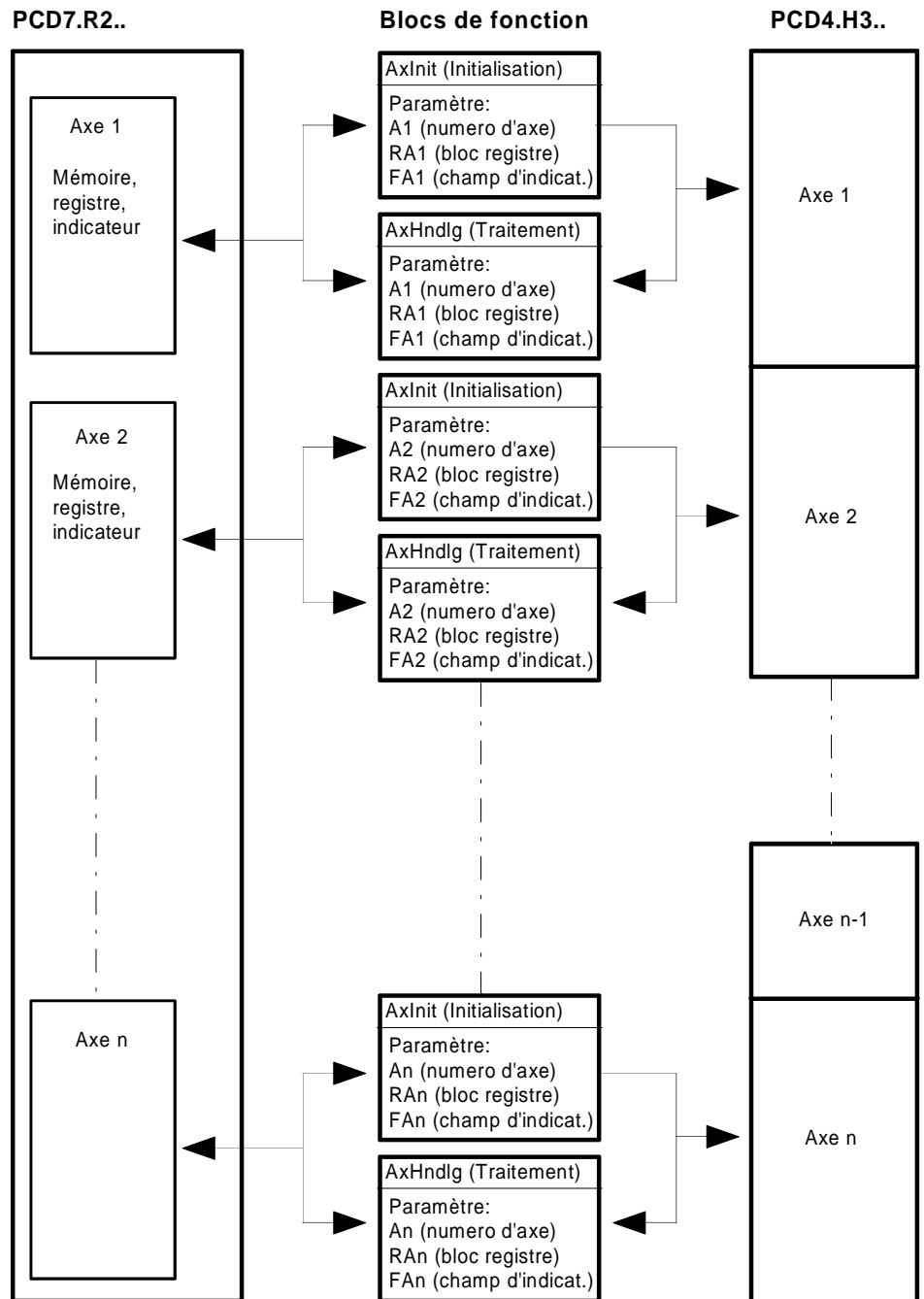
H320		H310		H320	
No.		No.		No.	
Axe X	--> 1	Axe X	--> 3	Axe X	--> 5
Axe Y	--> 2	(Axe Y	--> 4)	Axe Y	--> 6
		n'existe pas			

Si l'on entreprend la configuration du système, il est intéressant de connaître combien de mémoire est utilisée au total par le logiciel H3. Un tableau se trouve à la fin du fichier H3DEF.SRC dans lequel est calculée, lors de l'assemblage, la place mémoire occupée. Le fichier H3DEF.LST renseigne donc sur la quantité de mémoire occupée (voir le tableau ci-dessous).

Mémoire utilisée par le logiciel H3 :

Symbole		Adresse		Désignation
TFI	EQU F	NBF-BAF		; Total used Flags ; Total des indicateurs utilisés
TCO	EQU C	NBC-BAC		; Total used Counters ; Total des compteurs utilisés
TRE	EQU R	NBR-BAR		; Total used Registers ; Total des registres utilisés
TFB	EQU	NFB-BAFB		; Total used Functionblocks ; Total blocs de fonction utilisés
NFF	EQU F	NBF		; Next free Flag ; Prochain indicateur libre
NFC	EQU C	NBC		; Next free Counter ; Prochain compteur libre
NFR	EQU R	NBR		; Next free Register ; Prochain registre libre
NFFB	EQU	NFB		; Next free Functionblock ; Prochain bloc de fonction libre

### 7.1.4 Transfert des données CPU <--> Module H3



Le module H3 est géré par les deux blocs de fonction (FB) "AxInit" et "AxHndlg" et les données mémorisées, spécifiques à chaque axe. Les blocs de registres et des indicateurs déclarés par chaque axe stockent les paramètres et pourront être échangés en lecture et écriture avec le module H3. En précisant le numero d'un axe (A<sub>i</sub>), le bloc registre (RA<sub>i</sub>) et des indicateurs (FA<sub>i</sub>), l'ensemble des données spécifiques à l'axe (i) sera utilisé par le module H3.

### 7.1.5 Organisation et accès des données d'axe dans la mémoire PCD7.R2..

#### Occupation des registres

Constante de bloc registres	Adresse	Symbole	Désignation
Adresse de base registres		Axe 1	
	BAR+RA1+0	KProp	Facteur proportionnel
	BAR+RA1+1	KInt	Facteur intégral
	BAR+RA1+2	KDer	Facteur dérivée
	BAR+RA1+3	IntL	Limite d'integration
	BAR+RA1+4	SampI	Temps d'échantillonnage (sur dérivée)
	BAR+RA1+5	MCFac	Facteur mécanique
	BAR+RA1+6	PosEr	Erreur de position (tolerance de position)
	BAR+RA1+7	DestP	Position cible
	BAR+RA1+8	BrkP	Position d'interruption
	BAR+RA1+9	Veloc	Vitesse
	BAR+RA1+10	Accel	Accélération
	BAR+RA1+11	StaFRR	Status Flag Reset Register
	BAR+RA1+12	MCW	Mot de contrôle mouvement
	BAR+RA1+13	RActP	Position effective (actuelle)
	BAR+RA1+14	RSetP	Consigne de position
	BAR+RA1+15	RActV	Vitesse effective (actuelle)
	BAR+RA1+16	RSetV	Consigne de vitesse
	BAR+RA1+17	RIndP	Position d'index
	BAR+RA1+18	RIntTS	Somme d'intégration
BAR+RA1+19	RSigB	Registre de signalisation	
	BAR+RA2+0	Axe 2	
	BAR+RA2+19		
	BAR+RA3+0	Axe 3	
	BAR+RA3+19		
	BAR+RA <sub>n</sub> +0	Axe n	
	BAR+RA <sub>n</sub> +19		
		Registre commun	Utilisé comme mémoire tampon par les FBs



		<b>Occupation des indicateurs</b>	
Constante de champ d'indicateurs	Adresse	Symbole	Désignation
Adresse de base indicateurs		Axe 1	
	BAF+FA1+0	OnDest	Position cible atteinte
	BAF+FA1+1	IPuls	Impulsion d'index saisie
	BAF+FA1+2	WrapOc	Dépassement de registre de position
	BAF+FA1+3	ExcPEr	Erreur de position
	BAF+FA1+4	BrkPos	Position d'interruption atteinte
	BAF+FA1+5	DplM	Type d'affichage sur PCA2.D14
			Indicateur de commande fonctionnelle
	BAF+FA1+6	FLdDR	Charger position cible relative
	BAF+FA1+7	FLdDA	Charger position cible absolue
	BAF+FA1+8	FLdVR	Charger vitesse relative
	BAF+FA1+9	FLdVA	Charger vitesse absolue
	BAF+FA1+34	FBackw	En arrière à vitesse définie
	BAF+FA2+0	Axe 2	
BAF+FA2+34			
BAF+FA3+0	Axe 3		
BAF+FA3+34			
BAF+FA <sub>n</sub> +0	Axe n		
BAF+FA <sub>n</sub> +34			
		Indicateur commun	Utilisé comme mémoire tampon par les FBs

## Occupation des compteurs

Un seul compteur est utilisé en tampon pour l'ensemble des axes.

## Accès du programme utilisateur aux registres et aux indicateurs

L'adressage des paramètres pour chaque axe se fait au moyen de noms symboliques (définis dans le fichier H3DEF.SRC). Les paramètres portent le même nom pour tous les axes. On fait la différence entre les axes par l'addition au nom symbolique, de l'adresse de début du bloc paramètres de l'axe concerne. Si l'on ne donne que le nom symbolique, les paramètres de l'axe 1 seront choisis automatiquement. Les adresses de début des blocs paramètres sont définies au moyen des constantes R<sub>Ai</sub> (constante de bloc registres pour l'axe i) et F<sub>Ai</sub> (constante de champ d'indicateurs pour l'axe i). Les exemples qui suivent indiquent comment on accède concrètement aux paramètres de chaque axe. Les noms symboliques sont écrits dans le logiciel H3 en majuscules et en minuscules, pour une meilleure lisibilité. Etant donné que l'assembleur PCD ne fait pas la distinction entre les majuscules et les minuscules, on peut écrire les noms dans le programme utilisateur à volonté en lettres majuscules ou minuscules.

Avant de pouvoir charger un paramètre dans la mémoire du module H3, il faut préalablement charger le registre correspondant avec la valeur désirée.

Exemple : chargement des registres "DestP" (position cible) pour les différents axes :

```
- Axe 1    —> LD  R   DestP+RA1
              Valeur

- Axe 3    —> LD  R   DestP+RA3
              Valeur

- Axe i    —> LD  R   DestP+RAi
              Valeur
```

Pour charger maintenant cette position cible dans le module H3, il faut exécuter la fonction "charger position cible". En armant l'indicateur "FLdDA" (charger position cible absolue) le registre "DestP" du bloc de fonction "AxHndlg" est lu et chargé dans le module H3.

Exemple : charger la "position cible absolue" dans les modules H3 pour les axes suivants :

- Axe 1    —> SET F    FLdDA+FA1
- Axe 3    —> SET F    FLdDA+FA3
- Axe n    —> SET F    FLdDA+FA<sub>n</sub>

Après avoir lancé un mouvement, il faut attendre jusqu'à ce que la position cible soit atteinte. Pour ce faire, il faut interroger l'indicateur d'état "OnDest".

Interrogation de l'indicateur d'état par le biais du programme utilisateur pour les axes suivants :

- Axe 1    —> STH F    OnDest+FA1
- Axe 3    —> STH F    OnDest+FA3
- Axe n    —> STH F    OnDest+FA<sub>n</sub>

## 7.2 Blocs de fonctions principaux "AxInit" et "AxHndlg"

La communication avec les modules H3 se fait exclusivement au moyen des deux FB "AxInit" (initialisation) et "AxHndlg" (traitement ou paramétrisation).

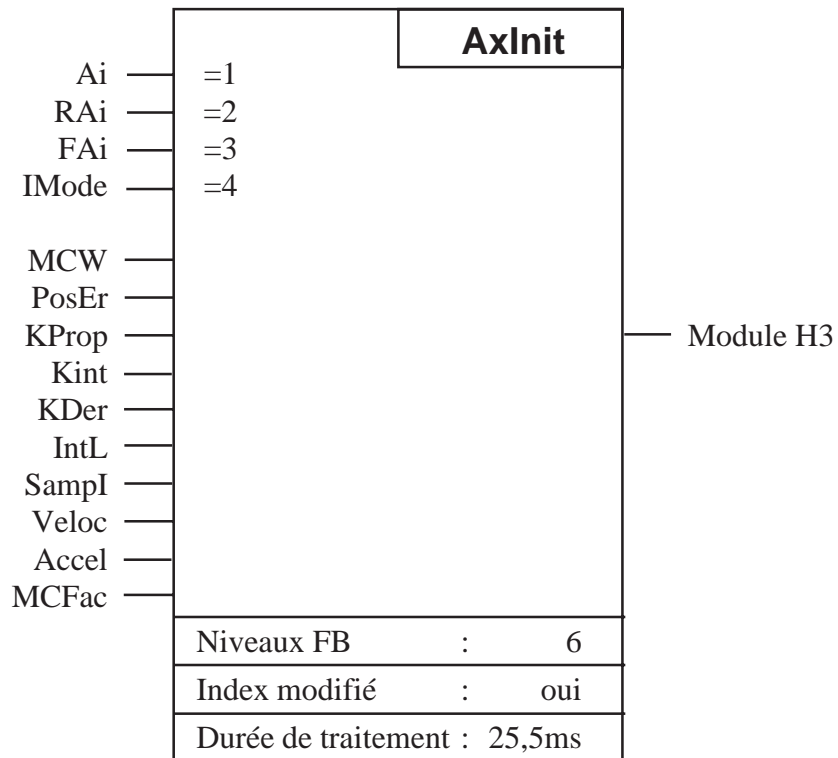
Lors de l'appel de ces FB, on transmet en tant que paramètres le numéro, la constante de bloc registres et la constante de champ d'indicateurs de l'axe à traiter, afin d'adresser les données correspondantes dans le module mémoire du PCD4 et du module H3. Ces deux FB, avec l'indication des paramètres susmentionnés, doivent être appelés pour chacun des axes.

### AxInit

Bloc de fonction: - **Ax** **I**nitialisation  
- Initialisation d'axe

### AxInit

Paquet logiciel: PCD9.H3E1



### Description de la fonction

Ce FB initialise un axe dans le module H3. Avant de pouvoir travailler avec un axe, il faut d'abord traiter une fois ce bloc de fonction. Il est avantageux de l'appeler dans le XOB 16, car il ne sera traité qu'une fois. Lors de l'exécution de ce FB, une série de fonctions est automatiquement exécutée. Une partie des fonctions sont les mêmes que celles concernées par le FB de traitement des axes "AxHndlg"; c'est pourquoi elles ne sont pas décrites plus en détail ici.

Les fonctions suivantes sont exécutées :

### 1. Reset du contrôleur micro processeur dans le module H3

Tous les paramètres de mouvement (accélération, vitesse, position-cible et position d'interruption) et PID (y compris grandeur de réglage) sont mis à zéro. De même, la position actuelle est définie comme position origine.

### 2. Initialisation du port de sortie de la grandeur de réglage

Le port de sortie est initialisé en fonction du paramètre "IMode". La constante "IMode" doit être définie de manière correspondant au module H3 utilisé dans le fichier H3DEF.SRC.

- |   |                            |
|---|----------------------------|
| <b>3. Choix du mode d'exploitation</b>                            | —> voir fonction "FSeIOM"  |
| <b>4. Chargement de l'erreur de position</b>                      | —> voir fonction "FSetPEr" |
| <b>5. Chargement des paramètres de réglage</b>                    | —> voir fonction "FLdRP"   |
| <b>6. Mise à jour des paramètres de réglage</b>                   | —> voir fonction "FUpDRP"  |
| <b>7. Chargement de l'accélération</b>                            | —> voir fonction "FLdAA"   |
| <b>8. Chargement de la vitesse</b>                                | —> voir fonction "FLdVA"   |
| <b>9. Reset de tous les indicateurs de commande fonctionnelle</b> | —> voir FB "AxHndlg"       |

Exemple : appel du bloc de fonction pour l'axe 3

CFB	AxInit	
	3	; numéro d'axe
	RA3	; bloc registres axe 3
	FA3	; champ d'indicateurs axe 3
	IMode	; mode d'initialisation

**Description des entrées et des sorties :**

Si aucune autre indication n'est donnée, tous les noms symboliques utilisés sont définis dans le fichier H3DEF.SRC et doivent être repris sans modification.

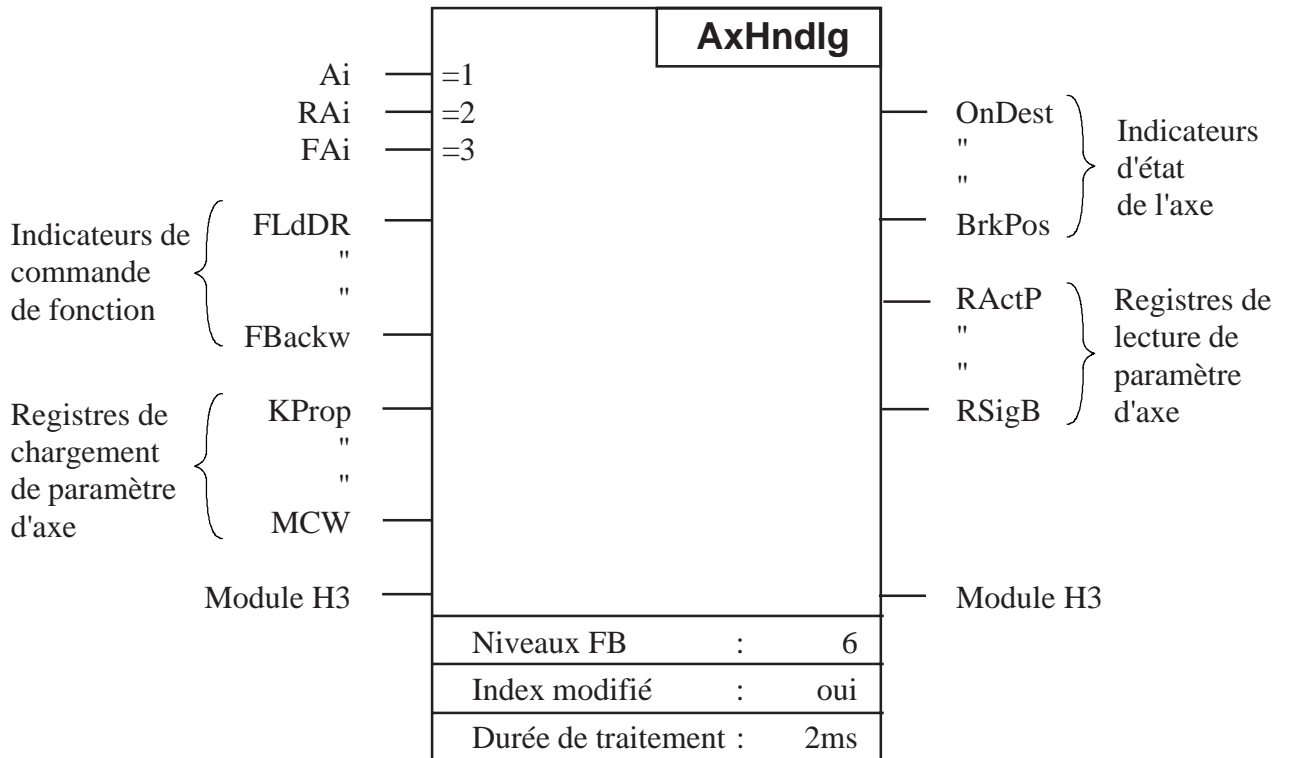
Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
Ai	Numéro d'axe i Ai n'est pas défini comme symbole dans H3DEF.SRC; il peut être directement affecté à une constante.	oui	K	entier	1...32
RAi	Constante de bloc registres axe i	oui	K	entier	$(i-1) * 20$
FAi	Constante de champ d'indicateurs axe i	oui	K	entier	$(i-1) * 35$
IMode	<b>Initialisation Mode</b> Initialisation de la sortie de grandeur de réglage	oui	K	entier	5H/6H
MCW	<b>Motion Control Word</b> Mot de contrôle de mouvement	non	R	binaire	-
PosEr	<b>Position Error</b> Tolerance position Unité: impulsion	non	R	entier	0...32767
KProp	<b>Proportional factor</b> Facteur proportionnel	non	R	entier	0...32767
KInt	<b>Integral Factor</b> Facteur intégral	non	R	entier	0...32767
KDer	<b>Derivative Factor</b> Facteur dérive	non	R	entier	0...32767
IntL	<b>Integration Limit</b> Limite d'intégration	non	R	entier	0...32767
Sampl	<b>Sampling Interval</b> Intervalle d'échantillonnage	non	R	entier	0...255
Veloc	<b>Velocity</b> Vitesse	non	R	entier	voir foction "FLdVA"
Accel	<b>Acceleration</b> Accélération	non	R	entier	voir fonction "FLdAA"
MCFac	<b>Motion Control Factor</b> Facteur mecanique	non	R	virgule flottante	voir fonction "FLdDA"

# AxHndlg

Bloc de fonction: - **Axis Handling**  
- Traitement des axes

# AxHndlg

Paquet logiciel: PCD9.H3E1



### Description de la fonction

Après l'initialisation d'un axe au moyen du FB "AxInit", la communication avec le module H3 se fait exclusivement au moyen de ce bloc. Les paramètres d'axe peuvent être chargés dans le module H3 et lus par celui-ci. Le module reçoit aussi par l'intermédiaire de ce FB les commandes de mouvement. Les entrées et sorties du FB peuvent se répartir dans les groupes suivants :

- Entrées :
- paramètres FB
  - indicateur de commande de fonction
  - registre de chargement de paramètre d'axe
  - module H3

- Sorties :
- indicateur d'état d'axe
  - registre de lecture de paramètre d'axe
  - module H3

## **Paramètre FB**

L'axe correct et les données relatives dans le module H3 (indicateurs de commande, indicateurs d'état, registres de chargement et registres de lecture) dans le module mémoire PCD7.R2.. sont, lors de l'appel du FB, automatiquement adressés par la transmission du numéro d'axe (Ai), la constante du bloc registres (RAi) et la constante de champ d'indicateurs (FAi).

## **Indicateurs de commande de fonction**

En armant un indicateur de commande dans le programme utilisateur, il est possible d'activer une fonction déterminée.

(Par exemple : chargement d'une position cible, lancement d'un mouvement, etc.) Lors du traitement du FB, tous les indicateurs de commande de fonction sont interrogés. Si un indicateur est armé (mis à "1"), le FB exécute la fonction immédiatement après l'interrogation correspondante. Celle-ci est réalisée par un sous-bloc fonctionnel (portant le même nom que l'indicateur de commande, mais sans la lettre préfixe F).

Après l'exécution de cette fonction, l'indicateur est remis à zéro.

Chaque indicateur est affecté à une fonction. Si aucun indicateur n'est armé, le FB est quitté sans qu'aucune fonction ne soit exécutée.

Ces indicateurs de commande sont stockés séparément pour chaque axe dans le module mémoire PCD7.R2.. Les indicateurs sont armés par l'utilisateur, de la façon d'écrite dans le chapitre 7.1.5 ci-dessus.

Afin d'éviter l'exécution par inadvertance de fonctions non désirées après le démarrage du PCD4, les indicateurs de commande sont automatiquement mis à zéro par le bloc fonctionnel d'initialisation de l'axe "AxInit". Voir le chapitre 7.4 pour la description de chacune des fonctions.

## **Registres de chargement des paramètres d'axe**

Les registres de chargement contiennent tous les paramètres nécessaires à l'exploitation d'un axe. Les registres doivent être chargés par l'utilisateur aux valeurs adéquats. Si un paramètre est chargé, la fonction correspondante doit être activée pour copier la valeur du registre de chargement dans la mémoire du module H3.

## **Indicateurs d'état de l'axe**

A chaque traitement du FB, il y a lecture automatique de l'état d'axe et copie dans ces indicateurs. L'état de l'axe peut alors être interrogé par le programme utilisateur au moyen de ces indicateurs. Pour la signification de chacun des indicateurs, voir la description de la fonction "FResSF".

## **Registres de lecture des paramètres d'axe**

Les paramètres d'axe pouvant être lus du module H3 sont copiés dans



les registres de lecture.

### Module H3

Repere symbolique pour le contrôleur d'axe dans le module H3.

#### Appel du bloc fonctionnel :

Appel du FB pour l'axe 2

CFB	AxHndlg	
	2	; numéro d'axe
	RA2	; bloc registres axe 2
	FA2	; champ d'indicateur axe 2

### Structure de programme et application des blocs de fonctions "AxInit" et "AxHndlg"

Le programme utilisateur destiné au module H3 peut, en gros, se répartir en trois parties :

- initialisation des axes,
- traitement cyclique des axes,
- définition du programme de déplacement (déroulement des mouvements de la machine).

#### a) Initialisation des axes

Le premier élément d'un programme H3 consiste toujours en l'initialisation des axes. Un axe est initialisé dans le XOB 16 par le biais de l'appel au FB "AxInit". Lors du traitement du FB, plusieurs registres sont lus; ils doivent donc être chargés au préalable. Pour déterminer les valeurs des registres, il faut déjà connaître différentes données de l'entraînement (voir aussi le chapitre 9, exemples d'application). L'axe est prêt à l'exploitation après son initialisation par le FB "AxInit".

#### b) Traitement cyclique des axes

Font partie du traitement cyclique toutes les tâches qui sont exécutées en permanence. C'est pourquoi elles sont programmées dans un COB qui est traité cycliquement. Le bloc de fonction "AxHndlg" assume l'ensemble des échanges de données avec le contrôleur du module H3. L'indicateur de commande de fonction communique au FB les tâches qu'il doit exécuter. Le rôle du FB consiste ainsi dans l'interrogation constante de ces indicateurs de commande et dans l'exécution des commandes correspondantes.

C'est pourquoi il convient de programmer ce bloc de fonction dans un COB.

### c) Définition du programme de déplacement

Etant donné que le programme de déplacement d'un entraînement consiste toujours en un déroulement séquentiel, il convient de le programmer dans une structure GRAFTEC appropriée.

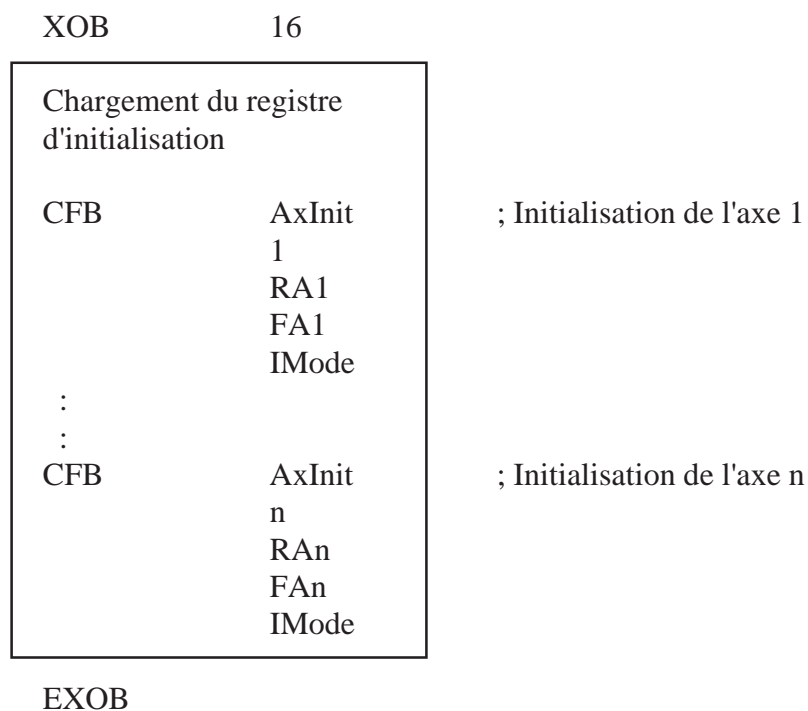
Le déroulement de principe est toujours le même et contient les pas suivants :

- un mouvement est défini et lancé (--> Step: Etape)
- on attend jusqu'à ce que l'axe ait terminé le mouvement (--> Transition)
- mouvement suivant
- attente
- etc.

Si la transition n'est pas franchissable, on quitte le programme GRAFTEC et l'on poursuit l'exécution cyclique du programme.

--> le bloc de fonction "AxHndlg" est appelé et exécute les tâches qui lui sont demandées par la structure GRAFTEC. (Charger le paramètre de mouvement dans le module H3 et lancer le mouvement).

Structure sommaire d'un programme H3 avec les appels au FB :





## 7.3 Vue d'ensemble sur les fonctions

---

Les fonctions suivantes peuvent être exécutées au moyen du FB "AxHndlg". Le nom indiqué désigne l'indicateur par lequel est activée la fonction.

### Définitions

---

<b>FSeIOM</b>	Choix du mode d'exploitation
<b>FSetPE</b>	Définition de l'erreur de position (tolerance)

---

### Entrées de paramètres pour le profil de vitesse

---

<b>FLdDA</b>	Charger la position cible absolue
<b>FLdDR</b>	Charger la position cible relative
<b>FLdVA</b>	Charger la vitesse absolue
<b>FLdVR</b>	Charger la vitesse relative
<b>FLdAA</b>	Charger l'accélération absolue
<b>FLdAR</b>	Charger l'accélération relative
<b>FLdRP</b>	Charger les paramètres de réglage (P.I.D)
<b>FUpDRP</b>	Activer les paramètres de réglage chargé

---

### Commandes de déplacement

---

<b>FStart</b>	Commencer le mouvement ou memorisations des paramètres nouvellement chargés
<b>FStop</b>	Arrêter le mouvement
<b>FMotOff</b>	Déclencher la régulation (mise hors asservissement)
<b>FSSStepV</b>	Avance d'un pas d'encodeur
<b>FSSStepB</b>	Recul d'un pas d'encodeur
<b>FForw</b>	Marche avant à vitesse définie
<b>FBackw</b>	Marche arrière à vitesse définie

---

---

**Commandes de lectures pour les données**

---

<b>FRdAP</b>	Lire la position effective (actuelle)
<b>FRdSP</b>	Lire la consigne de position
<b>FRdAV</b>	Lire la vitesse effective (actuelle)
<b>FRdSV</b>	Lire la consigne de vitesse
<b>FRdITS</b>	Lire la somme d'intégration
<b>FRdIP</b>	Lire la position d'index
<b>FRdSR</b>	Lire le registre de signalisation

---

**Fonctions auxiliaires**

---

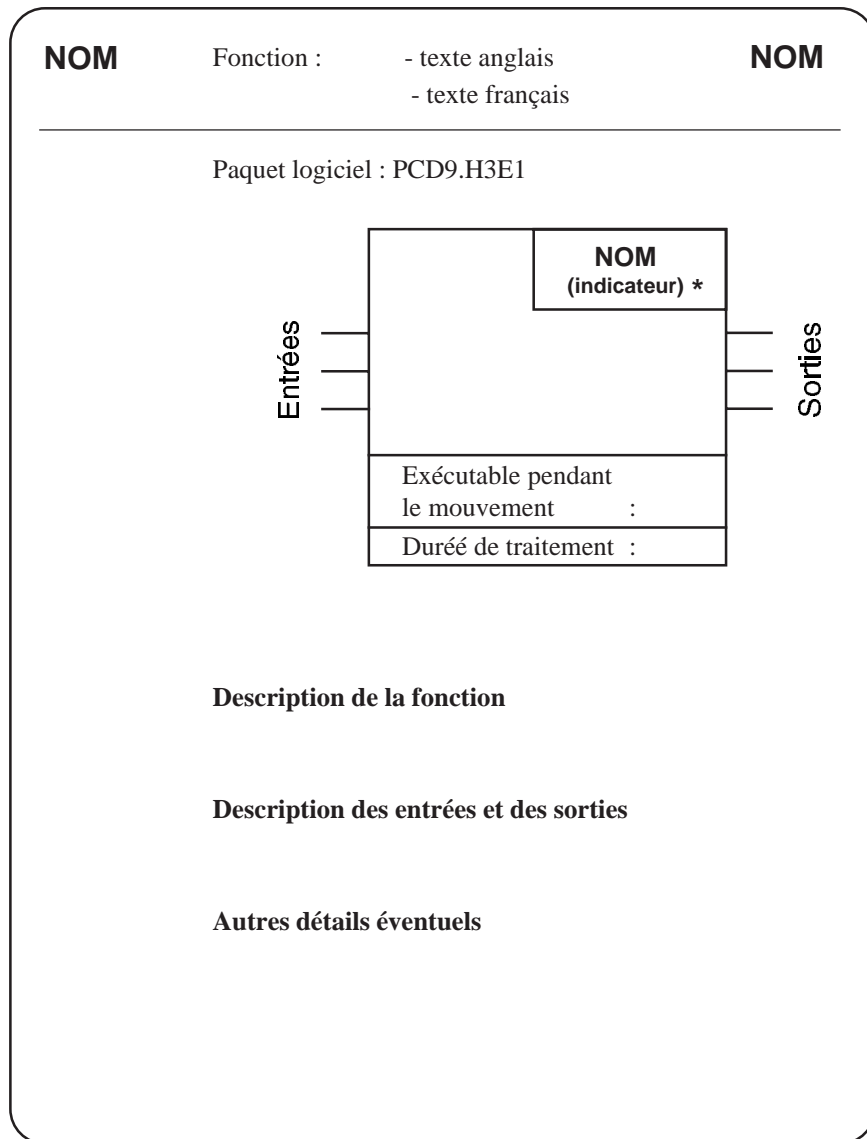
<b>FResSF</b>	Reset indicateur d'état
<b>FLdBPA</b>	Charger la position d'interruption absolue
<b>FLdBPR</b>	Charger la position d'interruption relative
<b>FSetIP</b>	Saisie de la position d'index
<b>FSetZP</b>	Déterminer la position zéro

---

**Notes personnelles :**

## 7.4 Description des fonctions

Afin d'assurer une vue d'ensemble claire, toutes les fonctions qui suivent sont présentées selon le même schéma :



Pour ce qui suit, l'ordre du sommaire du chapitre 7.3 est respecté. A la fin du présent manuel, se trouve de plus une liste alphabétique de tous les noms et symboles utilisés.

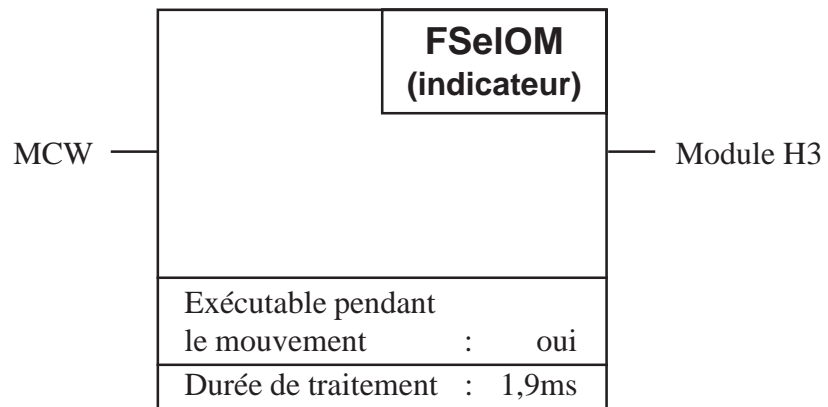
\*) Indique que le nom est utilisé par un indicateur et que la fonction peut être exécutée en armant cet indicateur.

**FSeIOM**

Fonction : - **Select Operation Mode**  
- Choix du mode d'exploitation

**FSeIOM**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

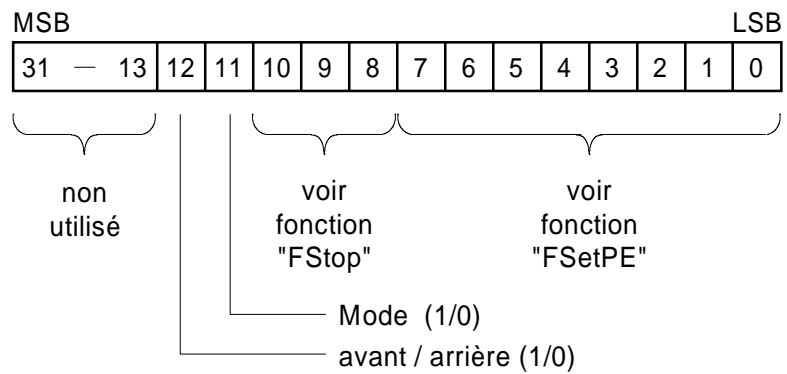
Cette fonction permet la définition du type d'exploitation des axes (mode de fonctionnement positionnement ou vitesse de rotation). Le mode de fonctionnement nouvellement défini n'est valable qu'après l'exécution de la commande suivante de démarrage "FStart".

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
MCW	Motion Control Word Mot de contrôle de mouvement	non	R	binaire	voir page suivant



**Signification du mot de contrôle de mouvement "MCW" :**



La fonction ne lit que les bits 11 et 12 du registre "MCW".

Bit 11: 0 --> mode "positionnement"  
 1 --> mode "vitesse de rotation"

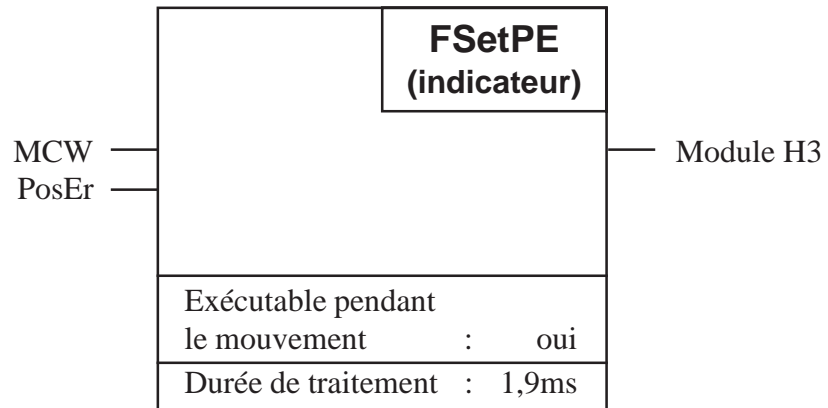
Bit 12: 0 --> en arrière  
 1 --> en avant

**FSetPE**

Fonction : - **Set Position Error**  
 - Définition de l'erreur de position  
 (tolerance)

**FSetPE**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

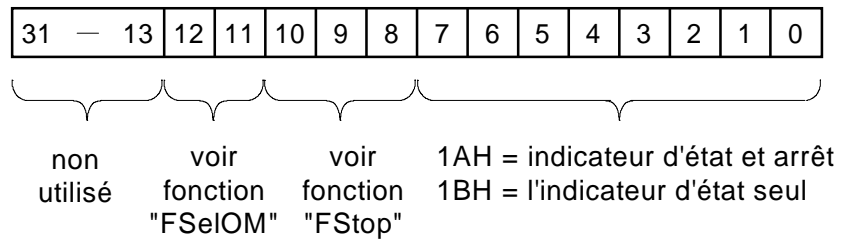
Cette fonction détermine la différence maximum (tolérance) entre la position réelle et la position de consigne instantanée. Si la différence (valeur) atteint la valeur définie par cette fonction, l'indicateur d'état "ExcPEr" est armé. On indiquera dans le registre "MCW" si, en cas d'erreur, l'indicateur d'état seul doit être armé, ou bien s'il faut aussi déclencher la régulation (sortie de grandeur de réglage égale à zéro).

Une erreur de position signale un problème sérieux; elle peut ainsi être surveillée.

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
PosEr	<b>Position Error</b> Erreur de position	non	R	entier	0.. 32'767 Imp.
MCW	<b>Motion Control Word</b> Mot de contrôle de mouvement	non	R	binaire	voir page suivant

**Signification du mot de contrôle de mouvement "MCW" :**



La différence entre la position de consigne et la position réelle est chargée directement en impulsions d'encodeur. Il faut savoir que les impulsions de l'encodeur sont multipliées par quatre (évaluation des flancs) dans le décodeur de position.

Exemple :

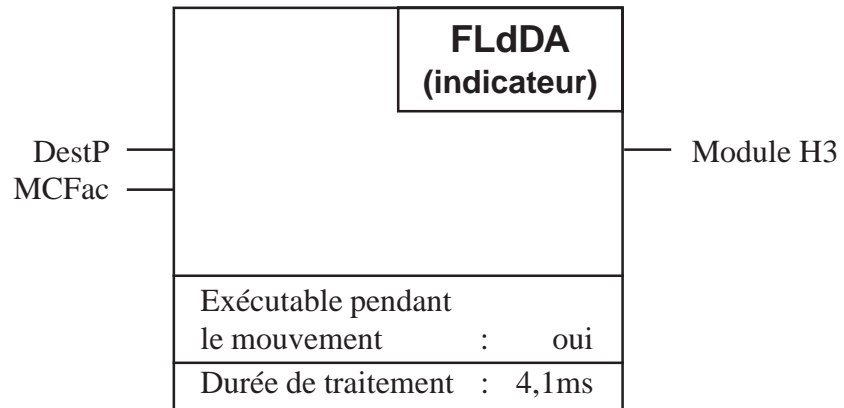
Si la différence doit comporter au maximum 500 impulsions d'encodeur, il faut charger la valeur  $4 * 500 = 2000$  dans le registre "PosEr".

**FLdDA**

Fonction : - **Load Destination Absolute**  
- Charger la position cible absolue

**FLdDA**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction charge une position cible absolue dans un registre intermédiaire sur le module H3. Absolu signifie que la valeur est par rapport à la position zéro (origine). La position nouvellement chargée n'est reprise par le module H3 dans le registre de travail que lors de la commande "FStart" suivante.

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
DestP	<b>Destination Position</b> Position cible  Valeur: [-2 <sup>30</sup> ..+(2 <sup>30</sup> -1)]/k*10 <sup>-3</sup> Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. 9,223371*10 <sup>18</sup>

**Facteur mécanique k dans le registre "MCFac" :**

Ce facteur détermine l'unité pour l'introduction d'une position cible, de la vitesse et de l'accélération. Le facteur se calcule à partir de la résolution d'encodeur et de la réduction de la mécanique. Le facteur k doit être calculé et chargé dans le registre "MCFac". Il est lu par plusieurs fonctions afin de convertir une mesure métrique en impulsions d'encodeur et vice-versa.

La formule  $k = \frac{4 \cdot In}{s}$  s'applique, où

In: pas / tour (résolution d'encodeur)

s : distance / tour (systeme vis-ecrou et engrenages)

En même temps que l'unité pour la distance, on détermine également l'unité d'une position, de la vitesse et de l'accélération.

Exemple:

Broche au pas de 3 mm

Résolution d'encodeur 1000 pas/tour

Il faut avancer vers la position cible de 60 mm et la donnée doit être introduite (resp. résolution) en  $\mu\text{m}$ .

$$k = \frac{4 \cdot In}{s} = \frac{4 \text{ imp/pas} \cdot 1000 \text{ pas/t.}}{3000 \mu\text{m/t.}} = 1,33333 \text{ imp}/\mu\text{m}$$

Registre d'entrée "DestP" = 60000  $\mu\text{m}$

Dans l'exemple ci-dessous, on veut donner la position en unité de 1/10 mm:

$$k = \frac{4 \cdot In}{s} = \frac{4 \text{ imp/pas} \cdot 1000 \text{ pas/t.}}{30 \text{ 1/10mm/t.}} = 133,333 \text{ imp/ 1/10mm}$$

Registre d'entrée "DestP" = 600 1/10 mm

Nota: Pour un déplacement en nombre de pas d'encodeur, la valeur k=4 doit être utilisée.

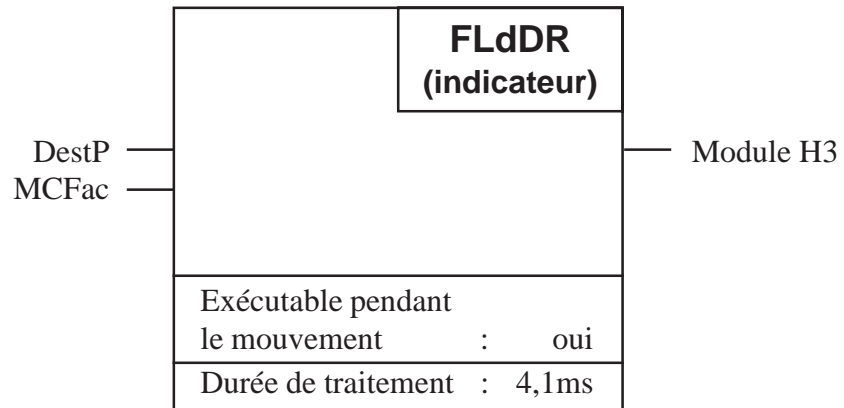
Pour un déplacement en nombre d'impulsion, la valeur k=1 doit être chargée dans le registre "MCFac".

**FLdDR**

Fonction : - **Load Destination Relative**  
- Charger la position cible relative

**FLdDR**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction charge une position cible relative dans un registre intermédiaire sur le module H3. Le chargement relatif veut dire que la valeur est relative à la position cible actuelle. La position nouvellement chargée n'est reprise par le module H3 qu'après la prochaine commande de démarrage "FStart".

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
DestP	<b>Destination Position</b> Position cible  Valeur: [-2 <sup>30</sup> ..+(2 <sup>30</sup> -1)]/k*10 <sup>-3</sup> Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	Virgule flottante	0.. 9,223371*10 <sup>18</sup>

Pour le premier mouvement, seule une position cible absolue peut être chargée après l'initialisation du module. Si une position cible relative est chargée, le contrôleur H3 émet une "Command Error".

**Facteur mécanique k dans le registre "MCFac" :**

Ce facteur a la même signification que dans le cas de la fonction "FLdDA" :

Exemple :

Broche au pas de 3 mm  
Résolution d'encodeur 1000 pas/t

Il faut parcourir un trajet relatif de -60 mm et l'entrée (resp. résolution) doit se faire en  $\mu\text{m}$ .

$$k = \frac{4 \cdot \text{In}}{s} = \frac{4 \text{ imp/pas} \cdot 1000 \text{ pas/t}}{3000 \mu\text{m/t}} = 1,33333 \text{ imp}/\mu\text{m}$$

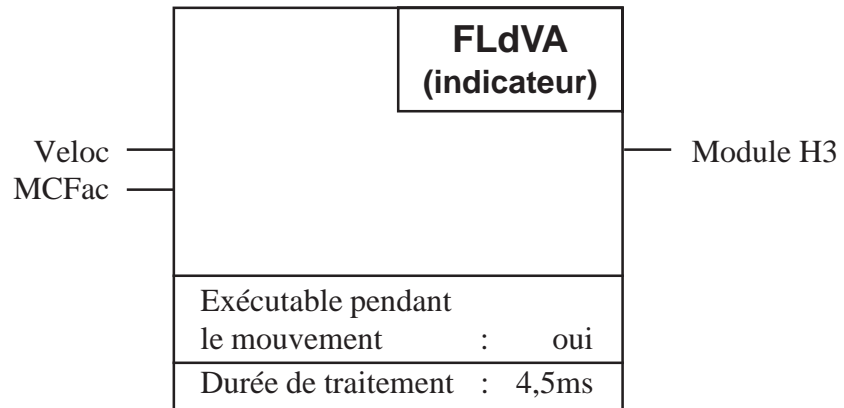
Registre d'entrée "DestP" = -60000  $\mu\text{m}$

**FLdVA**

Fonction : - **Load Velocity Absolute**  
- Charger la vitesse absolue

**FLdVA**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction charge une vitesse absolue d'un registre intermédiaire sur le module H3. Le chargement absolu signifie que la valeur est relative à zéro. La vitesse nouvellement chargée ne sera reprise par le module H3 dans le registre de travail qu'après la commande de démarrage suivante.

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
Veloc	<b>Velocity</b> Vitesse  Valeur: $[0..+(2^{30}-1)]/k*22348*10^{-6}$ Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. $9,223371*10^{18}$



**Facteur mécanique k dans le registre "MCFac" :**

Ce facteur a la même signification que dans le cas de la fonction "FLdDA".

Il faut tenir compte du fait que ce facteur est unique pour la position cible, la vitesse et l'accélération. Il est donc nécessaire de sélectionner les mêmes unités pour l'introduction de ce paramètre.

Exemple:

Broche au pas de 3 mm  
Résolution d'encodeur 1000 pas/t

Il faut s'approcher d'une position cible à une vitesse de 0,1 m/s et l'entrée (resp. résolution) doit se faire en mm/s.

$$k = \frac{4 \cdot I_n}{s} = \frac{4 \text{ imp/pas} \cdot 1000 \text{ pas/t}}{3 \text{ mm/t}} = 1333,3 \text{ imp/mm}$$

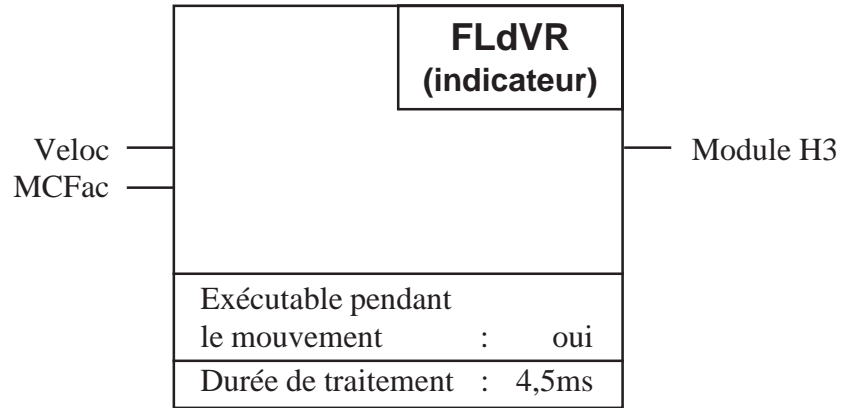
Registre d'entrée "Veloc" = 100 mm/s

# FLdVR

Fonction : - **Load Velocity Relative**  
 - Charger la vitesse relative

# FLdVR

Paquet logiciel : PCD9.H3E1



### Description de la fonction

Cette fonction charge une vitesse relative dans un registre intermédiaire sur le module H3. Chargement relatif signifie que la valeur est relative à la consigne de vitesse actuelle absolue. La vitesse nouvellement chargée n'est reprise par le module H3 dans le registre de travail qu'après la prochaine commande de démarrage.

### Description des entrées et des sorties

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
Veloc	<b>Velocity</b> Vitesse  Valeur: [-2 <sup>30</sup> ..+(2 <sup>30</sup> -1)]/k*22348*10 <sup>-6</sup> Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. 9,223371*10 <sup>18</sup>

Pour le premier mouvement, seule une vitesse absolue peut être chargée après l'initialisation du module. Si une vitesse relative est chargée, le contrôleur H3 émet une "Command Error".

**Facteur mécanique k dans le registre "MCFac" :**

Ce facteur a la même signification que dans le cas de la fonction "FLdDA".

Il faut tenir compte du fait que ce facteur est unique pour la position cible, la vitesse et l'accélération. Il est donc nécessaire de sélectionner les mêmes unités pour l'introduction de ce paramètre.

Exemple:

Broche à pas de 3 mm  
Résolution d'encodeur 1000 pas/t

Il faut s'approcher d'une position cible à une vitesse relative de -0,1 m/s et l'entrée (resp. résolution) doit se faire en mm/s.

$$k = \frac{4 \cdot I_n}{s} = \frac{4 \text{ imp/pas} \cdot 1000 \text{ pas/t}}{3 \text{ mm/t}} = 1333,3 \text{ imp/mm}$$

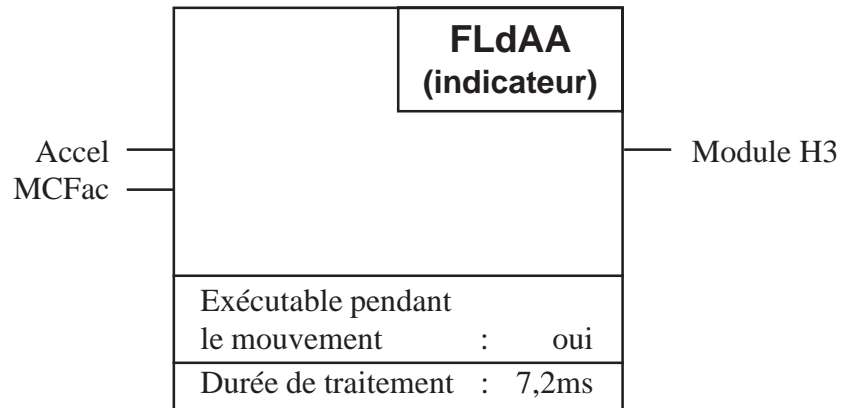
Registre d'entrée "Veloc" = -100 mm/s

**FLdAA**

Fonction : - **Load Acceleration Absolute**  
- Charger l'accélération absolue

**FLdAA**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction charge une accélération absolue dans un registre intermédiaire sur le module H3. Chargement absolu signifie que la valeur est relative à zéro. L'accélération nouvellement chargée n'est reprise par le module H3 dans le registre de travail qu'après la prochaine commande de démarrage. La fonction peut certes être exécutée pendant un mouvement, mais cependant une commande de démarrage "FStart" qui fait travailler le contrôleur H3 avec la nouvelle accélération ne peut être exécutée qu'après la fin d'un mouvement (elle peut aussi intervenir par le biais d'une commande d'arrêt).

Attention : lors de l'appel de cette fonction, la commande "FMotOff" est exécutée avant le chargement de l'accélération.

--> La régulation est déclenchée après l'exécution de cette fonction.

--> Réenclencher la régulation avec "FStart".

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Paramètre	Données		
			Type	Format	Valeur
Accel	<b>Accel</b> Accélération  Valeur: [0..+(2 <sup>30</sup> -1)]/k*76206*10 <sup>-9</sup> Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. 9,223371*10 <sup>18</sup>

**Facteur mécanique k dans le registre "MCFac" :**

Ce facteur a la même signification que dans le cas de la fonction "FLdDA".

Il faut tenir compte du fait que ce facteur est unique pour la position cible, la vitesse et l'accélération. Il est donc nécessaire de sélectionner les mêmes unités pour l'introduction de ce paramètre.

Exemple:

Broche au pas de 3 mm  
Résolution d'encodeur 1000 pas/t

Il faut s'approcher d'une position cible à une accélération de 0,005 m/s<sup>2</sup> et l'entrée (resp. résolution) doit se faire en mm/s<sup>2</sup>.

$$k = \frac{4 \cdot I_n}{s} = \frac{4 \text{ imp/pas} \cdot 1000 \text{ pas/t}}{3 \text{ mm/t}} = 1333,3 \text{ imp/mm}$$

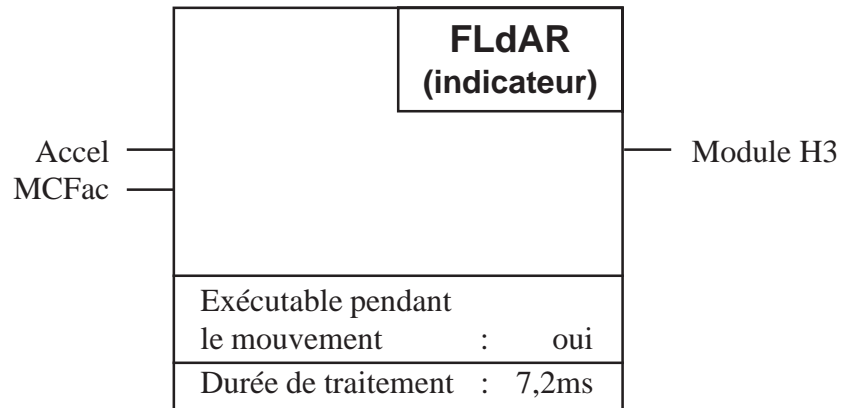
Registre d'entrée "Accel" = 5 mm/s<sup>2</sup>

**FLdAR**

Fonction : - **Load Acceleration Relative**  
- Charger l'accélération relative

**FLdAR**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction charge une accélération relative dans un registre intermédiaire du module H3. Chargement relatif signifie que la valeur est relative à la consigne d'accélération actuelle. L'accélération nouvellement chargée n'est reprise dans le registre de travail par le module H3 qu'après la prochaine commande de démarrage. La fonction peut certes être exécutée pendant un mouvement, mais cependant une commande de démarrage "FStart" qui fait travailler le contrôleur H3 avec la nouvelle accélération ne peut être exécutée qu'après la fin d'un mouvement (elle peut aussi intervenir par le biais d'une commande d'arrêt).

Attention : lors de l'appel de cette fonction, la commande "FMotOff" est exécutée avant le chargement de l'accélération.

- > La régulation est déclenchée après l'exécution de cette fonction.
- > Réenclencher la régulation avec "FStart".

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Paramètre	Données		
			Type	Format	Valeur
Accel	<b>Accel</b> Accélération  Valeur: [-2 <sup>30</sup> ..+(2 <sup>30</sup> -1)]/k*76206*10 <sup>-9</sup> Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. 9,223371*10 <sup>18</sup>

Pour le premier mouvement, seule une accélération absolue peut être chargée après l'initialisation du module. Si une accélération relative est chargée, le contrôleur H3 émet une "Command Error".

**Facteur mécanique k dans le registre "MCFac" :**

Ce facteur a la même signification que dans le cas de la fonction "FLdDA".

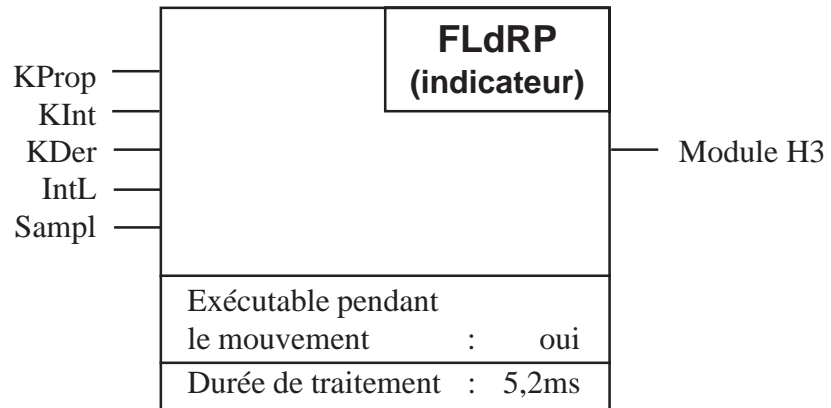
Il faut tenir compte du fait que ce facteur est unique pour la position cible, la vitesse et l'accélération. Il est donc nécessaire de sélectionner les mêmes unités pour l'introduction de ce paramètre.

**FLdRP**

Fonction : - **Load Regulator Parameter**  
- Charger les paramètres de réglage

**FLdRP**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction charge les paramètres de réglage à partir des registres d'axes dans des registres intermédiaires sur le module H3. Le régulateur reprend les valeurs dans le registre de travail après l'exécution de la fonction "FUpDRP".

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
KProp	<b>Proportional Factor</b> Facteur proportionnel	non	R	entier	0.. 32'767
KInt	<b>Integral Factor</b> Facteur intégral	non	R	entier	0.. 32'767
KDer	<b>Derivative Factor</b> Facteur dérivée	non	R	entier	0.. 32'767
IntL	<b>Integration Limit</b> Limite d'integration	non	R	entier	0.. 32'767
Sampl	<b>Sampling Interval</b> derivative term Temps d'échantillonnage (sur dérivée)	non	R	entier	0.. 255



Le temps d'échantillonnage pour la partie dérivée peut être programmée par pas de 341,33 µs.

Durée d'échantillonnage =  $(n + 1) * 341,33 \mu s$

Le nombre n est chargé dans le registre "Sampl".

Exemple: La durée d'échantillonnage doit être de 1024 µs.

--> registre "Sampl" = 2

Le temps d'échantillonnage pour les parties proportionnelle et intégrale est de 341,33 µs et ne peut pas être modifié.

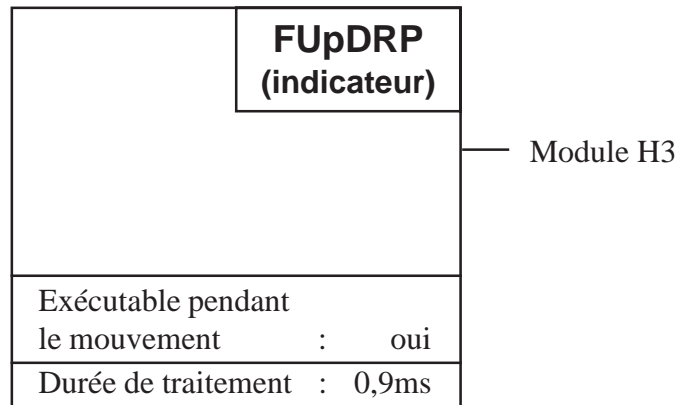
Si une valeur située en dehors de la plage des valeurs est chargée dans le registre, une erreur de commande est émise lors de l'exécution.

**FUpDRP**

Fonction : - **Up Date Regulator Parameter**  
 - Activer les paramètres de réglage chargé

**FUpDRP**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

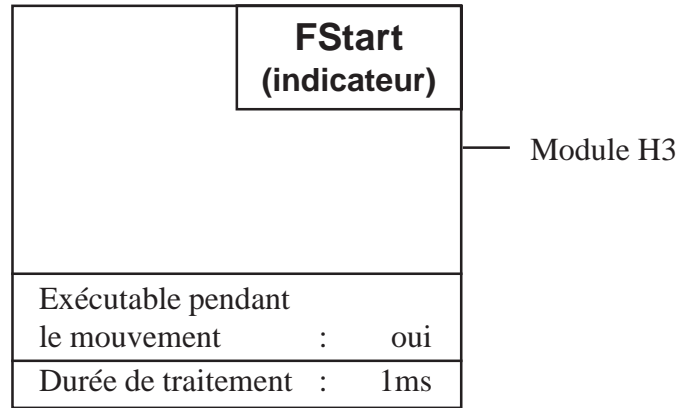
Cette fonction exécute une mise à jour des paramètres de réglage. Le régulateur dans le module H3 reprend les paramètres chargés au moyen de la fonction "FLdRP" à partir des registres intermédiaires dans le registre de travail.

# FStart

Fonction : - **Start** motion  
 - Commencer le mouvement ou memorisations des paramètres nouvellement chargés

# FStart

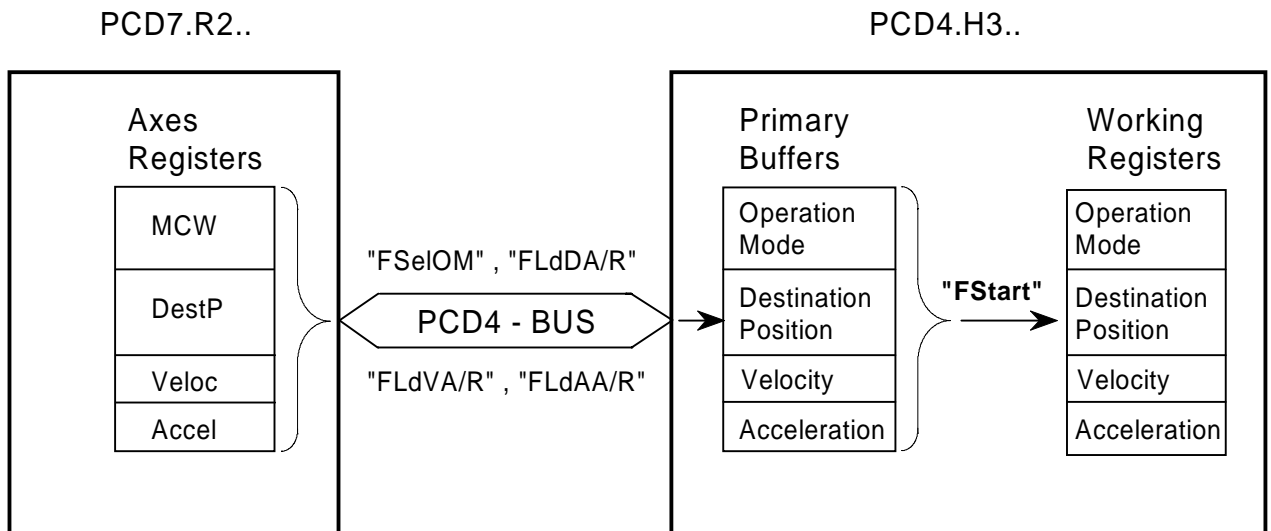
Softwarepaket : PCD9.H3E1



### Description de la fonction

Cette fonction permet de lancer un mouvement ou de demander au contrôleur dans le module H3 de travailler avec un paramètre de mouvement nouvellement chargé (une vitesse par exemple).  
 Si une nouvelle accélération est chargée pendant le mouvement, une commande de démarrage ne sera exécutée qu'après la conclusion de ce mouvement.

Le diagramme ci-dessous montre quels paramètres de mouvement ne sont repris dans le registre de travail du contrôleur H3 qu'après une commande de démarrage.

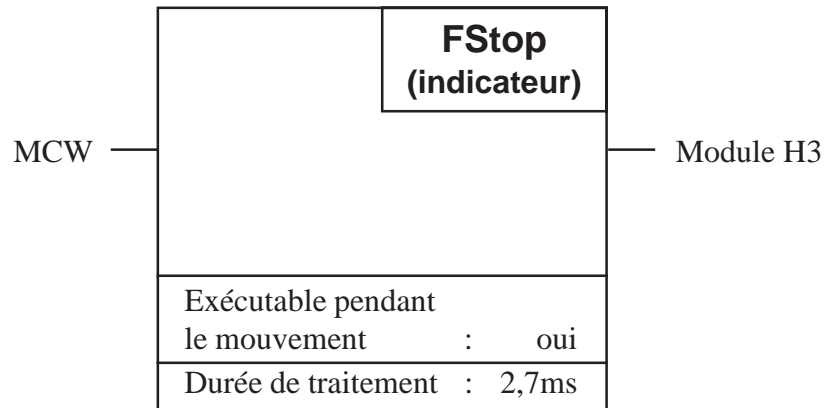


**FStop**

Fonction : - **Stop** motion  
- Arrêter le mouvement

**FStop**

Paquet logiciel : PCD9.H3E1

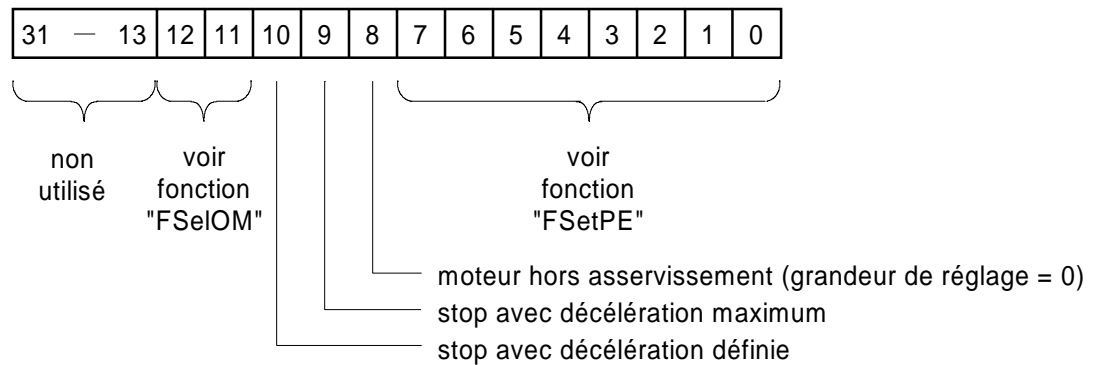
**Description de la fonction**

Cette fonction permet d'arrêter à tout moment un mouvement. L'arrêt se fait avec le type d'arrêt défini dans le registre MCW. L'indicateur d'état "OnDest" est armé après l'exécution du Stop.

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
MCW	<b>Motion Control Word</b> Mot de contrôle de mouvement	non	R	binaire	voir page suivante

**Signification du mot de contrôle de mouvement "MCW" :**



La fonction ne lit que les bits 8 à 10 du registre "MCW"

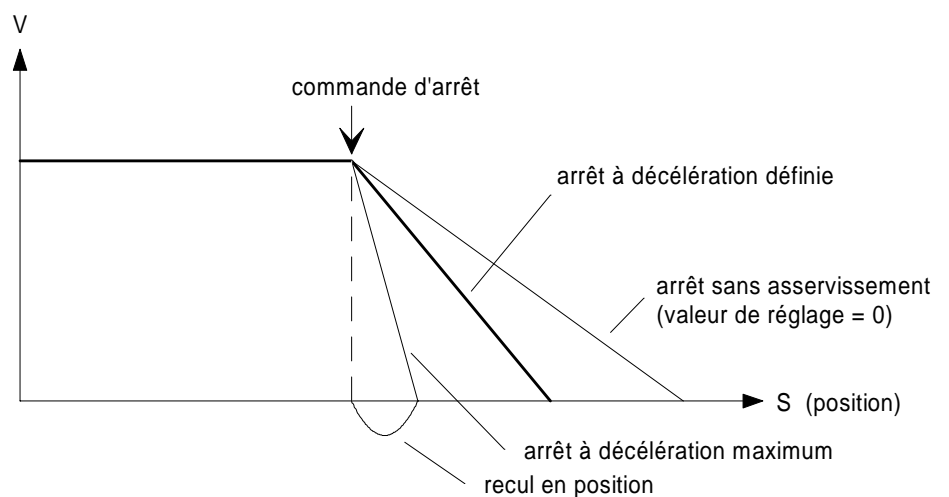
Bit 8 = "1" en cas de commande d'arrêt, la régulation est déclenchée, c'est-à-dire que la valeur de réglage est mise à zéro: le moteur n'est plus asservi.

Bit 9 = "1" en cas de commande d'arrêt, on freine à la décélération maximum, la position de consigne dans le registre de travail du contrôleur H3 est la même que la position actuelle ce qui déclenche un mouvement de recul.

Bit 10 = "1" en cas de commande d'arrêt, on freine à la décélération définie (accélération négative).

Un seul de ces trois bits peut être actif ("1") à un moment donné. Après un arrêt, le contrôleur ne perd pas la dernière position cible chargée. Toutefois avant de pouvoir poursuivre le mouvement interrompu sans charger de nouveau paramètre (position cible, vitesse ou accélération), il convient de charger à nouveau le type d'exploitation ("FSeIOM").

Le diagramme ci-dessous indique les trois types d'arrêt :

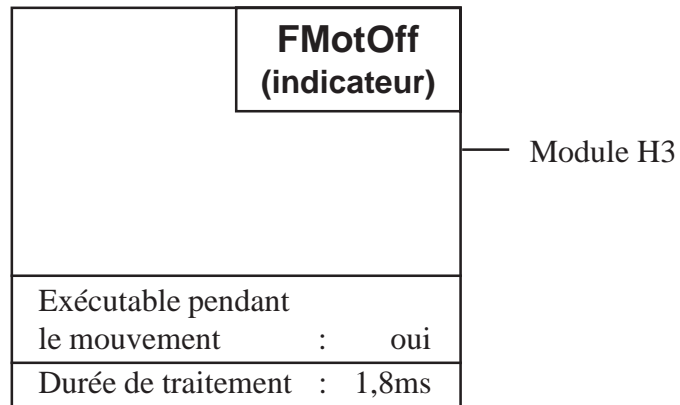


**FMotOff**

Fonction : - **Motor Off**  
 - Déclencher la régulation  
 (mise hors asservissement)

**FMotOff**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

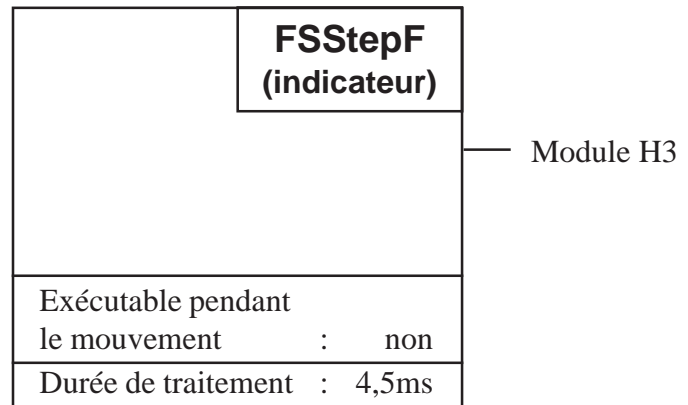
Cette fonction arrête la régulation, c'est-à-dire que la sortie de grandeur de réglage est mise à zéro. La commande "FMotOff" a la même fonction que la commande "FStop" lorsque le bit 8 du registre "MCW" est à "1".

**FSStepF**

Fonction : - **Single Step Forward**  
- Avance d'un pas

**FSStepF**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction provoque l'avance d'une impulsion unique dans le sens de déplacement positif. Une position cible relative de + 1 impulsion est chargée et le mouvement commence. Il faut veiller lors de l'exécution à la plage des valeurs de la position cible. La fonction ne doit pas être exécutée lorsque la position cible absolue a atteint la limite positive de la plage des valeurs.

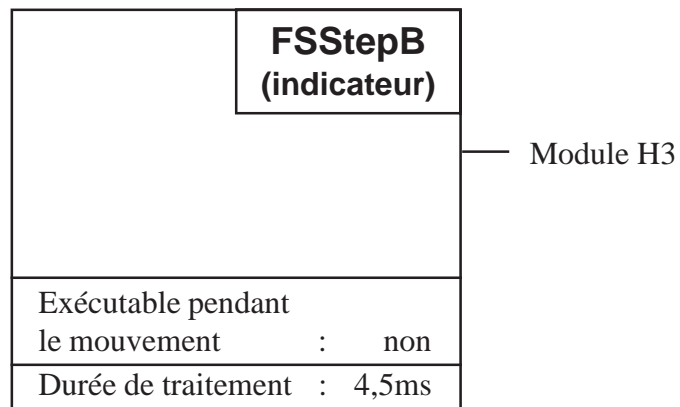
Attention : 1 impulsion correspond à 1/4 de pas d'encodeur  
(multiplication par 4 des impulsions au décodeur de position).

**FSStepB**

Fonction : - **Single Step Backwards**  
- Recul d'un pas

**FSStepB**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction provoque le déplacement d'une impulsion unique dans le sens de déplacement négatif. Une position cible relative de - 1 impulsion est chargée et le mouvement commence. Il faut veiller lors de l'exécution à la plage des valeurs de la position cible. La fonction ne doit pas être exécutée quand la position cible absolue a atteint la limite négative de la plage des valeurs.

Attention : 1 impulsion correspond à 1/4 de pas d'encodeur  
(multiplication par 4 des impulsions au décodeur de position).

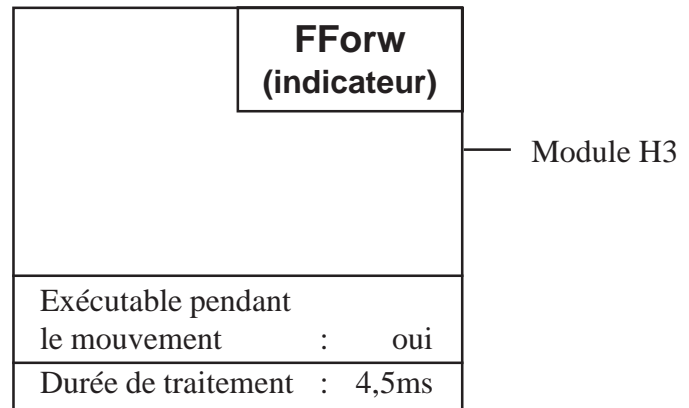


**FForw**

Fonction : - **F**orward with defined velocity  
 - Marche avant à vitesse définie

**FForw**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

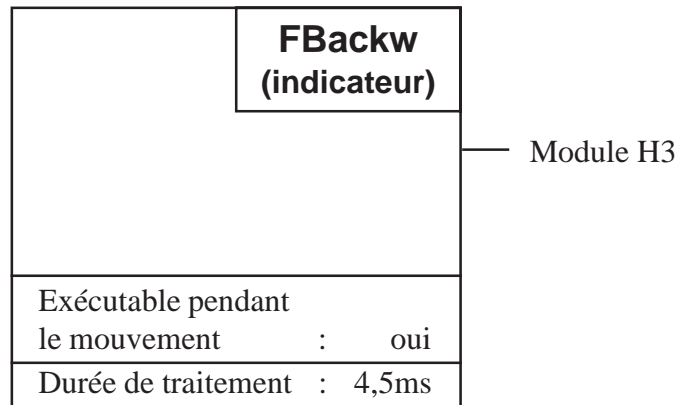
Cette fonction provoque l'avance à la vitesse chargée en dernier dans le sens de mouvement positif. Pour arrêter à nouveau le mouvement, il faut exécuter une commande d'arrêt manuelle. La fonction est réalisée en ce sens que c'est la position cible positive la plus élevée possible qui est chargée; ensuite, une commande de démarrage est émise.

**FBackw**

Fonction : - **Backwards** with defined velocity  
- Marche arrière à vitesse définie

**FBackw**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

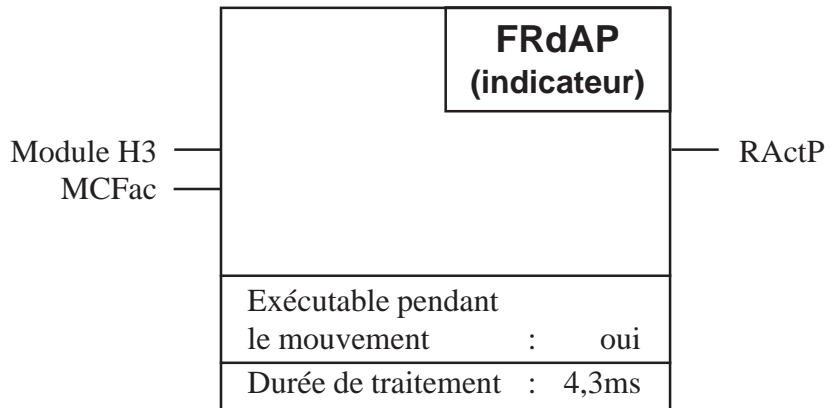
Cette fonction provoque le recul à la vitesse chargée en dernier dans le sens de mouvement négatif. Pour arrêter à nouveau le mouvement, il faut exécuter une commande d'arrêt manuelle. La fonction est réalisée en ce sens que c'est la position cible négative la plus élevée possible qui est chargée; ensuite, une commande de démarrage est émise.

# FRdAP

Fonction : - **Read Actual Position**  
 - Lire la position effective (actuelle)

# FRdAP

Paquet logiciel : PCD9.H3E1



### Description de la fonction

Cette fonction lit la position effective du module H3 et la copie dans le registre "RActP".

### Description des entrées et des sorties

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
RActP	<b>Register Actual Position</b> Position effective (actuelle)  Valeur: $[-2^{30}..+(2^{30}-1)]/k*10^{-3}$ Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. $9,223371*10^{18}$

### Facteur mécanique k dans le registre "MCFac" :

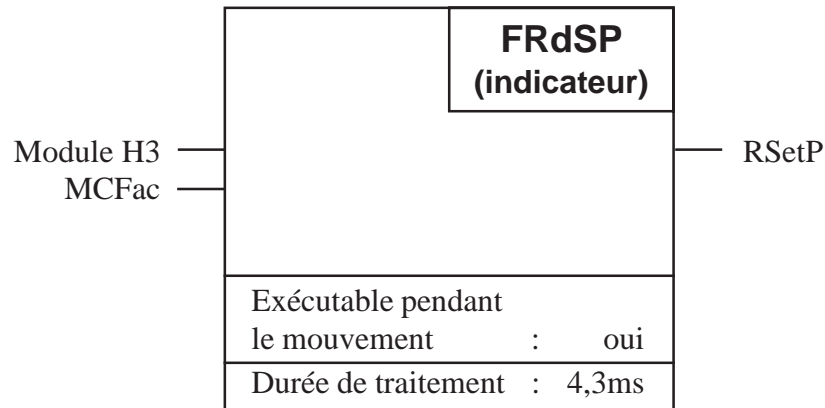
Le facteur k a la même signification que dans le cas de la fonction "FLdDA". Le registre "MCFac" est lu par la fonction ci-dessus et il est utilisé pour la conversion de la position à partir d'un nombre d'impulsions en une mesure métrique.

**FRdSP**

Fonction : - **Read Setpoint Position**  
- Lire la consigne de position

**FRdSP**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction lit la position de consigne instantanée à la sortie du générateur du profil de vitesse et la copie dans le registre "RSetP". La différence entre cette position de consigne et la position réelle est transmise au régulateur PID.

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Paramètre	Données		
			Type	Format	Valeur
RSetP	<b>Register Setpoint Position</b> Consigne de position  Valeur: $[-2^{30}..+(2^{30}-1)]/k*10^{-3}$ Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. $9,223371*10^{18}$

**Facteur mécanique k dans le registre "MCFac" :**

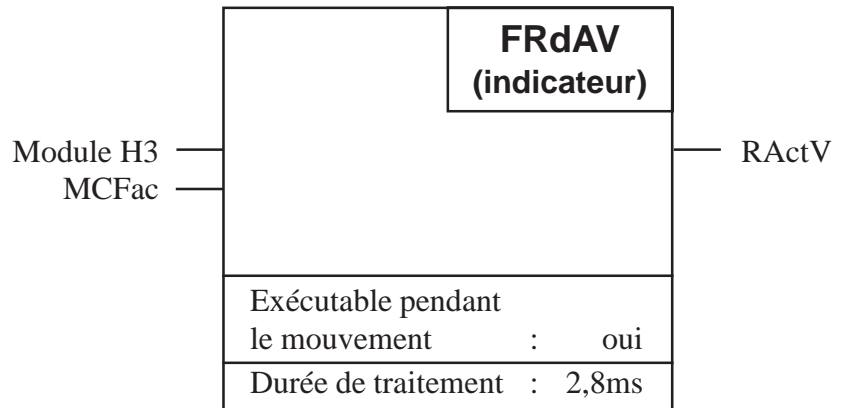
Le facteur k a la même signification que dans le cas de la fonction "FLdDA". Le registre "MCFac" est lu par la fonction ci-dessus et il est utilisé pour la conversion de la position à partir d'un nombre d'impulsions en une mesure métrique.

# FRdAV

Fonction : - **Read Actual Velocity**  
 - Lire la vitesse effective (actuelle)

# FRdAV

Paquet logiciel : PCD9.H3E1



### Description de la fonction

Cette fonction lit la vitesse réelle de l'axe du module H3 et la copie dans le registre "RActV". Le contrôleur ne peut toutefois lire dans le module H3 que les 14 bits les plus significatifs de la vitesse effective. Pour les petites vitesses, il en découle qu'aucune valeur raisonnable ne peut être lue et il est recommandé de lire la vitesse de consigne au lieu de la vitesse réelle.

### Description des entrées et des sorties

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
RActV	<b>Register Actual Velocity</b> Vitesse effective (actuelle)  Valeur: $[0..+(2^{30}-1)]/k*22348*10^{-6}$ Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. $9,223371*10^{18}$

### Facteur mécanique k dans le registre "MCFac" :

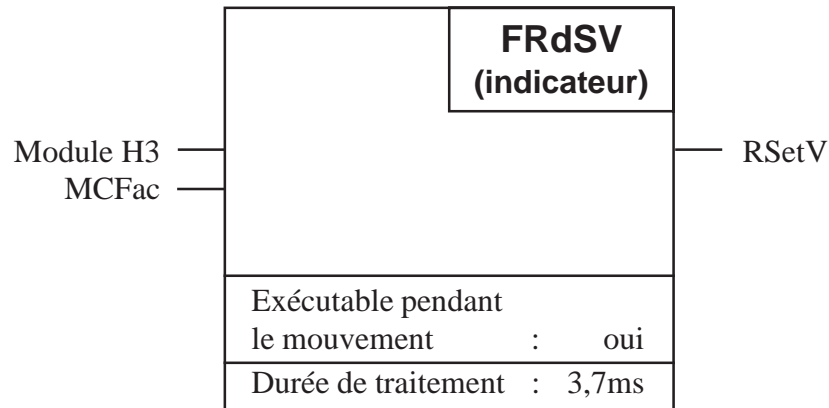
Le facteur k a la même signification que dans le cas de la fonction "FLdDA". Le registre "MCFac" est lu par la fonction ci-dessus et il est utilisé pour la conversion de la vitesse à partir d'un nombre d'impulsions /s en une mesure métrique/s.

**FRdSV**

Fonction : - **Read Setpoint Velocity**  
- Lire la consigne de vitesse

**FRdSV**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction lit la vitesse de consigne instantanée du générateur de profil et la copie dans le registre "RSetV".

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
RSetV	<b>Register Setpoint Velocity</b> Consigne de vitesse  Valeur: $[-2^{30}..+(2^{30}-1)]/k*22348*10^{-6}$ Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. $9,223371*10^{18}$

**Facteur mécanique k dans le registre "MCFac" :**

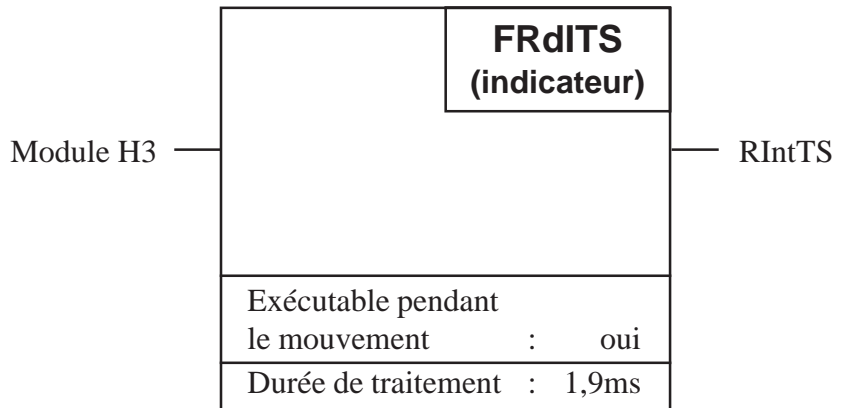
Le facteur k a la même signification que dans le cas de la fonction "FLdDA". Le registre "MCFac" est lu par la fonction ci-dessus et il est utilisé pour la conversion de la vitesse à partir d'un nombre d'impulsion /s en une mesure métrique/s.

**FRdITS**

Fonction : - **Read Integration Term Sum**  
 - Lire la somme d'intégration

**FRdITS**

Paquet logiciel : PCD9.H3E1



**Description de la fonction**

Cette fonction lit la valeur d'intégration ( $\sum e[n]$ ) du régulateur PID et la copie dans le registre "RIntS". Cette fonction est utilisée avant tout pour la détermination des paramètres de réglage P.I.D pendant la mise en service.

**Description des entrées et des sorties**

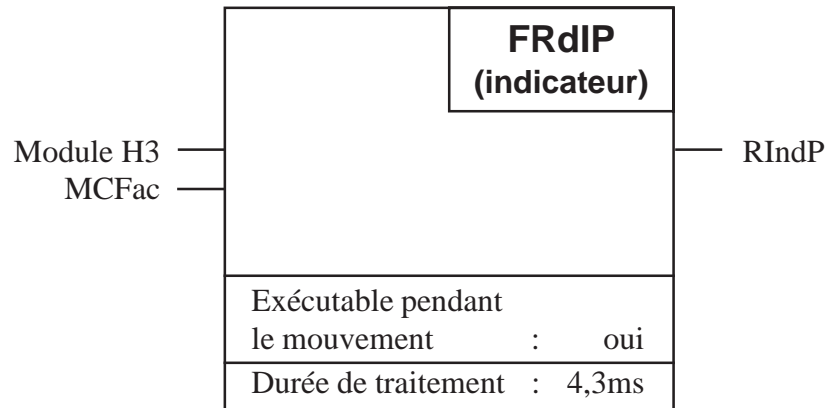
Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
RIntTS	<b>Register Integration Term Sum</b> Somme d'intégration  Valeur: La valeur se situe dans le domaine de la limite d'intégration définie au moyen de la fonction "FLdRP" (registre "IntL").	non	R	entier	--

**FRdIP**

Fonction : - **Read Index Position**  
- Lire la position d'index

**FRdIP**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction lit la position d'index à partir du registre de position d'index du module H3 et la copie dans le registre "RIndP" (voir aussi la fonction "FSetIP").

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
RIndP	<b>Register Index Position</b> Position d'index  Valeur: $[(-2^{30} \dots + (2^{30} - 1))] / k * 10^{-3}$ Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. $9,223371 * 10^{18}$

**Facteur mécanique k dans le registre "MCFac" :**

Le facteur k a la même signification que dans le cas de la fonction "FLdDA". Le registre "MCFac" est lu par la fonction ci-dessus et il est utilisé pour la conversion de la position à partir d'un nombre d'impulsions en une mesure métrique.

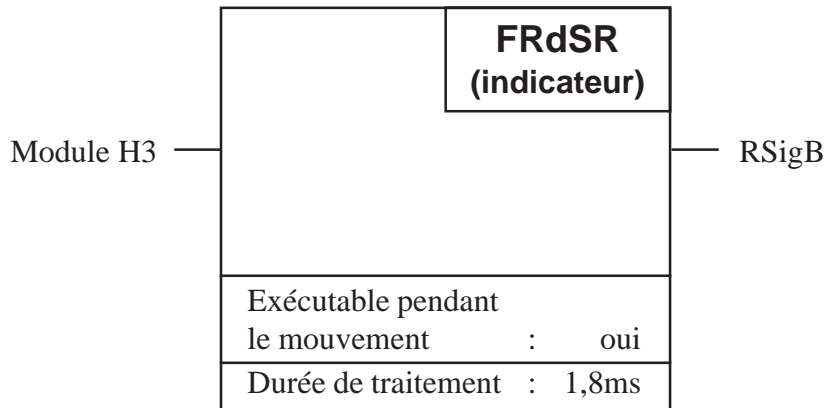


# FRdSR

Fonction : - **Read Signal Register**  
 - Lire le registre de signalisation

# FRdSR

Paquet logiciel : PCD9.H3E1



### Description de la fonction

Cette fonction permet la lecture du registre de signalisation d'un axe du module H3.

### Description des entrées et des sorties

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
RSigB	<b>Register Signalisation Bits</b> Registre de signalisation	non	R	entier	voir page suivant

Chacun des bits du registre "RSigB" a la signification suivante :

- Bit 0: Il est mis à "1" après l'exécution de la fonction "SetIP" (détermination de la position d'index). Le bit est remis à zéro après l'enregistrement de la prochaine position d'index (impulsion d'index).
- Bit 1: pas de signification
- Bit 2 à 6: Ils indiquent l'état des indicateurs d'état (voir la fonction "FResSF").
- Bit 7: Il est mis à "1" quand l'asservissement du moteur est arrêté (sortie de grandeur de réglage = 0).  
La régulation est déclenchée par les événements suivants :  
  - enclenchement de l'alimentation
  - après le traitement du FB "AxInit"
  - en cas de dépassement de la tolérance de position (si ainsi défini)
  - après exécution de la fonction "FMotOff" (commande d'arrêt du moteur)
  - après exécution de la fonction "FStop" (lorsque le type d'arrêt est ainsi défini)
Ce bit est remis à zéro avec la prochaine commande de démarrage ("FStart").
- Bit 8: Il est mis à "1" lors de l'enclenchement de l'alimentation ou lorsque la sortie de grandeur de réglage est définie en tant que sortie PWM avec le FB "AxInit". Ce bit est mis à zéro quand la sortie est définie comme sortie analogique  $\pm 10$  V au moyen du FB "AxInit".
- Bit 9: Il indique la mesure définie en cas de dépassement de l'erreur de position maximum.  
"0" --> seul l'indicateur d'état "ExcEr" est mis à "1"  
"1" --> l'indicateur d'état est mis à "1" et la régulation est déclenchée (plus d'asservissement du moteur).
- Bit 10: Il est mis à "1" quand le générateur a terminé le profil de vitesse calculé. Ce bit est remis à zéro lors de la prochaine commande de démarrage ("FStart").
- Bit 11: Il indique le mode d'exploitation sélectionné au moyen de la commande "FSeIOM" :  
"0" --> mode de régulation "position"  
"1" --> mode de régulation "vitesse de rotation"  
La mise à "1" ou à "0" de ce bit se fait à l'occasion de la prochaine commande de démarrage.

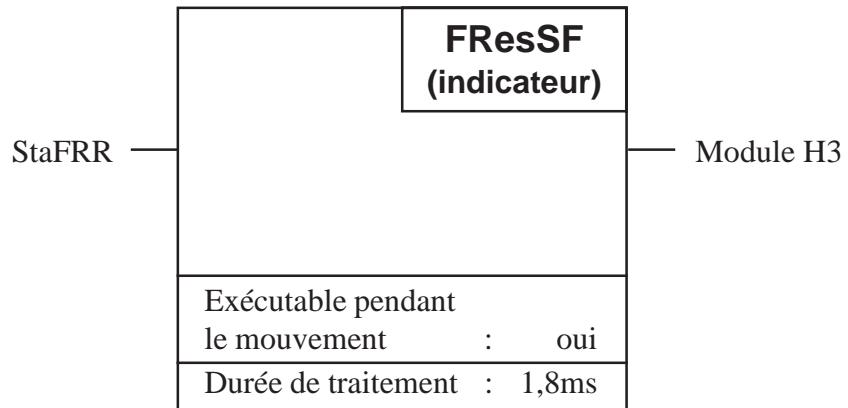
- Bit 12: Il indique le sens de rotation en mode "vitesse de rotation" :  
"1" --> en avant  
"0" --> en arrière  
La mise à "1" ou à "0" de ce bit se fait à l'occasion de la prochaine commande de démarrage.
- Bit 13: pas de signification
- Bit 14: Il indique qu'une nouvelle accélération a été chargée au moyen des fonctions "FLdAA/R". Ce bit est remis à zéro lors de la prochaine commande de démarrage.
- Bit 15 à 31: pas de signification

**FResSF**

Fonction : - **Reset Status Flag**  
- Reset indicateur d'état

**FResSF**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

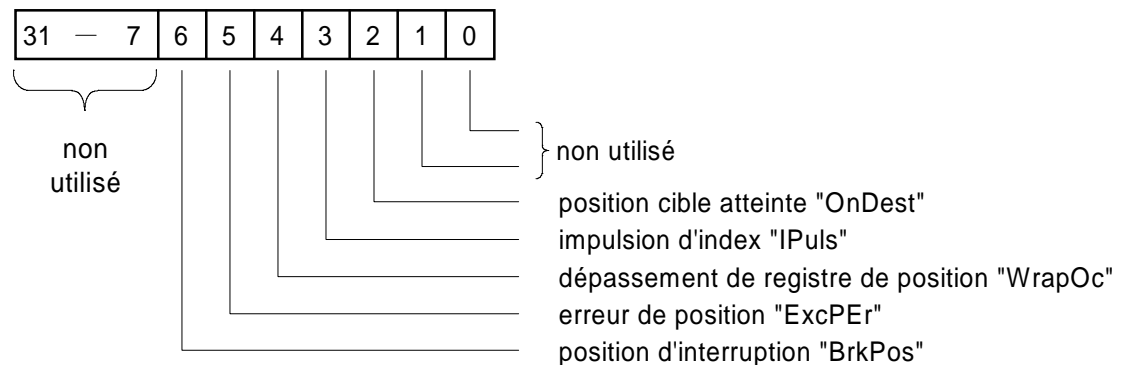
Cette fonction remet à zéro l'indicateur d'état d'un axe. Les indicateurs peuvent être remis à zéro isolément ou tous simultanément.

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
StaFRR	<b>Status Flag Reset Register</b>	non	R	binaire	voir page suivant

La fonction lit à partir du registre "StaFRR" quels sont les indicateurs d'états qui doivent être remis à zéro. L'indicateur est remis à zéro quand le bit correspondant du registre est à zéro (masque).

## Signification du registre "StaFRR"



## Signification des indicateurs d'état

Bit 2 : "OnDest" **On Destination** / position cible atteinte

Il est mis à "1" par les événements suivants :

- le générateur a terminé le profil de vitesse calculé; il peut arriver à la suite d'un paramètre erroné ou d'un problème mécanique que le moteur ne se trouve pas encore définitivement en position cible et que l'indicateur soit quand même mis à "1" étant donné que le générateur a déjà terminé le profil de consigne;
- la régulation est déclenchée (par exemple après la fonction "FMotOff")
- après un arrêt manuel (fonction "FStop")

Cet indicateur est automatiquement remis à "0" à l'occasion d'une commande de démarrage (fonction "FStart").

Bit3 : "IPuls" **Index Puls** / impulsion d'index enregistrée

Il est mis à "1" quand une impulsion d'index a été enregistrée et que l'on a écrit dans le registre de position d'index du module H3 la position effective (voir aussi la fonction "FSetIP").

Bit 4 : "WrapOc" **Wraparound Occured** / débordement de registre de position

Il est mis à "1" quand intervient un débordement du registre de position. Il peut y avoir débordement en mode d'exploitation à régulation de la vitesse de rotation.

Bit 5 : "ExcPEr" **Excessive Position Error** / dépassement d'erreur de position

Il est mis à "1" dès que l'erreur de position dépasse la tolérance définie au moyen de la fonction "FSetPE".

Bit 6 : "BrkPos" **Break Position** / position d'interruption

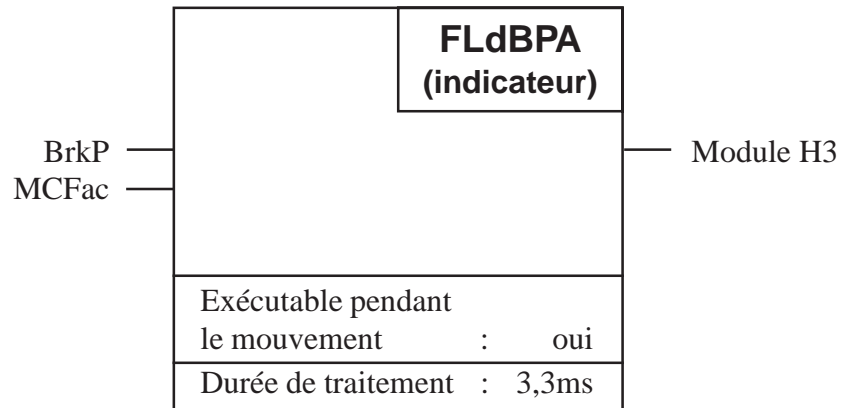
Il est mis à 1 dès que la position atteint ou dépasse la position d'interruption chargée au moyen des fonctions "FLdBPA/R".

# FLdBPA

Fonction : - **Load Break Position Absolute**  
 - Charger la position d'interruption absolue

# FLdBPA

Paquet logiciel : PCD9.H3E1



### Description de la fonction

Cette fonction charge une position d'interruption absolue dans le module H3. Chargement absolu signifie que la valeur est relative à la position zéro. La position chargée est immédiatement reprise par le module H3 dans le registre de travail. Si la position d'interruption est atteinte, l'indicateur d'état "BrkPos" est mis à "1".  
 L'indicateur peut être remis à zéro au moyen de la fonction "ResSF".  
 Cette fonction permet d'obtenir un message à une position déterminée afin, par exemple, de modifier la vitesse ou les paramètres de réglage.

### Description des entrées et des sorties

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
BrkP	<b>Break Position</b> Position d'interruption atteinte  Valeur: $[(-2^{30}..+(2^{30}-1)]/k*10^{-3}$ Unité: déterminée par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. $9,223371*10^{18}$

**Facteur mécanique k dans le registre "MCFac" :**

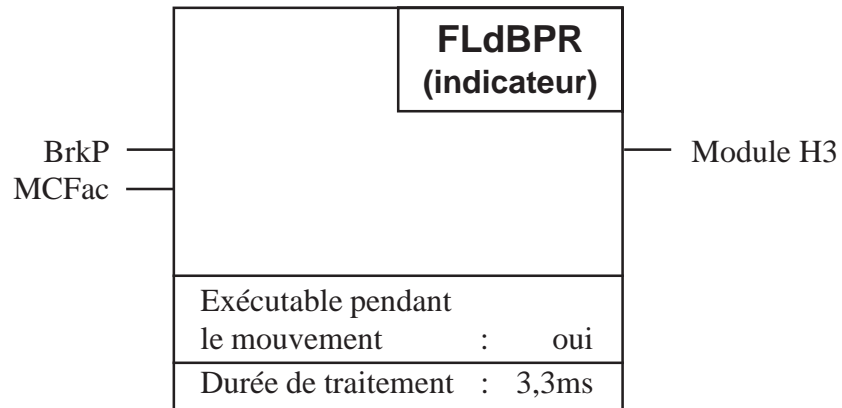
Le facteur k a la même signification que dans le cas de la fonction "FLdDA". Il faut tenir compte de ce que ce facteur est unique pour la position cible, la vitesse et l'accélération. Il est donc recommandé de sélectionner les mêmes unités pour l'introduction de ce paramètre.

**FLdBPR**

Fonction : - **Load Break Position Relative**  
 - Charger la position d'interruption relative

**FLdBPR**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Cette fonction charge dans le module H3 une position relative d'interruption. Chargement relatif signifie que la valeur est relative à la position cible instantanée. Il faut tenir compte dans ce contexte que la position d'interruption négative additionnée à la position cible ne dépasse pas la plage de valeurs valable pour la position cible. La position chargée est immédiatement reprise dans le registre de travail du module H3. Si la position d'interruption est atteinte, l'indicateur d'état "BrkPos" est mis à "1". L'indicateur peut être remis à zéro à l'aide de la fonction "ResSF". Cette fonction permet d'obtenir un message à une position déterminée afin, par exemple, de modifier la vitesse ou les paramètres de réglage.

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
BrkP	<b>Break Position</b> Position d'interruption atteinte  Valeur: $[(-2^{30}..+(2^{30}-1)]/k*10^{-3}$ Unité: déterminer par k	non	R	entier	--
MCFac	<b>Motion Control Factor</b> Facteur mécanique	non	R	virgule flottante	0.. 9,223371*10 <sup>18</sup>



**Facteur mécanique k dans le registre "MCFac" :**

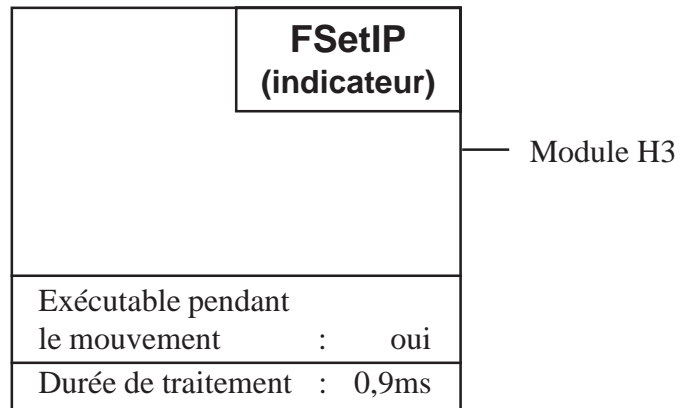
Le facteur k a la même signification que dans le cas de la fonction "FLdDA". Il faut tenir compte de ce que ce facteur est unique pour la position cible, la vitesse et l'accélération. Il est donc recommandé de sélectionner les mêmes unités pour l'introduction de ce paramètre.

**FSetIP**

Fonction : - **Set Index Position**  
- Saisie de la position d'index

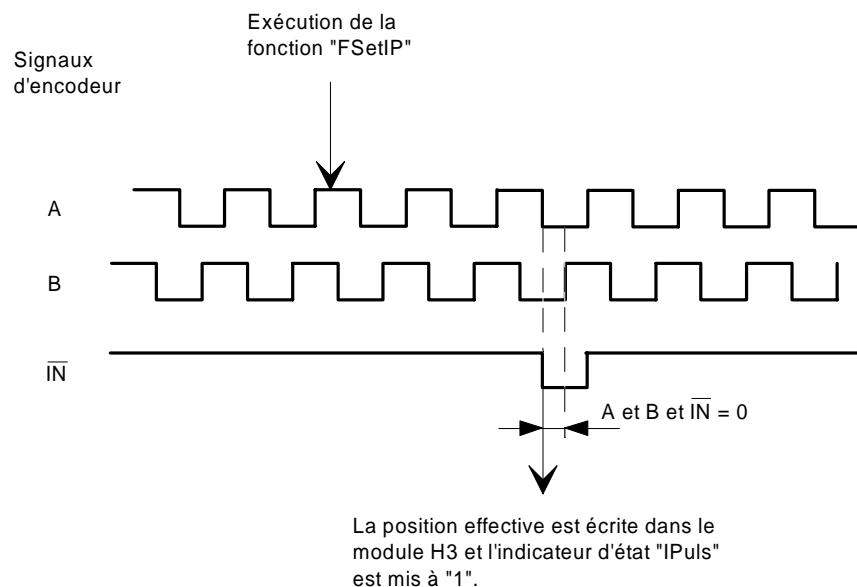
**FSetIP**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

Après l'exécution de cette fonction, la prochaine fois que les signaux d'encodeur A et B ainsi que l'entrée d'impulsion d'index sont à l'état zéro, la position réelle est écrite dans le registre de position d'index sur le module H3. Une fois que la position d'index est enregistrée, l'indicateur d'état "IPuls" est mis à "1". La position d'index peut être lue du module H3 au moyen de la fonction "FRdIP".

Le diagramme ci-dessous indique le déroulement dans le module H3 en relation avec les signaux d'encodeur, après l'exécution de cette fonction.

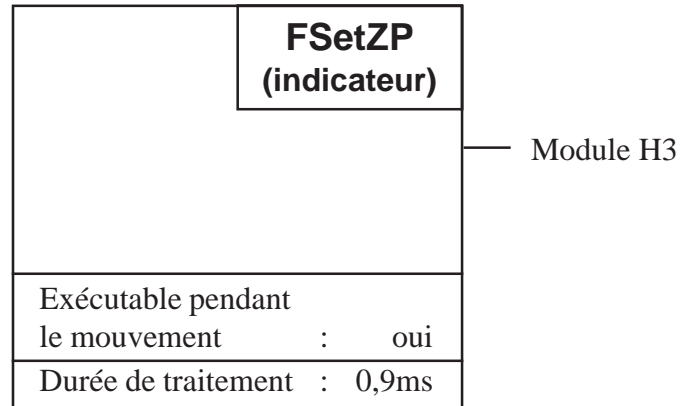


**FSetZP**

Fonction : - **Set Zero Position**  
 - Déterminer la position zéro

**FSetZP**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

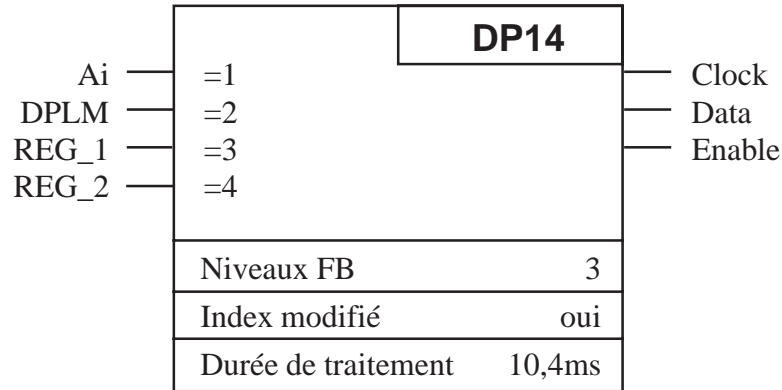
Cette fonction définit la position réelle en tant que position zéro (origine). Si la fonction est exécutée pendant un mouvement, la position cible momentanée n'est pas affectée tant qu'aucune commande de démarrage "FStart" n'est exécutée.

# DP14

Fonction : - Display Contents of Register on PCA2.D14  
 - Affichage de la valeur de registre

# DP14

Paquet logiciel : PCD9.H3E1



### Description de la fonction

Cette fonction permet d'afficher une valeur de registre avec 1\* 10 chiffres ou avec 2\*6 chiffres sur le module d'affichage PCA2.D14. Les formats d'affichage suivants sont possibles :

**1\*10 chiffres** (1 registre)

b	v	1	2	3	4
5	6	7	8	9	10

b = blank ; v = blank ou "-"  
 1 .. 10 = chiffres

plage d'affichage = plage de valeur du registre ± 2'147'483'647

**2\*6 chiffres** (2 registres)

v	x	x	x	x	x
v	x	x	x	x	x

w = contenu du registre  
 0 ≤ w ≤ + 999'999 : v = MSD  
 w > + 999'999 : v = "A"  
 0 > w ≥ - 99'999 : v = "-"  
 w < - 99'999 : v = "U"

plage d'affichage : - 99'999 ... + 999'999

2 modules d'affichage peuvent être pilotés par le module PCD4.H320.

L'indicateur DPLM détermine le type d'affichage :

DPLM = "0" --> 1\*10 chiffres  
 DPLM = "1" --> 2\* 6 chiffres

Paramètre: REG\_1 : registre pour la première valeur affiché  
 REG\_2 : registre pour la deuxième valeur affiché

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
Ai	Numéro d'axe i		K	entier	1.. 32
DPLM	Mode d'affichage		F	binaire	0, 1
REG_1	première valeur d'affichage		R	entier	0.. 10 <sup>9</sup> -1
REG_2	deuxième valeur d'affichage		R	entier	0.. 10 <sup>6</sup> -1
Clock	Sortie vers D14		O	binaire	0, 1
Data	" "		O	binaire	0, 1
Enable	" "		O	binaire	0, 1

**Exemple de programme** pour l'affichage de la position effective et de la vitesse de consigne pour l'axe 1 avec un PCA2.D14 :

```

|
ACC H
SET F DplM+FA1 ; Display Mode = 2*6 Digits
CFB DP14 ; Display on PCA2.D14
      1 ; Axis 1
      F DplM+FA1 ; Display Mode
      R RActP+RA1 ; Actual Position
      R RSetV+RA1 ; Setpoint Position
|

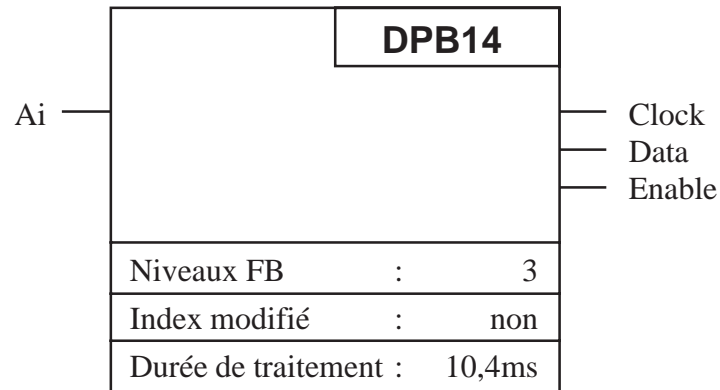
```

**DPB14**

Fonction : - Clear Display on PCA2.D14  
- Effacement de l'affichage

**DPB14**

Paquet logiciel : PCD9.H3E1

**Description de la fonction**

La commande "DPB14" remplace tous les chiffres de l'affichage PCA2.D14 par des blancs, ce qui revient à les effacer.

**Description des entrées et des sorties**

Symbole	Désignation / Fonction	Para- mètre	Données		
			Type	Format	Valeur
Ai	Numéro d'axe i		K	entier	1.. 32
Clock	Sortie vers D14		O	binaire	0, 1
Data	" "		O	binaire	0, 1
Enable	" "		O	binaire	0, 1

## 8. Recherche et traitement des erreurs

---

Le contrôleur H3 signale une erreur en mettant à "1" l'indicateur d'erreur de commande. L'indicateur est armé quand le contrôleur ne peut interpréter une commande ou des données qui lui sont adressées.

C'est le plus souvent une erreur de programmation de l'utilisateur qui est à l'origine d'une erreur de commande; par exemple, on tente de charger une valeur dans le module H3 qui se situe en dehors de la plage de valeurs autorisées.

Exemples d'erreur :

- chargement d'une vitesse négative avec la fonction "FLdVA" (chargement d'une vitesse absolue),
- le facteur mécanique k n'est pas chargé dans le registre "MCFac" en format virgule flottante --> ceci peut provoquer un débordement de la plage de valeurs lors de la conversion de l'unité d'un paramètre,
- un facteur de réglage > 32767 est chargé dans le module H3,
- tentative de chargement d'un paramètre de mouvement relatif pour le premier mouvement, avant la première commande de démarrage ("FLdDR", "FLdVR", "FLdAR" et "FLdBPR"),
- dérangement sur le bus PCD4.

### Traitement d'une erreur de commande

En cas d'erreur, le contrôleur H3 ignore la commande envoyée et met l'indicateur d'erreur à "1". Cet indicateur est surveillé de façon autonome par toutes les fonctions qui communiquent avec le module H3. L'utilisateur n'a aucun accès direct à l'indicateur d'erreur. Si une erreur se produit, la dernière commande envoyée est répétée deux fois au maximum. Après la troisième erreur successive, le FB de traitement des erreurs "ComErS" est appelé et l'on passe, après son traitement, au pas suivant du programme. Le FB "ComErS" se trouve dans le fichier H3FB.SRC. Il peut être programmé à volonté par l'utilisateur. C'est ainsi l'utilisateur lui-même qui détermine en cas d'erreur les mesures qui doivent être prises. Un appel à ce FB signifie dans tous les cas qu'une commande n'a pas été exécutée; en conséquence, le déroulement correct du programme H3 ne peut plus être garanti. Il est recommandé, pour la mise en service, de provoquer l'appel de l'XOB 13 par le biais d'une division par zéro.

Exemple :

```

FB          ComErS      ; Command Error Stop

DIV        R 0          ;
           K 0          ;   provoque l'appel d' XOB 13
           R 0          ;
           R 0          ;
  
```

EFB

Si la commande "DIAG" est programmée dans XOB 13, il est possible de déterminer rapidement où l'erreur s'est produite dans le programme utilisateur.

Exemple:

XOB 13

```

DIAG      R 1          ; registre de diagnostic
  
```

Déclencher éventuellement l'entraînement
---

HALT

EXOB

### Procédure de localisation de l'erreur

Déterminer en mode Debugger à l'aide du registre de diagnostic sur quelle commande s'est produite l'erreur.

Voir aussi à ce sujet la description de l'instruction PCD "DIAG".



## 9. Exemples didactiques d'application

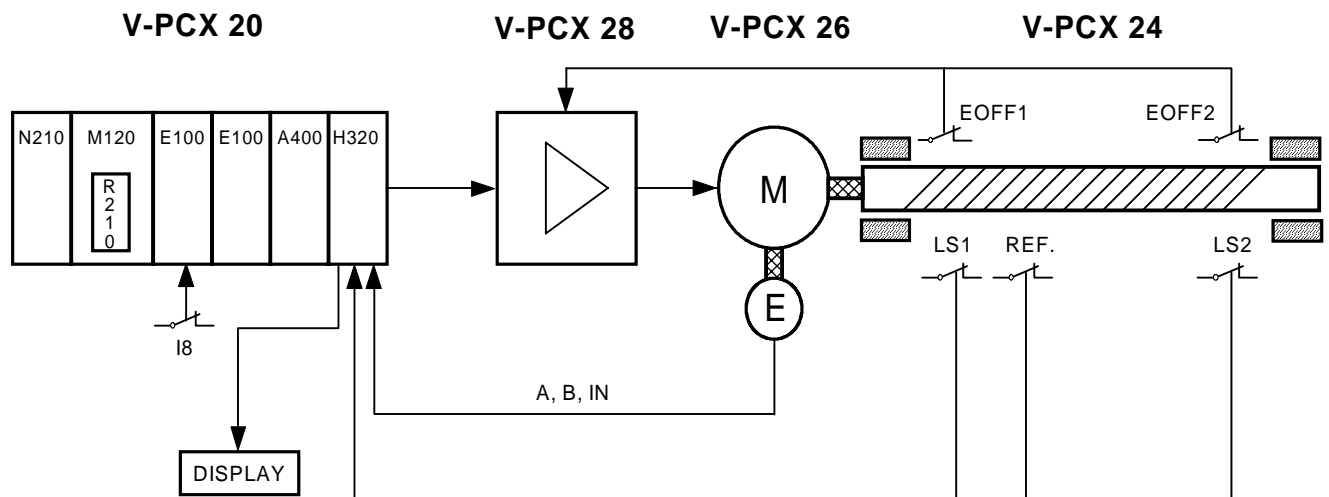
### 9.1 Exemple 1

Dans cet exemple, il s'agit d'une application très simple à un axe. Il montre quels pas doivent être exécutés et dans quel ordre afin d'effectuer un mouvement simple.

#### Enoncé

Matériel :

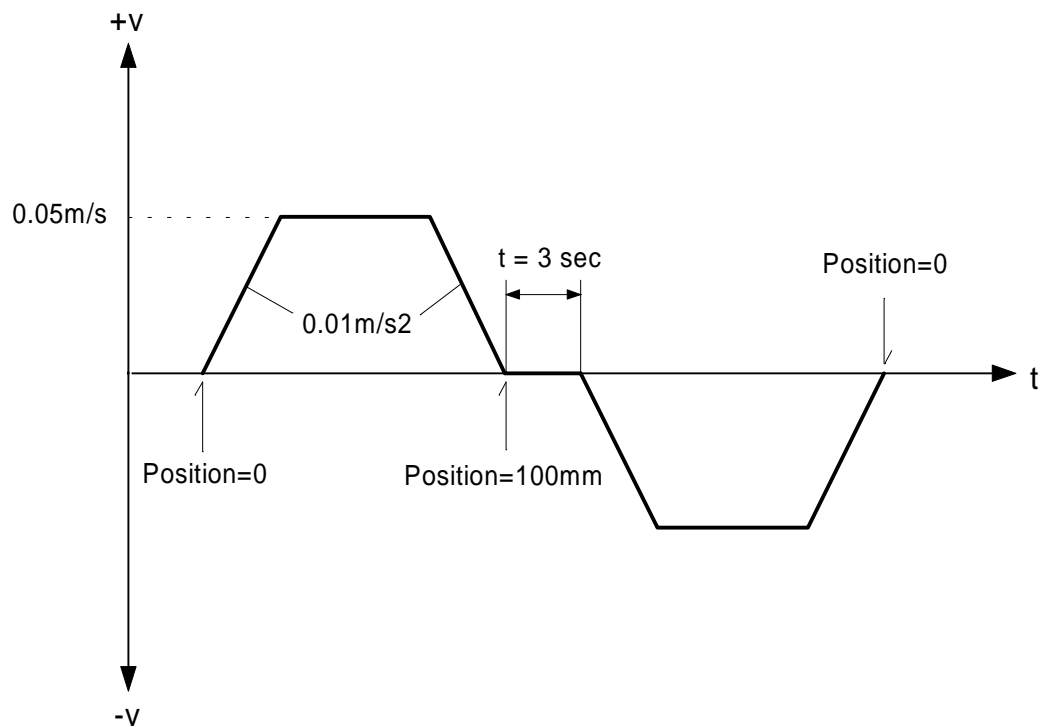
L'exemple repose sur le matériel des modèles d'atelier V-PCX 20, V-PCX 24, V-PCX 26 et V-PCX 28.



Données de l'axe :

- encodeur 500 pas/t
- broche à pas de 2 mm
- donnée de la position cible avec résolution de 1/10 mm
- vitesse maximum de l'entraînement 0,1 m/s
- accélération maximum autorisée 0,05 m/s<sup>2</sup>

Il faut obtenir le déroulement ci-dessous du mouvement :



Le mouvement doit être lancé au moyen de la touche connectée à l'entrée 8.

On admet que la position zéro a déjà été définie après l'enclenchement de l'automate; en conséquence, aucun mouvement de référence n'est nécessaire. Dans ce premier exemple, on renonce également à la surveillance de l'interrupteur de fin de course ainsi qu'à l'affichage au moyen du module d'affichage PCA2.D14.

## Solution

On dispose pour le problème énoncé du matériel cité et du paquet logiciel PCD9.H3E1. Nous partons de l'idée que l'ensemble du matériel a déjà été mis fonctionnellement en service. Les pas décrits ci-après doivent être inclus dans la programmation.

### a) Installation du logiciel

Les deux fichiers H3 H3DEF.SRC et H3FB.SRC sont copiés dans le répertoire de travail. En premier lieu, il faut configurer comme suit l'installation H3 dans le fichier H3DEF.SRC :

```
FMAH3 EQU 48 ; adresse de base du moduleH3
IMode EQU 6 ; init. du port de sortie (analogique)
MNA EQU 1 ; 1 axe
BAF EQU 200 ; adresse de base indicateur
BAR EQU 200 ; adresse de base registre
BAC EQU 40 ; adresse de base compteur
BAFB EQU 900 ; adresse de base blocs de fonctions
RA1 EQU 0*NoRfeA ; constante de bloc registres axe 1
FA1 EQU 0*NoFfeA ; constante de champ d'indicateurs axe 1
```

Etant donné que l'on ne travaille pas avec des affectations de symboles externes, le symbole

```
PUBLSYM EQU 0
```

est défini.

Toutes les autres affectations de symboles sont à laisser telles quelles.

On définit le symbole

```
EXTNSYM EQU 0
```

dans le fichier H3FB.SRC.

Cette définition fait que le fichier de définition des symboles H3DEF.SRC sera automatiquement inclus avec l'instruction \$INCLUDE lors de l'assemblage.

L'inclusion du fichier de définition dans le fichier de l'utilisateur et dans le fichier H3FB.SRC présente l'avantage de mettre à disposition dans le fichier liste les adresses absolues déjà après l'assemblage. Celles-ci sont nécessaires lorsqu'une valeur doit être affichée par le Debugger lors du test du programme.

Si, par contre, on travaillait avec des affectations de symboles externes, les adresses absolues ne seraient disponibles qu'après la génération du fichier DOC.

A la fin du fichier H3FB.SRC se trouve le FB "ComErS" qui est appelé lors de la survenance répétitive de "Command Errors". L'utilisateur peut déterminer dans ce FB les mesures à prendre en cas d'erreur. Il est recommandé de provoquer l'appel d'XOB 13 lors de la mise en service par le biais d'une division par zéro. On peut alors programmer la commande "DIAG" dans l'XOB 13. De cette manière, on peut rapidement déterminer où s'est produite l'erreur dans le programme utilisateur. Pour de plus amples détails, se référer au chapitre 8 sur la recherche des erreurs.

```

FB      ComErS   ; Command Error Stop
DIV     R  0
        K  0      ; provoque l'appel d'XOB 13
        R  0
        R  0
EFB

```

A part ces deux adaptations, le fichier H3FB.SRC ne doit pas être modifié.

Pour le pas suivant, le fichier est assemblé afin de déterminer si les adaptations mentionnées ont été correctement effectuées. Si l'assemblage a réussi, le fichier ne doit plus être assemblé pendant toute la programmation; plus tard, il doit seulement encore être lié avec le programme utilisateur.

## b) programme utilisateur

Nous nous tenons, pour la rédaction du programme, à la structure évoquée au chapitre 7.2

### 1. Initialisation dans XOB 16

Dans un premier temps, il faut déterminer les valeurs pour les registres d'initialisation suivants:

(les registres d'initialisation sont lus par le FB "AxInit")

#### Mot de contrôle de mouvement "MCW"

Le mode d'arrêt de fonctionnement et la mesure à prendre en cas d'erreur sont définis dans ce registre.

Type d'arrêt :    Voir à ce sujet la fonction "FStop"  
 Dans ce premier exemple, le type d'arrêt ne joue certes encore aucun rôle étant donné qu'aucune commande d'arrêt manuel n'est prévue dans le programme.  
 Nous définissons toutefois le bit 10 = "1" --> stop à décélération définie.

Mode de fonctionnement: Voir à ce sujet la fonction "FSeIOM"  
 Nous travaillons en mode de positionnement --> bits 11 et 12 = "0"

Erreur de position: Voir à ce sujet la fonction "FSetPE"  
 En cas d'erreur de position, seul l'indicateur d'état "ExcPEr" doit être mis à "1" --> valeur pour les bits 0 à 7 = 1BH (hexadécimal).

Registre "MCW"

Bit	31		13	12	11	10	9	8	7	0
	non utilisé		0	0	1	0	0			1BH

—> charger le registre "MCW" avec la valeur 41BH (hexadécimal).

**- Erreur de position "PosEr"** --> voir fonction "FSetPE"

Dans ce premier exemple, l'indication pour l'erreur de position n'a pas de signification étant donné que l'indicateur d'état "ExcPEr" n'est pas surveillé par le programme utilisateur (par exemple pour déclencher l'entraînement). Etant donné que l'erreur de position est toutefois chargée lors de l'initialisation dans le module H3 et qu'elle doit se situer dans les limites de la plage autorisée, nous définissons que la différence maximum entre la position de consigne et la position effective doit être d'une révolution pour mettre à "1" l'indicateur d'état "ExcPEr".

—> charger le registre "PosEr" avec la valeur 2000 = 4\*500 pas/t  
 (évaluation des flancs des impulsions d'encodeur)

**- Facteurs PID** --> voir la fonction "FLdRP"

Par souci de simplicité, nous montrons dans un autre exemple la procédure de détermination des facteurs PID.

Nous partons de l'hypothèse que ces facteurs ont déjà été enregistrés.

Les registres doivent être chargés avec les valeurs suivantes :

Facteur proportionnel "KProp"	150
Facteur intégral "KInt"	50
Facteur différentiel "KDer"	50
Limite d'intégration "IntL"	500
Durée d'échantillonnage partie D "Sampl"	15 (5,46 ms=16*0,341ms)

**- Facteur mécanique k "MCFac" --> voir fonction "FLdDA"**

Le facteur machine détermine l'unité pour les valeurs en entrée et en sortie pour une position, vitesse et accélération.

Dans l'énoncé du problème, on a demandé une résolution de 1/10 mm.

$$\text{--> } k = \frac{4 * In}{s} = \frac{4 \text{ imp/pas} * 500 \text{ pas/t.}}{20 * 1/10\text{mm/t.}} = 100 \text{ imp par } 1/10\text{mm}$$

—> charger le registre "MCFac" avec la valeur 100  
(en format virgule flottante)

**- Vitesse "Veloc" --> voir fonction "FldVA"**

Etant donné que de nombreuses application n'utilisent qu'une vitesse, on charge une vitesse absolue déjà lors de l'initialisation.

Dans notre exemple, la vitesse doit être de 0,05 m/s.

—> charger le registre "Veloc" avec la valeur 500 (unité de 1/10 mm/s)

**- Accélération "Accel" --> voir fonction "FLdAA"**

Selon l'énoncé, l'accélération doit être de 0,01 m/s<sup>2</sup>

—> charger le registre "Accel" avec la valeur 100 (unité de 1/10 mm/s<sup>2</sup>)

Les valeurs enregistrées à ce stade doivent être chargées dans XOB 16, dans le registre d'initialisation, avant l'appel du FB "AxInit".

## 2. Traitement cyclique des axes

Seul FB "AxHndlg" est appelé dans COB 0 pour le traitement des axes.

## 3. Définition du programme de déplacement

Le programme de déplacement est programmé dans une structure GRAFTEC (SB 0) en fonction du profil donné de vitesse / temps.

Les pages suivantes contiennent le fichier source (BSP01.SRC) du programme utilisateur de cet exemple.

```

; Demo programm for the motion control module PCD4.H3..
; =====
; Name   : BSP01.SRC
; U. Jäggi 21.08.90

$ include H3DEF.SRC

;***** Cold-Start (Initialisation)
XOB      16
;----- Cold-Start Definitions
;----- loading of the initialisation registers

Ld       R  MCW+RA1    ; Motion Control Word
          41BH        ; Stop smoothly, Position mode
                   ; only statusflag (Pos.error)
Ld       R  PosEr+RA1 ; Position Error
          2000        ; 4 * 500 pulses
Ld       R  KProp+RA1 ; Proportional factor
          150
Ld       R  KInt+RA1  ; Integral factor
          50
Ld       R  KDer+RA1  ; Derivative factor
          50
Ld       R  IntL+RA1  ; Integration Limit
          500
Ld       R  SampI+RA1 ; Sampling Interval
          15          ; 5.46ms
Ld       R  MCFac+RA1 ; Motion Control Factor
          100.0       ; 100 Imp./1/10mm
Ld       R  Veloc+RA1 ; Velocity
          500         ; 0.05m/s
Ld       R  Accel+RA1 ; Acceleration
          100         ; 0.01m/s2

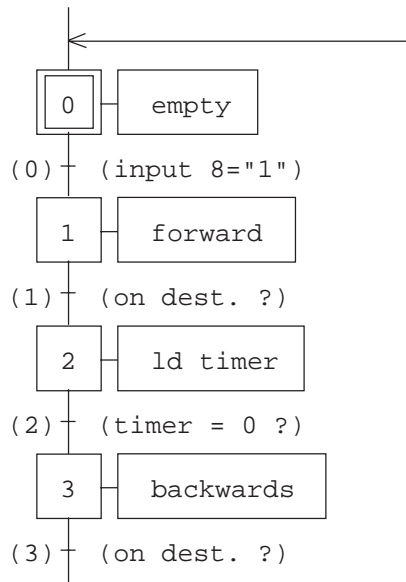
;-----
CFB      AxInit      ; Axis Initialisation
          1           ; X axis
          RA1
          FA1
          IMode      ; Initialisation mode: Analog/PWM
;----- End XOB 16
EXOB

;***** Cyclic program
COB      0
          0
;-----
CFB      AxHndlg     ; Axis Handling
          1           ; X axis
          RA1
          FA1
;-----
CSB      0           ; Call of the motion control program
;----- ; End cyclic program
ECOB

```

\*\*\* SAIA PCD GRAFTEC EDITOR \$113 \*\*\*  
FILE: BSP01.GLS (29.08.90 10.51)  
\*\*\* SAIA AG - CH-3280 MURTEN \*\*\*  
SB-NUMBER: 0  
PAGE-NB: 0

PAGE: 1  
PRODUCED: 29.08.90 10.51





```

;***** Motion control program
SB      0
;-----
IST      0      ; empty
          I  3      ; on dest. ?
          O  0      ; input 8="1"
EST
;-----
ST       1      ; forward
          I  0      ; input 8="1"
          O  1      ; on dest. ?
ld      r  DestP+RA1 ; Destination Position
          1000      ; 100mm = 1000*1/10mm
set     f  FLdDA+FA1 ; Load Destination Absolute
set     f  FStart+FA1 ; Start motion
EST
;-----
ST       2      ; ld timer
          I  1      ; on dest. ?
          O  2      ; timer = 0 ?
ld      t  0      ; timer 0
          30      ; 3s
EST
;-----
ST       3      ; backwards
          I  2      ; timer = 0 ?
          O  3      ; on dest. ?
ld      r  DestP+RA1 ; Destination Position
          0      ; Position 0
set     f  FLdDA+FA1 ; Load Destination Absolute
set     f  FStart+FA1 ; Start motion
EST
;-----
TR       0      ; input 8="1"
          I  0      ; empty
          O  1      ; forward
sth     i  8      ; motion free ?
ETR
;-----
TR       1      ; on dest. ?
          I  1      ; forward
          O  2      ; ld timer
sth     f  OnDest+FA1 ; On Destination ?
ETR
;-----
TR       2      ; timer = 0 ?
          I  2      ; ld timer
          O  3      ; backwards
stl     t  0      ; timer = 0 ?
ETR
;-----
TR       3      ; on dest. ?
          I  3      ; backwards
          O  0      ; empty
sth     f  OnDest+FA1 ; On Destination ?
ETR
;-----
ESB

```

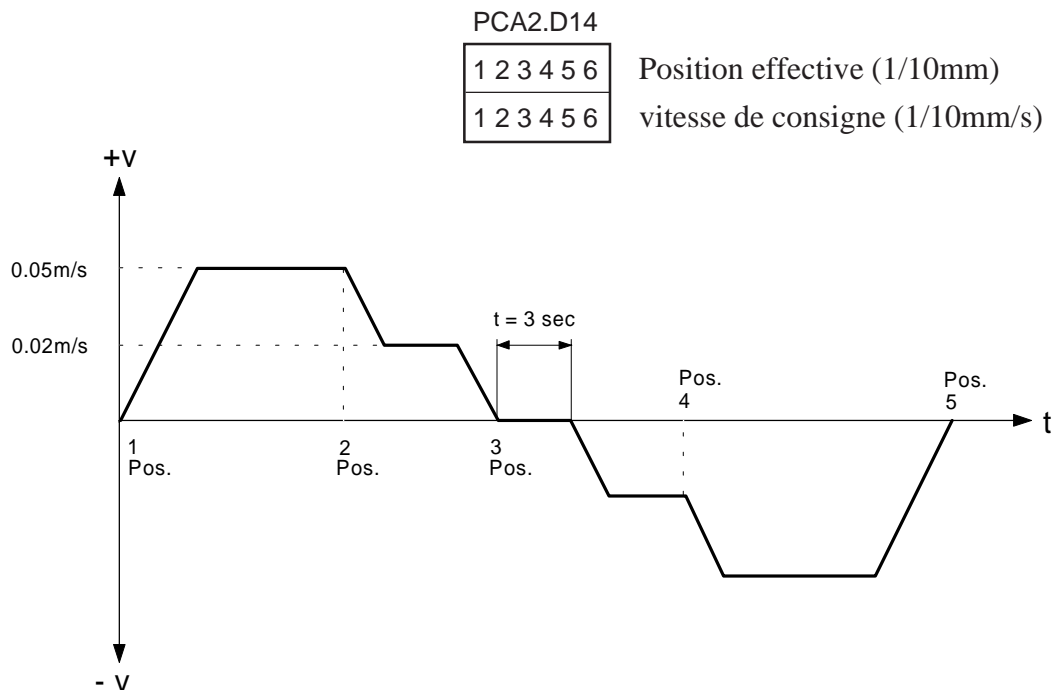
## 9.2 Exemple 2

Cet exemple repose sur l'exemple 1. L'extension traite d'une modification supplémentaire de la vitesse pendant le mouvement et de l'affichage sur le module d'affichage PCA2.D14.

### Enoncé

Matériel : modèles d'atelier comme pour l'exemple 1

Le processus de mouvement suivant doit être effectué :



Position 1 : 0 mm position de départ  
 Position 2 : 70 mm position d'interruption  
 Position 3 : 100 mm position cible  
 Position 4 : 70 mm position d'interruption  
 Position 5 : 0 mm position cible

- vitesse pour l'avance et le recul rapide : la même que pour l'avance
- l'accélération doit être constante à  $0,01 \text{ m/s}^2$
- le mouvement doit être lancé au moyen de la touche connectée à l'entrée 8
- l'affichage sur PCA2.D14 doit comporter en haut la position effective et en bas la vitesse de consigne.

## **Solution**

### **a) Installation du logiciel**

Nous partons de l'hypothèse que le matériel et le logiciel sont déjà installés, comme dans le cas de l'exemple 1.  
(les fichiers H3DEF.SRC et H3FB.SRC peuvent être repris sans modification).

### **b) Programme utilisateur**

#### **1. Initialisation dans XOB 16**

Elle peut être reprise de l'exemple 1, sans modification.

#### **2. Traitement cyclique des axes**

En extension de l'exemple 1, la position effective et la vitesse de consigne sont lues par le module H3 au moyen des fonctions "FRdAP" et "FRdSV"; elles sont ensuite affichées sur le module PCA2.D14 au moyen du bloc de fonction "DP14"

#### **3. Définition du programme de déplacement**

Définition du programme de déplacement selon le profil donnée vitesse/temps dans une structure GRAFTEC (SB 0). Si la vitesse doit être modifiée pendant le mouvement à une position donnée, ceci peut se faire en chargeant une position d'interruption et en interrogeant l'indicateur d'état "BrkPos". Il faut ce faisant veiller à remettre d'abord à zéro l'indicateur d'état "BrkPos" avec la fonction "FResSF" après le chargement de la position d'interruption.  
Voir aussi à ce sujet les fonctions "FLdBPA" et "FResSF".

Les pages suivantes contiennent le fichier source (BSP02.SRC) du programme utilisateur pour cet exemple.

```

; Demo programm for the motion control module PCD4.H3..
; =====
; Name   : BSP02.SRC
; U. Jäggi 27.08.90

$ include H3DEF.SRC

;***** Cold-Start (Initialisation)
XOB      16
;----- Cold-Start Definitions
;----- loading of the initialisation registers

Ld       R   MCW+RA1      ; Motion Control Word
          41BH           ; Stop smoothly, Position mode
                          ; only statusflag (Pos.error)
Ld       R   PosEr+RA1   ; Position Error
          2000           ; 4 * 500 pulses
Ld       R   KProp+RA1   ; Proportional factor
          150
Ld       R   KInt+RA1    ; Integral factor
          50
Ld       R   KDer+RA1    ; Derivative factor
          50
Ld       R   IntL+RA1    ; Integration Limit
          500
Ld       R   SampI+RA1   ; Sampling Interval
          15             ; 5.46ms
Ld       R   MCFac+RA1   ; Motion Control Factor
          100.0          ; 100 Imp./1/10mm
Ld       R   Veloc+RA1   ; Velocity
          500            ; 0.05m/s
Ld       R   Accel+RA1   ; Acceleration
          100            ; 0.01m/s2

;-----
CFB      AxInit          ; Axis Initialisation
          1              ; X axis
          RA1
          FA1
          IMode          ; Initialisation mode: Analog/PWM
;----- End XOB 16
EXOB

```

```

;***** Cyclic program
COB      0
        0
;-----
CFB      AxHndlg      ; Axis Handling
        1             ; X axis
        RA1
        FA1
;-----
SET      F  FRdAP+FA1  ; Read Actual Position
SET      F  FRdSV+FA1  ; Read Setpoint Velocity
;-----
SET      F  DplM+FA1   ; Display Mode = 2*6 digits
CFB      DP14          ; Display on PCA2.D14
        1             ; Axis 1
        F  DplM+FA1   ; Display Mode
        R  RActP+RA1  ; Actual Position
        R  RSetV+RA1  ; Setpoint Velocity
;-----
CSB      0             ; Call of the motion control program
;----- ; End cyclic program
ECOB

```

```

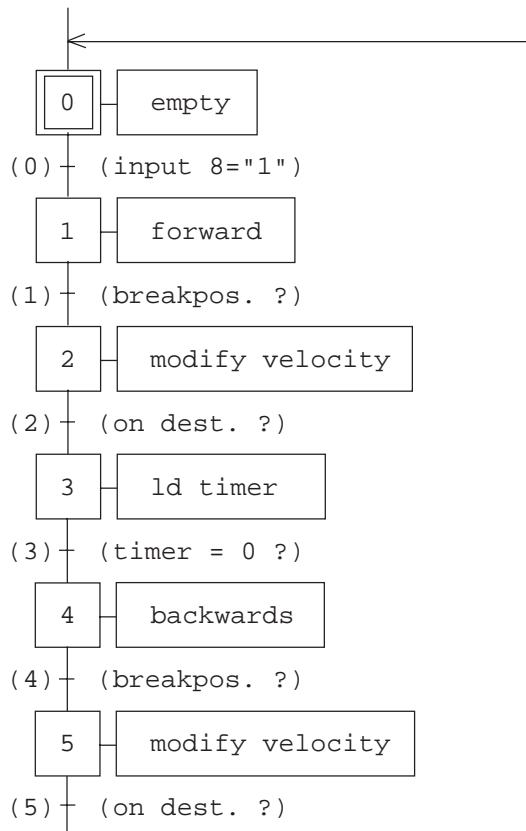
*** SAIA PCD GRAFTEC EDITOR $113 ***
FILE: BSP01.GLS (29.08.90 10.46)
*** SAIA AG - CH-3280 MURTEN ***
SB-NUMBER: 0
PAGE-NB: 0

```

```

PAGE: 1
PRODUCED: 29.08.90 10.52

```



```

;***** Motion control program
SB      0
;-----
IST      0      ; empty
          I 5      ; on dest. ?
          O 0      ; input 8="1"
EST
;-----
ST       1      ; forward
          I 0      ; input 8="1"
          O 1      ; breakpos. ?
ld      r DestP+RA1 ; Destination Position
          1000      ; 100mm = 1000*1/10mm
set     f FLdDA+FA1 ; Load Destination Absolute
ld      r BrkP+RA1 ; Break Position
          700       ; 70mm
set     f FLdBPA+FA1 ; Load Break Position Absolute
ld      r StaFRR+RA1 ; Status Flag Reset Register
          0         ; reset all Status Flag
set     f FResSF+FA1 ; Reset Status Flag
set     f FStart+FA1 ; Start motion
EST
;-----
ST       2      ; modify velocity
          I 1      ; breakpos. ?
          O 2      ; on dest. ?
ld      r Veloc+RA1 ; Velocity
          200      ; 0.02m/s
set     f FLdVA+FA1 ; Load Velocity Absolute
set     f FStart+FA1 ; Update Velocity
EST
;-----
ST       3      ; ld timer
          I 2      ; on dest. ?
          O 3      ; timer = 0 ?
ld      t 0      ; timer 0
          30      ; 3 sec.
EST
;-----
ST       4      ; backwards
          I 3      ; timer = 0 ?
          O 4      ; breakpos. ?
ld      r DestP+RA1 ; Destination Position
          0         ; Position 0
set     f FLdDA+FA1 ; Load Destination Absolute
set     f FResSF+FA1 ; Reset Status Flag
set     f FStart+FA1 ; Start motion
EST
;-----
ST       5      ; modify velocity
          I 4      ; breakpos. ?
          O 5      ; on dest. ?
ld      r Veloc+RA1 ; Velocity
          500      ; 0.05m/s
set     f FLdVA+FA1 ; Load Velocity Absolute
set     f FStart+FA1 ; Update Velocity
EST

```

```

;-----
TR      0          ;input 8="1"
        I  0          ;empty
        O  1          ;forward
sth   i  8        ;motion free ?
ETR
;-----
TR      1          ;breakpos. ?
        I  1          ;forward
        O  2          ;modify velocity
sth   f BrkPos+FA1 ;Break Position reached ?
ETR
;-----
TR      2          ;on dest. ?
        I  2          ;modify velocity
        O  3          ;ld timer
sth   f OnDest+FA1 ;On Destination ?
ETR
;-----
TR      3          ;timer = 0 ?
        I  3          ;ld timer
        O  4          ;backwards
stl   t  0        ;timer = 0 ?
ETR
;-----
TR      4          ;breakpos. ?
        I  4          ;forward
        O  5          ;modify velocity
sth   f BrkPos+FA1 ;Break Position reached ?
ETR
;-----
TR      5          ;on dest. ?
        I  5          ;modify velocity
        O  0          ;empty
sth   f OnDest+FA1 ;On Destination ?
ETR
;-----
ESB

```

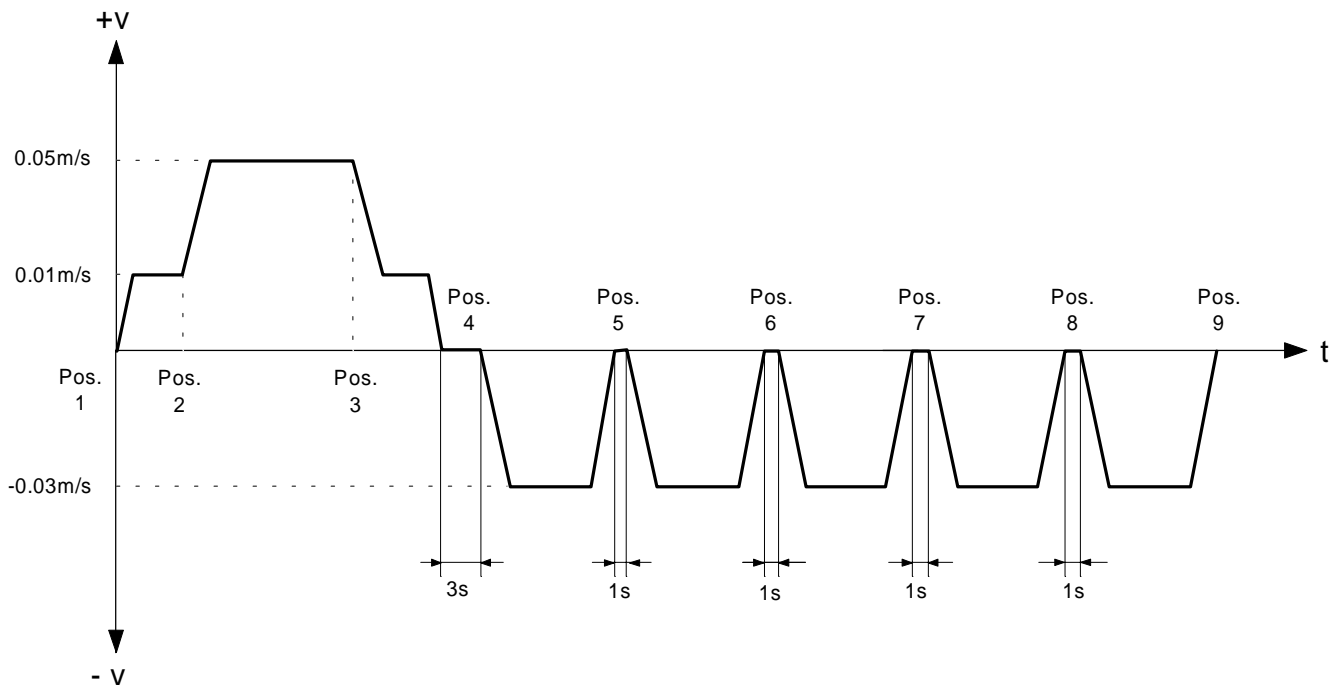
### 9.3 Exemple 3

L'exemple part de l'exemple 2. L'extension contient un programme de déplacement plus complet comprenant une partie de commande automatique et une partie manuelle.

#### Enoncé

Matériel : modèles d'atelier comme dans l'exemple 1

En mode de fonctionnement automatique, les mouvements suivants doivent être exécutés :



Position 1:	0mm	position de départ
Position 2:	30mm	position d'interruption
Position 3:	110mm	position d'interruption
Position 4:	150mm	position cible
Position 5:	120mm	position cible
Position 6:	90mm	position cible
Position 7:	60mm	position cible
Position 8:	30mm	position cible
Position 9:	0mm	position cible



**Conditions :**

- l'accélération doit être constante à  $0,01 \text{ m/s}^2$ ,
- il doit être possible de commuter à tout moment à l'aide d'un interrupteur entre le mode manuel et le mode automatique,
- après le lancement de programme automatique au moyen d'une touche, il tourne sans fin jusqu'à commutation en mode manuel ou jusqu'à interruption au moyen de la touche Stop,
- l'axe doit pouvoir être arrêté à tout moment au moyen d'une touche.

**Mode manuel :**

- il faut se déplacer à une vitesse de  $\pm 0,02 \text{ m/s}$  en sens de déplacement positif ou négatif; dès que la touche est relâchée, il doit y avoir arrêt avec le mode d'arrêt défini;
- la position effective doit être définie comme position zéro par pression sur une touche.

**Affectation des entrées**

Entrée 0	-->	commutation automatique / manuel (0/1)
Entrée 8	-->	démarrage du programme automatique
Entrée 9	-->	avance manuelle
Entrée 10	-->	recul manuel
Entrée 11	-->	définition de la position zéro
Entrée 15	-->	Stop

**Solution****a) Installation du logiciel**

Nous partons de l'hypothèse que le matériel et le logiciel sont déjà installés, comme dans l'exemple 2.

(Les fichiers H3DEF.SRC et H3FB.SRC peuvent être repris sans modification.)

**b) Programme utilisateur****1. Initialisation dans XOB 16**

Elle peut être reprise sans changement de l'exemple 2

## 2. Traitement cyclique des axes

En extension de l'exemple 2, on réalise ici en plus la commande des modes de fonctionnement. Afin de pouvoir commuter à tout moment entre le fonctionnement automatique et le fonctionnement manuel, on redémarre le programme GRAFTEC au Step 0 en pressant sur l'interrupteur relié à l'entrée 0. Ceci signifie que le mouvement momentané se poursuit jusqu'à sa conclusion jusqu'à l'arrêt de l'entraînement lors de la commutation du fonctionnement automatique en fonctionnement manuel.

En actionnant la touche reliée à l'entrée 15, l'entraînement est arrêté et le programme GRAFTEC redémarre ensuite au Step 0.

## 3. Définition du programme de déplacement

Définition du programme de déplacement conformément à l'énoncé, dans une structure GRAFTEC (SB 0).

Programme automatique :

Etant donné que pour le déplacement aux positions 5 à 9 le même trajet est parcouru à la même vitesse, ces mouvements peuvent être programmés dans une boucle (la position cible sera chargée relativement).

Les pages suivantes contiennent le fichier source (BSP03.SRC) du programme utilisateur pour cet exemple.

```

; Demo programm for the motion control module PCD4.H3..
; =====
; Name   : BSP03.SRC
; U. Jäggi 27.08.90

$ include H3DEF.SRC

;***** Cold-Start (Initialisation)
XOB          16
;----- Cold-Start Definitions
;----- loading of the initialisation registers

Ld          R  MCW+RA1    ; Motion Control Word
              41BH      ; Stop smoothly, Position mode
                       ; only statusflag (Pos.error)
Ld          R  PosEr+RA1  ; Position Error
              2000      ; 4 * 500 pulses
Ld          R  KProp+RA1  ; Proportional factor
              150
Ld          R  KInt+RA1   ; Integral factor
              50
Ld          R  KDer+RA1   ; Derivative factor
              50
Ld          R  IntL+RA1   ; Integration Limit
              500
Ld          R  SampI+RA1  ; Sampling Interval
              15        ; 5.46ms
Ld          R  MCFac+RA1  ; Motion Control Factor
              100.0     ; 100 Imp./1/10mm
Ld          R  Veloc+RA1  ; Velocity
              500       ; 0.05m/s
Ld          R  Accel+RA1  ; Acceleration
              100       ; 0.01m/s2

;-----
CFB          AxInit      ; Axis Initialisation
              1          ; X axis
              RA1
              FA1
              IMode      ; Initialisation mode: Analog/PWM

;----- End XOB 16
EXOB

```

```

;***** Cyclic program
COB          0
             0
;-----
CFB          AxHndlg      ; Axis Handling
             1            ; X axis
             RA1
             FA1
;-----
SET          F FRdAP+FA1  ; Read Actual Position
SET          F FRdSV+FA1  ; Read Setpoint Velocity
;-----
SET          F DplM+FA1   ; Diplay Mode = 2*6 digits
CFB          DP14         ; Display on PCA2.D14
             1            ; Axis 1
             F DplM+FA1   ; Display Mode
             R RActP+RA1   ; Actual Position
             R RSetV+RA1   ; Setpoint Velocity
;-----

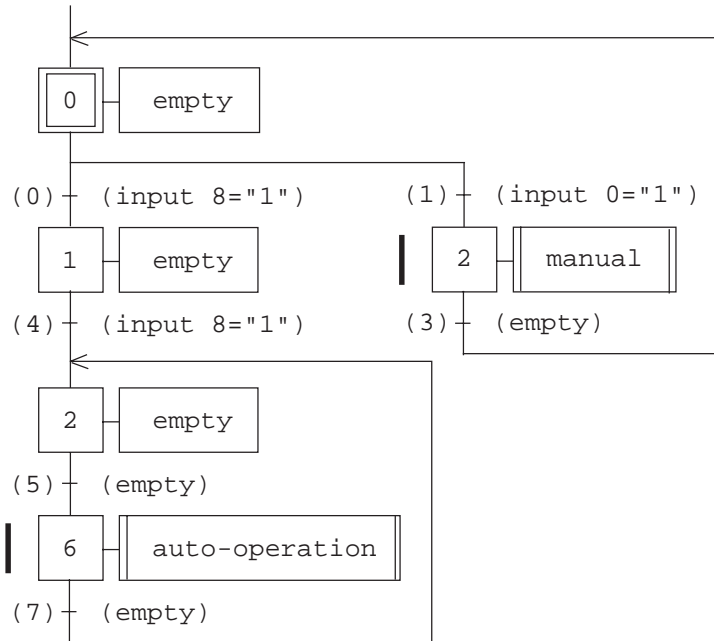
CSB          0            ; Call of the motion control program
;-----
STH          I 0          ; Manual operation mode
DYN          F 0
RSB          H 0          ; Restart SB
             0            ; at Step 0
STL          I 0          ; Automatic operation mode
DYN          F 1
RSB          H 0          ; Restart SB
             0            ; at Step 0
STH          I 15         ; Stop switch
DYN          F 15
JR           L END
SET          F FStop+FA1  ; Stop X axis
RSB          0            ; Restart SB
             0            ; at Step 0

;----- ; End cyclic program
END:        ECOB

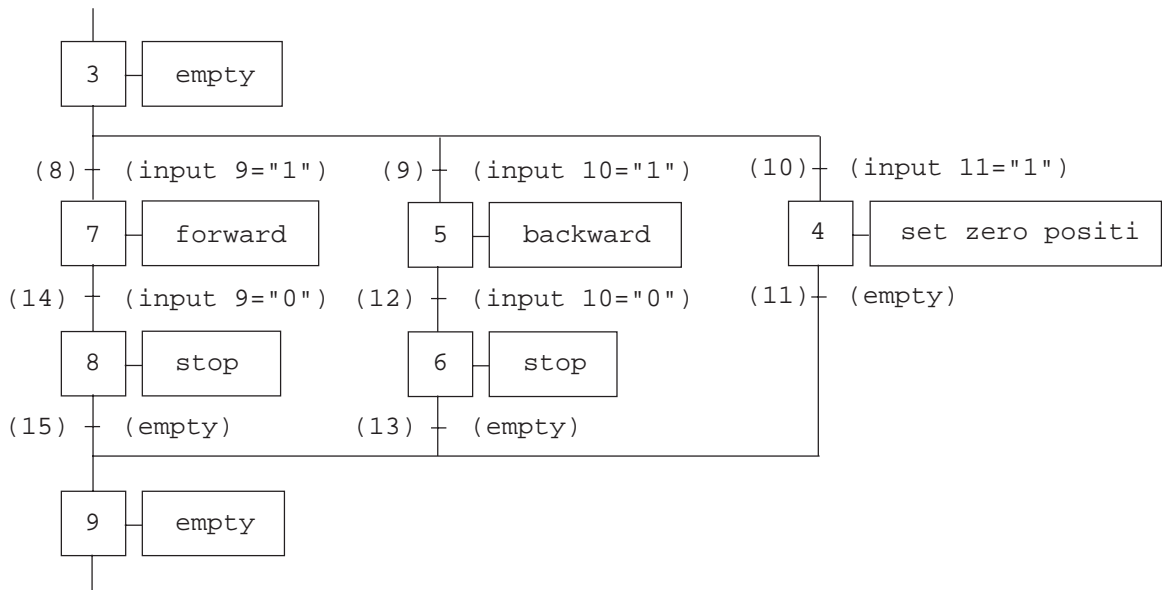
```

\*\*\* SAIA PCD GRAFTEC EDITOR \$113 \*\*\*  
 FILE: BSP03.GLS (29.08.90 11.08)  
 \*\*\* SAIA AG - CH-3280 MURTEN \*\*\*  
 SB-NUMBER: 0  
 PAGE-NB: 0

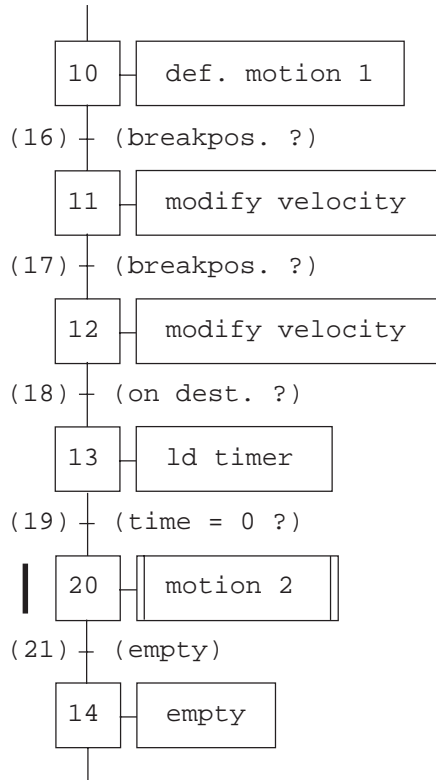
PAGE: 1  
 PRODUCED: 29.08.90 11.11



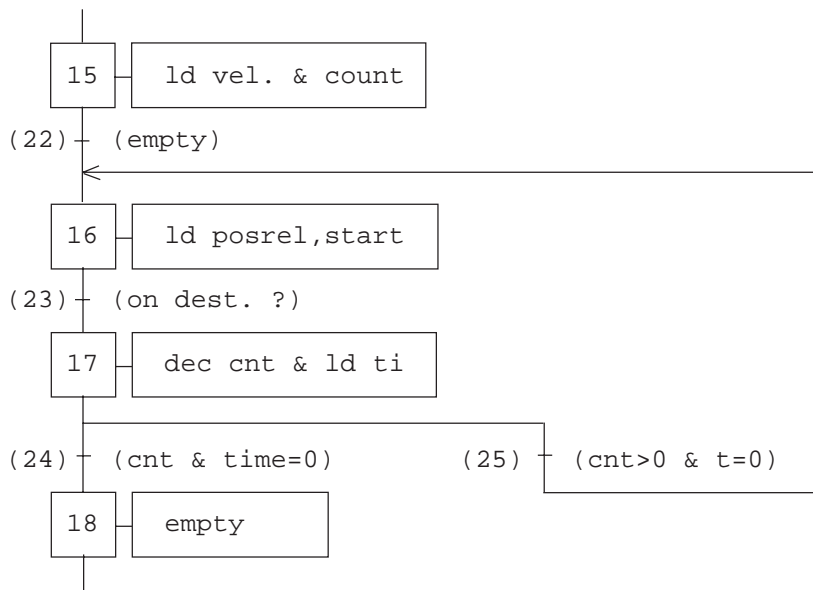
PAGE-NB: 2 manual



PAGE-NB: 6 auto-operation



PAGE-NB: 20 motion 2



```

;***** Motion control program
SB      0
;-----
IST      0      ; empty
          I  3      ; empty
          O  0      ; input 0="0"
EST
;-----
ST       1      ; empty
          I  0      ; input 0="0"
          O  4      ; input 8="1"
EST
;-----
ST       2      ; empty
          I  4      ; input 8="1"
          I  7      ; empty
          O  5      ; empty
EST
;-----
ST       3      ; empty
          I  1      ; input 0="1"
          O  8      ; input 9="1"
          O  9      ; input 10="1"
          O 10      ; input 11="1"
EST
;-----
ST       4      ; set zero position
          I 10      ; input 11="1"
          O 11      ; empty
set      f FSetZP+FA1 ;set zero position X axis
EST
;-----
ST       5      ; backward
          I  9      ; input 10="1"
          O 12      ; input 10="0"

ld       r Veloc+RA1 ;velocity
          200      ; 0.02m/s
set      f FLdVA+FA1 ;load velocity
set      f FBackW+FA1 ;backward X axis
EST
;-----
ST       6      ; stop
          I 12      ; input 10="0"
          O 13      ; empty
set      f FStop+FA1 ;stop X axis
EST
;-----
ST       7      ; forward
          I  8      ; input 9="1"
          O 14      ; input 9="0"

ld       r Veloc+RA1 ;velocity
          200      ; 0.02m/s
set      f FLdVA+FA1 ;load velocity
set      f FForw+FA1 ;forward X axis
EST

```

```

;-----
ST      8          ; stop
        I  14      ; input 9="0"
        O  15      ; empty
set    f  FStop+FA1 ; stop X axis
EST
;-----
ST      9          ; empty
        I  15      ; empty
        I  13      ; empty
        I  11      ; empty
        O  3       ; empty
EST
;-----
ST      10         ; def. motion 1
        I  5       ; empty
        O  16      ; breakpos. ?
ld     r  Veloc+RA1 ; Velocity
        100       ; 0.01m/s
set    f  FLdVA+FA1 ; Load Velocity Absolute
ld     r  DestP+RA1 ; Destination Position
        1500      ; 150mm
set    f  FLdDA+FA1 ; Load Destination Absolute
ld     r  BrkP+RA1  ; Break Position
        300       ; 30mm
set    f  FLdBPA+FA1 ; Load Break Position Absolute
ld     r  StaFRR+RA1 ; Status Flag Reset Register
        0         ; reset all Status Flag
set    f  FResSF+FA1 ; Reset Status Flag
set    f  FStart+FA1 ; Start motion
EST
;-----
ST      11         ; modify velocity
        I  16      ; breakpos. ?
        O  17      ; breakpos. ?
ld     r  Veloc+RA1 ; Velocity
        500       ; 0.05m/s
set    f  FLdVA+FA1 ; Load Velocity Absolute
ld     r  BrkP+RA1  ; Break Position
        1100      ; 110mm
set    f  FLdBPA+FA1 ; Load Break Position Absolute
set    f  FResSF+FA1 ; Reset Status Flag
set    f  FStart+FA1 ; Start motion
EST
;-----
ST      12         ; modify velocity
        I  17      ; breakpos. ?
        O  18      ; on dest. ?
ld     r  Veloc+RA1 ; Velocity
        100       ; 0.01m/s
set    f  FLdVA+FA1 ; Load Velocity Absolute
set    f  FStart+FA1 ; Start motion
EST

```



```

;-----
ST      13          ;ld time
        I 18          : on dest. ?
        O 19          ;time = 0 ?
ld    t 0
        20
EST
;-----
ST      14          ; empty
        I 21          ; empty
        O 7           ; empty
EST
;-----
ST      15          ;ld vel. & counter
        I 19          : time = 0 ?
        O 22          ; empty
ld    r Veloc+RA1 ; Velocity
        300          ; 0.03m/s
set   f FLdVA+FA1 ; Load Velocity Absolute
ld    c 100
        5
EST
;-----
ST      16          ;ld posrel,start
        I 22          : empty
        I 25          : cnt>0 & t=0
        O 23          ; on dest. ?
ld    r DestP+RA1 ; Destination Position
        -300         ; -30mm
set   f FLdDR+FA1 ; Load Destination Relative
set   f FStart+FA1 ; Start motion
EST
;-----
ST      17          ;dec cnt & ld timer
        I 23          : on dest. ?
        O 24          ; cnt & time=0
        O 25          ; cnt>0 & t=0
dec   c 100
ld    t 0
        10
EST
;-----
ST      18          ; empty
        I 24          ; cnt & time=0
        O 21          ; empty
EST
;-----

```

```

;-----
TR      0          ; input 0="0"
        I  0          ; empty
        O  1          ; empty
sth    i  0        ; auto op. ?
ETR
;-----
TR      1          ; input 0="1"
        I  0          ; empty
        O  3          ; empty
sth    i  0        ; manual op. ?
ETR
;-----
TR      2          ; manual
        I  3          ; empty
        O  9          ; empty
ETR
;-----
TR      3          ; empty
        I  9          ; empty
        O  0          ; empty
ETR
;-----
TR      4          ; input 8="1"
        I  1          ; empty
        O  2          ; empty
sth    i  8        ; start auto op. ?
ETR
;-----
TR      5          ; empty
        I  2          ; empty
        O  10         ; def. motion 1
ETR
;-----
TR      6          ; auto-operation
        I  10         ; def. motion 1
        O  14         ; empty
ETR
;-----
TR      7          ; empty
        I  14         ; empty
        O  2          ; empty
ETR
;-----
TR      8          ; input 9="1"
        I  3          ; empty
        O  7          ; forward
sth    i  9        ; manual forward ?
ETR
;-----
TR      9          ; input 10="1"
        I  3          ; empty
        O  5          ; backward
sth    i  10       ; manual backwards ?
ETR

```

```

;-----
TR      10          ; input 11="1"
        I  3          ; empty
        O  4          ; set zero position
sth    i  11       ; define zero pos. ?
ETR
;-----
TR      11          ; empty
        I  4          ; set zero position
        O  9          ; empty
ETR
;-----
TR      12          ; input 10="0"
        I  5          ; backward
        O  6          ; stop
stl    i  10       ; end zero pos. ?
ETR
;-----
TR      13          ; empty
        I  6          ; stop
        O  9          ; empty
ETR
;-----
TR      14          ; input 9="0"
        I  7          ; forward
        O  8          ; stop
stl    i  9        ; end manual forward ?
ETR
;-----
TR      15          ; empty
        I  8          ; stop
        O  9          ; empty
ETR
;-----
TR      16          ; breakpos. ?
        I  10         ; def. motion 1
        O  11         ; modify velocity
sth    f BrkPos+FA1 ; Break Position reached ?
ETR
;-----
TR      17          ; breakpos. ?
        I  11         ; modify velocity
        O  12         ; modify velocity
sth    f BrkPos+FA1 ; Break Position reached ?
ETR
;-----
TR      18          ; on dest. ?
        I  12         ; modify velocity
        O  13         ; ld time
sth    f OnDest+FA1 ; on destination ?
ETR
;-----
TR      19          ; time = 0 ?
        I  13         ; ld time
        O  15         ; ld vel. & counter
stl    t  0
ETR

```

```

;-----
TR      20          ;motion 2
        I 15          ;ld vel. & counter
        O 18          ;empty
ETR
;-----
TR      21          ;empty
        I 18          ;empty
        O 14          ;empty
ETR
;-----
TR      22          ;empty
        I 15          ;ld vel. & counter
        O 16          ;ld posrel,start
ETR
;-----
TR      23          ;on dest ?
        I 16          ;ld posrel,start
        O 17          ;dec cnt & ld timer
sth    f OnDest+FA1 ;on destination ?
ETR
;-----
TR      24          ;cnt & time=0
        I 17          ;dec cnt & ld timer
        O 18          ;empty
stl    c 100
anl    t 0
ETR
;-----
TR      25          ;cnt>0 & t=0
        I 17          ;dec cnt & ld timer
        O 16          ;ld posrel,start
sth    c 100
anl    t 0
ETR
;-----

ESB

```

## 10. Vue d'ensemble des commandes et des symboles

### Vue d'ensemble des fonctions exécutables avec le FB "AxHndlg" avec indications des valeurs d'entrée et de sortie.

#### Définitions du mode de fonctionnement

FB indicateur d'exécution	Désignation / Fonction	Durée de traitement	Exécutable pendant un mouvement	Page	Valeurs d'entrée / Sortie			
					Symbole	Type	Format	Désignation
FSeIOM	Select Operation Mode	1,9ms	oui	7-24	MCW	R	binaire	Motion Control Word
FSetPE	Set Position Error	1,9ms	oui	7-25	PosEr MCW	R R	entier binaire	Position Error Motion Control Word

#### Commandes de déplacement

FB indicateur d'exécution	Désignation / Fonction	Durée de traitement	Exécutable pendant un mouvement	Page	Valeurs d'entrée / Sortie			
					Symbole	Type	Format	Désignation
FStart	Start Motion	1ms	oui	7-43	--			
FStop	Stop Motion	2,7ms	oui	7-44	MCW	R	binaire	Motion Control Word
FMotOff	Motor Off	1,8ms	oui	7-46	--			
FSStepF	Single Step Forward	4,5ms	non	7-47	--			
FSStepB	Single Step Backwards	4,5ms	non	7-48	--			
FForw	Forward with def. Velocity	4,5ms	oui	7-49	--			
FBackw	Backwards with def. Velocity	4,5ms	oui	7-50	--			

## Entrées de paramètres pour le profil de vitesse

FB indicateur d'exécution	Désignation / Fonction	Durée de traitement	Exécutable pendant un mouvement	Page	Valeurs d'entrée / Sortie			
					Symbole	Type	Format	Désignation
FLdDR	Load Destination Position Relative	4,1ms	oui	7-30	DestP MCFac	R R	entier virg. flot.	Destination Position Motion Control Factor
FLdDA	Load Destination Position Absolote	4,1ms	oui	7-28	DestP MCFac	R R	entier virg flot.	Destination Position Motion Control Factor
FLdVR	Load Velocity Relative	4,5ms	oui	7-34	Veloc MCFac	R R	entier virg. flot.	Velocity Motion Control Factor
FLdVA	Load Velocity Absolote	4,5ms	oui	7-32	Veloc MCFac	R R	entier virg. flot.	Velocity Motion Control Factor
FLdAR	Load Acceleration Relative	7,2ms	conditionné	7-38	Accel MCFac	R R	entier virg. flot.	Acceleration Motion Control Factor
FLdAA	Load Acceleration Absolote	7,2ms	conditionné	7-36	Accel MCFac	R R	entier virg. flot.	Acceleration Motion Control Factor
FLdRP	Load Regulator Parameter	5,2ms	oui	7-40	KProp KInt KDer IntL Sampl	R R R R R	entier entier entier entier entier	Proportional Factor Integral Factor Derivative Factor Integration Limit Sampling Interval Derivative Term
FUpDRP	Up Date Regulator Parameters	0,9ms	oui	7-42	--			

## Commandes de lectures pour les données

FB indicateur d'exécution	Désignation / Fonction	Durée de traitement	Exécutable pendant un mouvement	Page	Valeurs d'entrée / Sortie			
					Symbole	Type	Format	Désignation
FRdAP	Read Actual Position	4,3ms	oui	7-51	RActP	R	entier	Actual Position
					MCFac	R	virg. flot.	Motion Control Factor
FRdSP	Read Setpoint Position	4,3ms	oui	7-52	RSetP	R	entier	Setpoint Position
					MCFac	R	virg. flot.	Motion Control Factor
FRdAV	Read Actual Velocity	2,8ms	oui	7-53	RActV	R	entier	Actual Velocity
					MCFac	R	virg. flot.	Motion Control Factor
FRdSV	Read Setpoint Velocity	3,7ms	oui	7-54	RSetV	R	entier	Setpoint Velocity
					MCFac	R	virg. flot.	Motion Control Factor
FRdITS	Read Integration Term Sum	1,9ms	oui	7-55	RIntTS	R	entier	Integration Term Sum
FRdIP	Read Index Position	4,3ms	oui	7-56	RIndP	R	entier	Index Position
					MCFac	R	virg. flot.	Motion Control Factor
FRdSR	Read Signal Register	1,8ms	oui	7-57	RSigB	R	binaire	Signalisation Bits

## Fonctions auxiliaires

FB indicateur d'exécution	Désignation / Fonction	Durée de traitement	Exécutable pendant un mouvement	Page	Valeurs d'entrée / Sortie			
					Symbole	Type	Format	Désignation
FResSF	Reset Status Flag	1,8ms	oui	7-60	StaFRR	R	binaire	Status Flag Reset Register
FLdBPR	Load Break Position Relative	3,3ms	oui	7-64	BrkP	R	entier	Break Position
					MCFac	R	virg. flot.	Motion Control Factor
FLdBPA	Load Break Position Absolute	3,3ms	oui	7-62	BrkP	R	entier	Break Position
					MCFac	R	virg. flot.	Motion Control Factor
FSetIP	Set Index Position	0,9ms	oui	7-66	--			
FSetZP	Set Zero Position	0,9ms	oui	7-67	--			

**Notes personnelles :**



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

Manuel PCD4.H3xx

Modules de positionnement pour  
servo-entraînements

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

**Vos commentaires :**