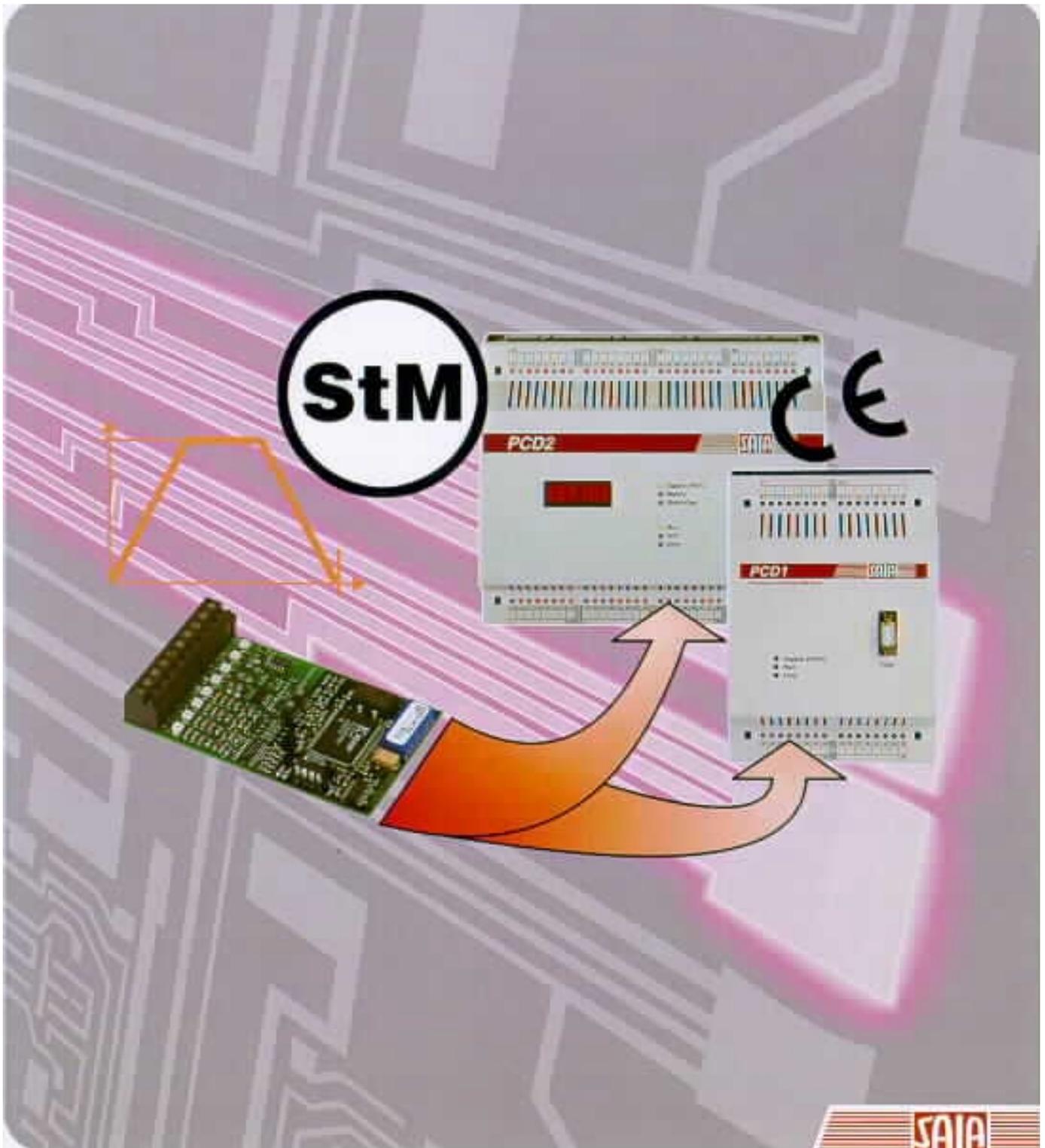


SAIA® PCD
Process Control Devices

PCD2.H210
Module de positionnement
pour moteurs pas-à-pas



SAIA® Process Control Devices

Module de positionnement pour moteurs pas-à-pas

PCD2.H210

SAIA-Burgess Electronics SA 1999. Tous droits réservés
Edition 26/760 F2 - 05.99

Sous réserve de modifications

Mise à jour

Manuel : Module de positionnement pour moteurs pas-à-pas PCD2.H210 - édition F2

Date	Chapitre	Page	Description
15.05.2000	7.1.2	7-5 et 7-6	Description position zéro (avec registre 'rPosAbs_x')
15.05.2000	Annexe A	A-13	Offset pour position de référence

Table des matières

	Page
1. Présentation	
1.1 Généralités	1-1
1.2 Fonctionnement et application	1-2
1.3 Points forts	1-3
1.4 Domaines d'utilisation	1-4
1.5 Programmation	1-5
2. Caractéristiques techniques	
2.1 Caractéristiques matérielles	2-1
2.2 Caractéristiques électriques	2-3
2.3 Caractéristiques de fonctionnement	2-4
3. Constitution	
4. Repérage des bornes, câbles et voyants de signalisation d'état	
5. Description fonctionnelle	
5.1 Schéma synoptique du module	5-1
5.2 Description du module	5-2
5.3 Complément d'information sur le profil de vitesse	5-6
5.4 Complément d'information sur la fonction Retour à l'origine (bloc de fonctions « HOME »)	5-7
6. Prise de contact	
6.1 Bien démarrer avec la programmation en liste d'instructions (IL)	6-2
6.2 Bien démarrer avec la programmation en FUPLA	6-5

	Page
7. Programmation	
7.1 Programmer en liste d'instructions IL avec les blocs de fonctions « FB »	7-2
7.1.1 Constitution de la fourniture	7-2
7.1.2 Description des blocs de fonctions	7-5
7.1.3 Définition des paramètres	7-7
7.2 Programmer en FUPLA avec les boîtes de fonctions « FBoxes »	7-9
7.3 Programmer en GRAFTEC avec les boîtes de fonctions	7-10
8. Typologie des erreurs et diagnostic	
8.1 Erreurs de définition	8-1
8.2 Erreurs de traitement	8-2
8.2.1 Paramètres incorrects	8-2
8.2.2 Erreur de Retour à l'origine	8-3
9. Exemples de programmation	
9.1 Exemple utilisateur en GRAFTEC et en liste d'instructions (IL)	9-1
9.2 Exemple utilisateur en FUPLA	9-11
9.3 Exemple utilisateur en GRAFTEC et en FUPLA	9-12

		Page
Annexe A	Récapitulatif des blocs de fonctions et des commandes PCD2.H210 en langage IL	
INIT	Initialisation du module PCD2.H210	A-1
EXEC	Exécution d'une commande PCD2.H210	A-2
LdAcc	Chargement de l'accélération	A-3
LdVmin	Chargement de la fréquence de Start/Stop (Vmin)	A-4
LdVmax	Chargement de la fréquence maximale (Vmax)	A-5
LdDestRel	Chargement du déplacement en mode relatif	A-6
LdDestAbs	Chargement du déplacement en mode absolu	A-7
RdPosition	Lecture de la position réelle (actuelle)	A-8
Start	Démarrage du déplacement	A-9
Stop	Arrêt du déplacement	A-9
Continue	Poursuite d'un déplacement interrompu	A-10
RdIdent	Lecture de l'identification du module	A-11
SetOut2/3	Mise à 1 des sorties A2 et A3	A-12
ResOut2/3	Mise à 0 des sorties A2 et A3	A-12
HOME	Retour à l'origine (recherche point de référence)	A-13
Annexe B	Récapitulatif des boîtes de fonctions et des commandes PCD2.H210 en langage FUPLA	

Notes personnelles :



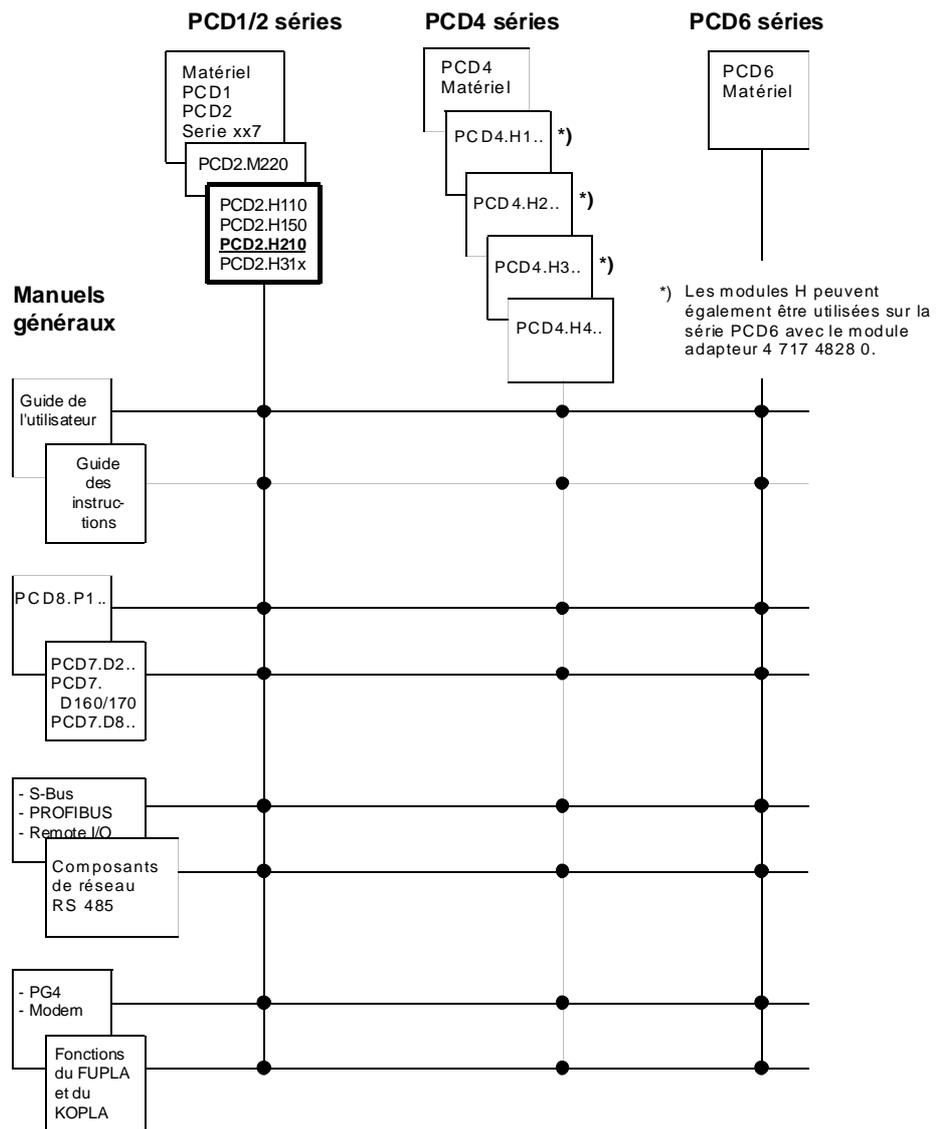
Avis aux lecteurs :

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).

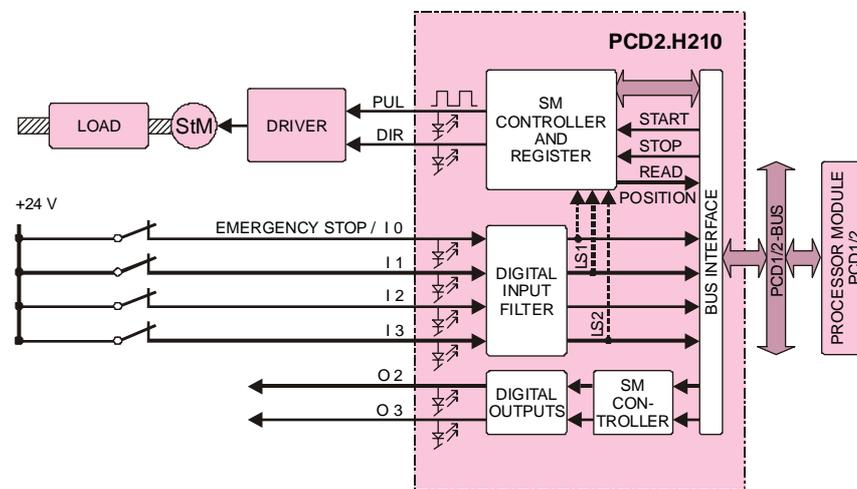
Exploitée dans les règles de l'art, la gamme SAIA® PCD intègre des constituants d'automatismes modernes, alliant sécurité et haute fiabilité, et capables d'assurer pendant des années les fonctions de contrôle-commande, de régulation et de surveillance de votre équipement.

1. Présentation

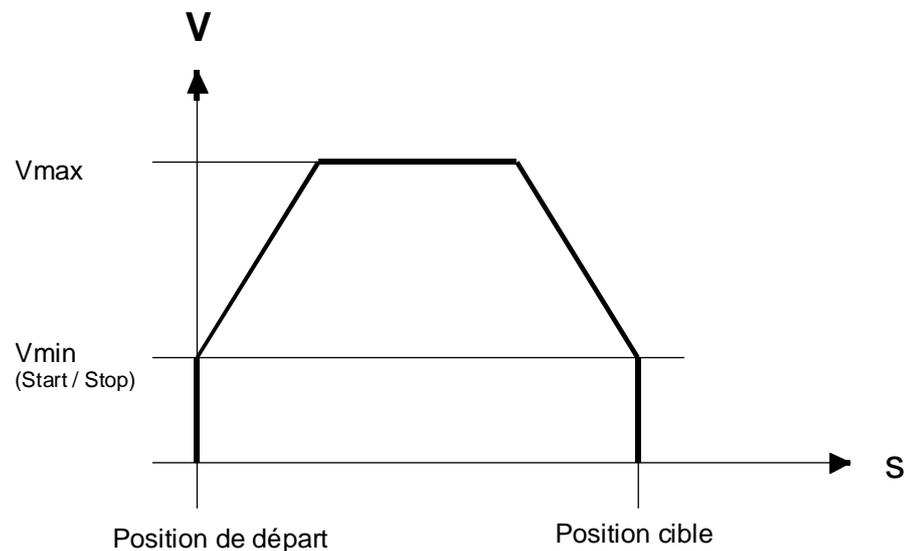
1.1 Généralités

Le PCD2.H210 assure la commande et la surveillance parfaitement autonomes du déplacement d'un moteur pas-à-pas avec rampes au démarrage et au freinage. Chaque PCD2.H210 commande un axe indépendant et fournit un train d'impulsions monophasées à l'étage de puissance du moteur pas-à-pas.

Schéma synoptique d'un moteur pas-à-pas



Profil de vitesse type



1.2 Fonctionnement et application

Ce module économique peut s'enficher dans n'importe quel connecteur d'E/S d'un PCD1 ou d'un PCD2 pour piloter l'étage de puissance d'un moteur pas-à-pas, à une fréquence atteignant 19.454 kHz.

Il est ainsi possible de commander jusqu'à 16 axes sur le PCD2.M1../..M2.. et 4 sur le PCD1. Chaque PCD2.H210 disposant en outre de 4 entrées TOR et de 2 sorties TOR configurables au choix, 16 axes donnent un total de 64 entrées TOR et de 32 sorties TOR supplémentaires qui peuvent servir à d'autres tâches de commande du procédé.

Le PCD2.H210 autorise la commande et la surveillance totalement autonomes des cycles de déplacement d'un moteur pas-à-pas. Les commandes nécessaires aux mouvements du moteur sont transmises au module par des blocs de fonctions du programme utilisateur. Au cours du déplacement, le processeur du moteur contrôle le profil de vitesse, ainsi que les rampes d'accélération et de décélération afin d'atteindre la position cible sans perte de pas. Si chaque module H210 pilote un axe indépendant, il est néanmoins possible de démarrer plusieurs axes en coordination quasi synchrone.

1.3 Points forts

- Commande d'axe économique fonctionnant en boucle ouverte avec une très grande précision et une fiabilité à toute épreuve.
- Fréquence (9.5 à 19 454 Hz), accélération et nombre d'impulsions pilotés avec la précision du quartz.
- Position actuelle de l'axe ou nombre d'impulsions de sorties accessibles à tout moment.
- Entrée E2 faisant office d'entrée TOR 24 VCC normale, à défaut d'être utilisée par le bloc de fonctions « HOME ».
- Possibilité de 3 entrées supplémentaires, configurables à l'initialisation en entrées TOR 24 VCC normales, en fins de course (repérées E1 et E3, et provoquant un arrêt avec rampe de freinage) et/ou en arrêt d'urgence (E0).
- 2 sorties TOR (0.5 A, 24 VCC) disponibles sur le même module pour assurer d'autres tâches du procédé.
- Programmation simple grâce à des blocs de fonctions « FB » ou boîtes de fonctions « FBoxes » bien structurés.

1.4 Domaines d'utilisation

- Automatisation de machines (manutention, assemblage...),
- Chargement-déchargement,
- Palettisation,
- Déplacement angulaire automatique (de caméras, phares automobiles, antennes...),
- Commande de moteurs exigeant un couple stationnaire élevé,
- Pilotage d'axes statiques (définition).

1.5 Programmation

Les blocs de fonctions "FB" (liste d'instruction IL) ou boîtes de fonctions "FBox" (langage FUPLA) du logiciel de programmation graphique PG4 (sous Windows) permettent de ne saisir que les paramètres indispensables à la commande d'axe. Ils font l'objet d'un manuel descriptif complet, illustré d'exemples pratiques.

Commandes d'initialisation

INIT Choix du numéro de module
 Choix de la plage de fréquence
 Activation de la fonction Arrêt d'urgence
 Activation des fins de course
 Fréquence de démarrage-arrêt (Vmin)
 Fréquence maxi (Vmax)
 Accélération

Commandes d'exécution

EXEC Chargement de la position cible relative
 Chargement de la position cible absolue
 Démarrage du déplacement (génération d'impulsions de démarrage)
 Arrêt du déplacement (interruption)
 Poursuite du déplacement après interruption
 Lecture de la position (état du compteur)
 Lecture de l'identité du module
 Mise à 1 / Mise à 0 des sorties TOR

Commande de mise en service

HOME Recherche du point de référence (Retour à l'origine)

Typologie des erreurs et diagnostic

Détection des paramètres incorrects (Registre Diagnostic) et des erreurs de programmation.

Surveillance de la fonction Retour à l'origine (Home) par Timeout.

Notes personnelles :

2. Caractéristiques techniques

2.1 Caractéristiques matérielles

Entrées TOR

Nombre d'entrées	4
Tension nominale	24 V
Niveau bas :	-30 à +5 V
Niveau haut :	+15 à +30 V
Fonctionnement exclusivement en logique positive	Pour des raisons de sécurité, utiliser des contacts normalement fermés (NF - normally closed)
Courant d'entrée (typique)	6.5 mA
Type de circuit	Sans séparation galvanique
Filtrage d'entrée	< 1 ms

Sorties TOR

Nombre de sorties	4
Plage de courant	0.5 A pour chaque sortie dans la plage de tension de 10 à 32 VCC, ondulation résiduelle 10 % maxi
Type de circuit	Sans séparation galvanique, ni protection contre les courts-circuits Commutation du « + »
Chute de tension	0.3 V maxi à 0.5 A
Temps de réponse (typique)	50 µs, 100 µs maxi en cas de charge inductive

Alimentation

Consommation de courant à partir du bus PCD1/2	5 VCC, 20 à 45 mA
Externe (utilisateur) (alimentation des sorties)	24 VCC (10 à 32 VCC), 2 A maxi ondulation résiduelle lissée 10 % maxi

Conditions de fonctionnement

Température ambiante de	fonctionnement : 0 à +50°C (sans ventilation)
	stockage : -20 à + 85°C
Immunité aux parasites	" CE mark " selon EN 50 081-1 et EN 50 082-2

Références de commande

PCD2.H210	Module de positionnement pour 1 axe de moteur pas-à-pas
PCD9.H21E	Bibliothèque logicielle (software) avec les blocs de fonctions (FB)

Voyants de signalisation d'état

Nombre de voyants lumineux	8
Voyant 0 :	*) Tension sur entrée E0 (arrêt d'urgence)
Voyant 1 :	*) Tension sur entrée E1 (fin de course LS1)
Voyant 2 :	*) Tension sur entrée E2 (interrupteur de référence REF)
Voyant 3 :	*) Tension sur entrée E3 (fin de course LS2)
Voyant 4 :	Tension sur sortie A0 (PUL)
Voyant 5 :	Tension sur sortie A1 (DIR)
Voyant 6 :	Tension sur sortie A2
Voyant 7 :	Tension sur sortie A3

*) Etat inversé lorsqu'il s'agit d'une entrée fin de course.

Programmation

Par le programme utilisateur PCD
(PG4), assistée d'une bibliothèque
de blocs et boîtes de fonctions.

2.2 Caractéristiques électriques

Consommation de courant interne

+5 V	20 à 45 mA
Uext.	0 à 10 mA (sans courant de charge)

Alimentation externe

Bornes +/-	10 à 32 VCC lissée, ondulation résiduelle 10 % maxi, diode TVS 39 V \pm 10 % 2 A maxi par sorties, pas protégées contre une mauvaise polarité
------------	---

Entrées TOR	4 entrées TOR, repérées E0 à E3 (Cf. § 2.1)
--------------------	--

Sorties TOR	4 sorties TOR, repérées A0 à A3 (Cf. § 2.1)
--------------------	--

2.3 Caractéristiques de fonctionnement

Nombre de systèmes	1
Distance de positionnement (plage de déplacement)	0 à 16 777 215 (24 bits)
Plages de fréquence (au choix)	9.5 à 2 431 Hz 19 à 4 864 Hz 38 à 9 727 Hz 76 à 19 454 Hz Division de plage, selon la plage de fréquences choisie (Cf. § 7.1.3)
Accélération	0.6 à 153 kHz/s 1.2 à 306 kHz/s 2.4 à 612 kHz/s 4.8 à 1 224 kHz/s Division de plage non linéaire, selon la plage de fréquences choisie (Cf. § 7.1.3)
Générateur de profils	Avec rampes d'accélération et de freinage symétriques
Protection des données	Toutes les données de ce module sont volatiles (possibilité d'utiliser des registres non volatiles du PCD)

3. Constitution

Module équipé

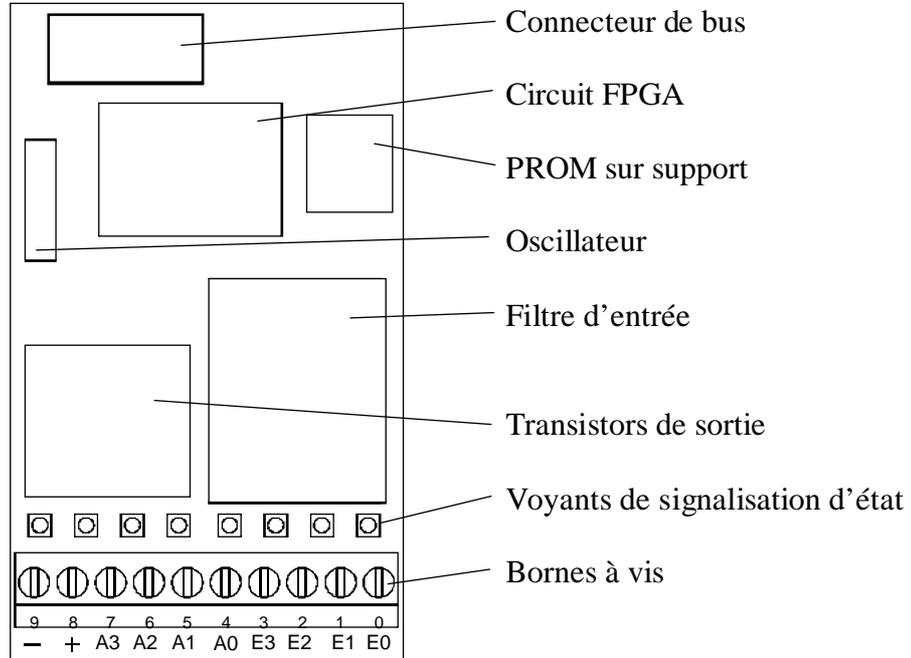
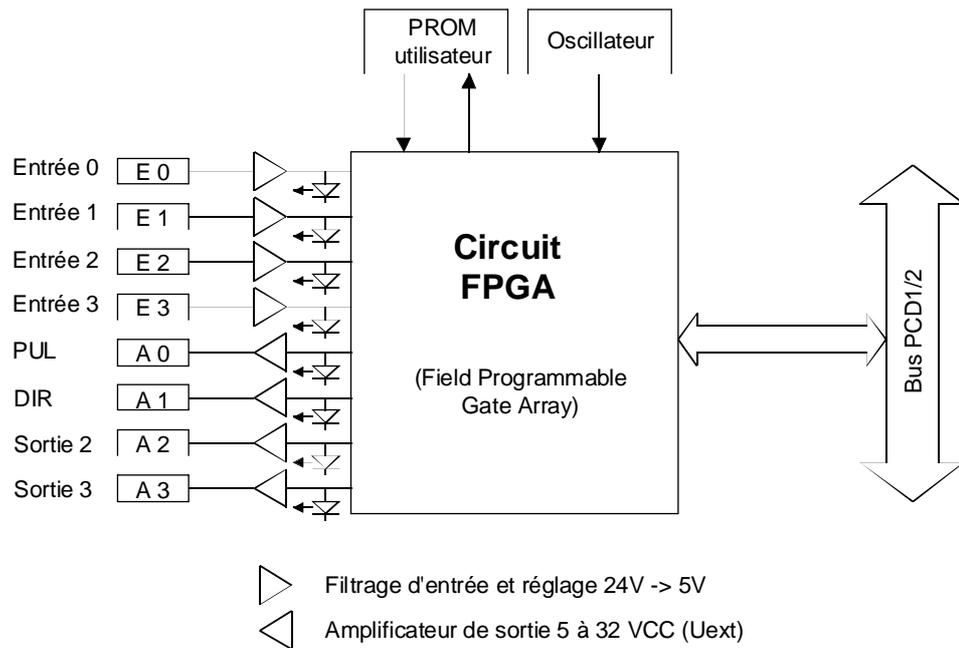


Schéma synoptique

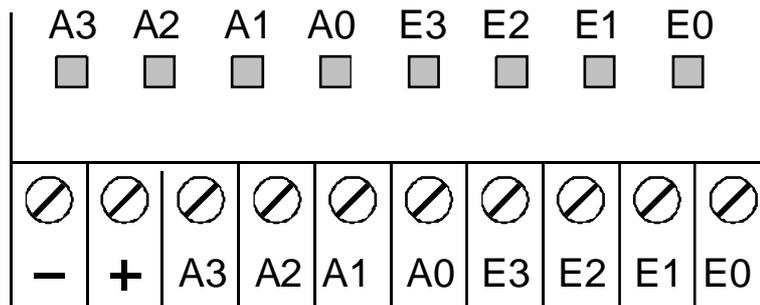


Notes personnelles :

4. Repérage des bornes, câbles et voyants de signalisation d'état

Bornes à vis

La figure ci-dessous montre le marquage sur le circuit imprimé.
Le connecteur E/S standard est numéroté de 0 à 9 (de droite à gauche).



Entrées 4

Borne 0 =	E0 :	Configurable en arrêt d'urgence ou en entrée normale
Borne 1 =	E1 :	Configurable en fin de course (LS1) ou en entrée normale
Borne 2 =	E2 :	Configurable en interrupteur de référence (REF) ou en entrée normale
Borne 3 =	E3 :	Configurable en fin de course (LS2) ou en entrée normale

Sorties 4

Borne 4 =	A0 :	Impulsions moteur (PUL)
Borne 5 =	A1 :	Sens de rotation du moteur (DIR)
Borne 6 =	A2 :	Programmable par l'utilisateur
Borne 7 =	A3 :	Programmable par l'utilisateur

Alimentation

Borne 8 =	+	+ 24 VCC
Borne 9 =	-	Masse (GND)

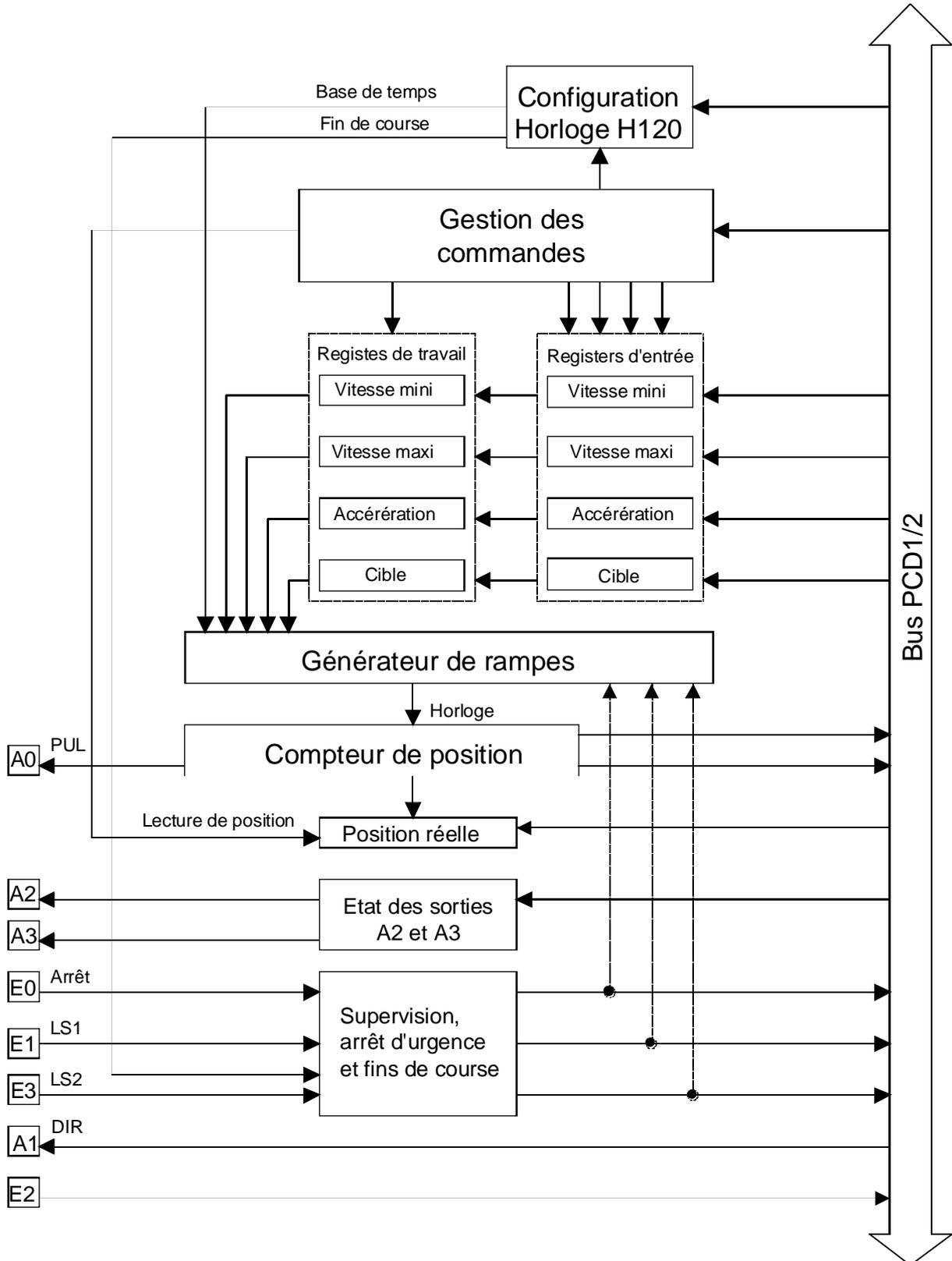
Voyants de signalisation d'état

Voyant 0 =	*) Tension sur entrée E0	(arrêt d'urgence)
Voyant 1 =	*) Tension sur entrée E1	(fin de course LS1)
Voyant 2 =	*) Tension sur entrée E2	(interrupteur de référence REF)
Voyant 3 =	*) Tension sur entrée E3	(fin de course LS2)
Voyant 4 =	Tension sur sortie A0	(PUL)
Voyant 5 =	Tension sur sortie A1	(DIR)
Voyant 6 =	Tension sur sortie A2	
Voyant 7 =	Tension sur sortie A3	

*) Etat inversé lorsqu'il s'agit d'une entrée fin course.

5. Description fonctionnelle

5.1 Schéma synoptique du module



5.2 Description du module

Le bon fonctionnement d'un moteur pas-à-pas oblige l'utilisateur à définir 4 paramètres :

- La fréquence « Start/Stop », c'est-à-dire la fréquence à laquelle un moteur peut démarrer et s'arrêter directement, sans perte de pas,
- La fréquence maximale à laquelle le moteur peut accélérer pour fonctionner en toutes circonstances,
- L'accélération optimale permettant de passer de la fréquence mini à la fréquence maxi, et inversement,
- Le nombre de pas à effectuer dans un déplacement.

Les trois premiers paramètres sont dictés par le moteur ou le système mis en œuvre; ils sont donc normalement définis une fois pour toutes et ne sont en principe plus modifiés (les paramètres 2 et 3 doivent éventuellement être adaptés selon la charge). Chargés à l'initialisation du module par le bloc ou la boîte de fonctions « INIT », ils peuvent aussi être modifiés individuellement.

Le quatrième paramètre dépend de la tâche à exécuter et doit être en permanence revu par le programme utilisateur. Il est normalement chargé par ce dernier, juste avant la sortie des impulsions. Pour des applications très rapides, il est possible de précharger dans le registre d'entrée, au cours d'un déplacement, le nombre de pas du déplacement suivant (EXEC_LdDest...); ce paramètre est alors transféré dans le registre de travail dès la fin du déplacement en cours, ce qui permet de démarrer immédiatement le déplacement suivant (EXEC_Start). (Notons que cette démarche est tout aussi envisageable pour les trois autres paramètres, bien que ceux-ci restent normalement inchangés.)

C'est la commande de démarrage « Start » du bloc de fonctions « EXEC » qui lance la sortie d'impulsions (sur A0 « PUL ») et le déplacement.

La base de temps est définie lors de la configuration initiale (FB_INIT), parmi quatre plages numérotées 0 à 3 :

Plage 0	→	9.5	à	2 431	Hz,
Plage 1	→	19	à	4 864	Hz,
Plage 2	→	38	à	9 727	Hz,
Plage 3	→	76	à	19 454	Hz.

Sur atteinte de la position cible, l'indicateur « OnDest_x » est positionné : son interrogation permet de contrôler le traitement séquentiel du programme utilisateur.

Il est également possible d'interrompre un train d'impulsions (commande d'arrêt « Stop » du bloc EXEC), puis de poursuivre le mouvement interrompu (commande « Continue » du bloc EXEC), pour le mener à terme sans perte de pas, conformément à la spécification. Précisons néanmoins que cette possibilité doit être prévue dans le programme utilisateur.

La position actuelle, à savoir l'état du compteur interne, peut être lue à tout moment et affichée par la commande « RdPosition » du bloc EXEC. La position absolue est rafraîchie dans le registre « rPosAbs_x » à chaque commande « RdPosition ».

Il importe de distinguer deux modes de déplacement :

- relatif et
- absolu (indicateur « Abs_x »).

En mode relatif, c'est toujours le parcours à effectuer pour atteindre la cible (soit le nombre de pas) qui est chargé. Ainsi, pour exécuter 1 000 pas en avant, il faut charger la valeur 1000. A l'inverse, pour revenir au point de départ, il faut programmer -1000. En cas de valeurs négatives, la sortie A1 (DIR) passe automatiquement à l'état bas, ce qui permet d'inverser le sens de rotation du moteur.

En mode absolu, c'est la position absolue à atteindre, c'est-à-dire la cible, qui est chargée. Le bloc de fonctions calcule alors le déplacement relatif du contrôleur de moteur pas-à-pas. La position absolue (registre « PosAb_x ») doit être chargée avec la position effective avant le premier déplacement. Pour effectuer, par exemple, un aller-retour de 1 000 pas à partir d'une position 0, il faut saisir la position 1000 correspondant à la cible, démarrer le moteur, puis charger la position 0 correspondant au point de départ. Si la nouvelle position absolue est plus petite que l'actuelle, le sens de rotation du moteur est automatiquement changé (la sortie DIR passe à l'état bas).

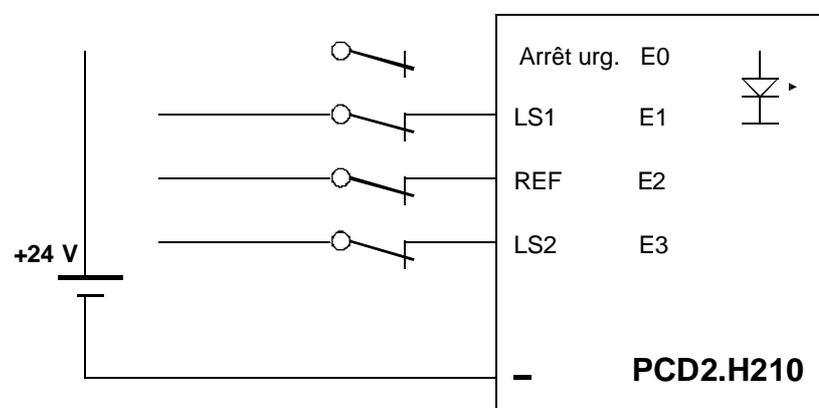
L'entrée d'arrêt d'urgence « E0 » permet d'arrêter brutalement (c'est-à-dire sans rampe de freinage) un déplacement. Il convient ensuite de reparamétrer et de réinitialiser le déplacement.

Il existe également trois autres entrées, repérées « E1 », « E2 » et « E3 ». E1 et E3 sont configurables, par le bloc INIT, en entrées TOR ou en fins de course (« LS1 » et « LS2 »). E2 sert d'interrupteur de référence (« REF ») au bloc de fonctions « HOME » (Cf. § 5.4). A défaut, c'est une entrée TOR normale.

Les fins de course LS1 et LS2, la référence REF et l'arrêt d'urgence sont des contacts normalement fermés et alimentent les entrées en +24 V. Les circuits d'entrée ont été conçus pour des arrêts brusques, avec une grande précision (constante de temps du filtre d'entrée < 1 ms) : En conséquence, il faut utiliser une tension continue lissée (Cf. Caractéristiques techniques). Si ces entrées ne sont pas configurées en fins de course ou en arrêt d'urgence, leur fonctionnement s'apparente à celui d'entrées *normales*.

Des voyants de signalisation d'état lumineux indiquent toujours la présence de tension sur l'entrée correspondante.

Raccordement des entrées configurées en arrêt d'urgence, en fins de course (LS) et en interrupteur de référence (REF) :



L'activation de l'entrée LS1 ou LS2 (tension = 0 → voyant éteint) déclenche directement une rampe de freinage si les interrupteurs ont été configurés en fins de course.

Le signal d'entrée « LSxTrig_x » indique que le déplacement a été stoppé par un fin de course (LS). Il passe et reste également à l'état haut si le fin de course devient inactif ; il repasse à l'état bas sur démarrage d'un nouveau déplacement (si aucun fin de course n'est actif).

L'activation de l'arrêt d'urgence provoque l'arrêt immédiat (sans rampe de freinage) du déplacement, ce qu'indique le signal d'entrée « EmergTrig_x ». Une lecture de position (commande « RdPosition ») intervenant après un arrêt d'urgence permet de connaître la position à laquelle l'arrêt d'urgence a été activé, ce qui ne correspond pas forcément à la position à laquelle le moteur s'est arrêté.

L'entrée REF n'a pas d'incidence directe sur le déplacement et ne sert qu'au bloc de fonctions HOME pour approcher le point de référence.

L'interrogation des entrées porte toujours sur des noms symboliques, quelle que soit la configuration, soit :

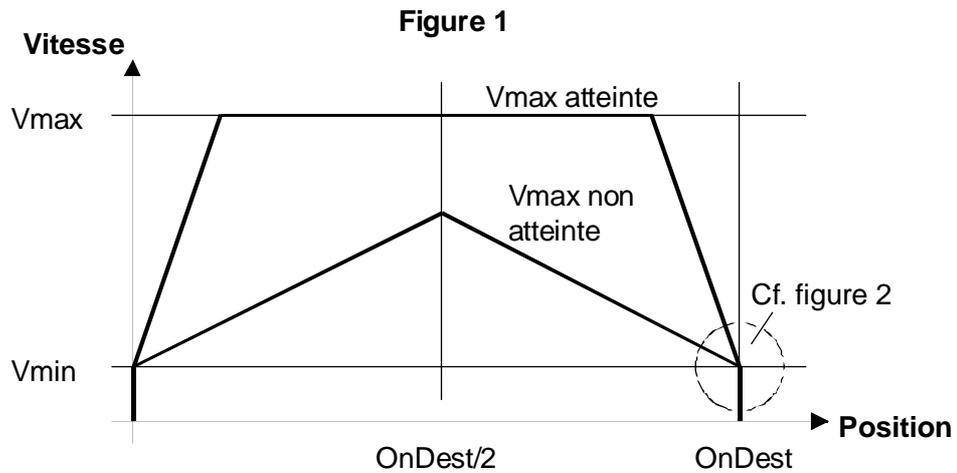
Emerg_x	→	Entrée E0,
LS1_x	→	Entrée E1,
REF_x	→	Entrée E2,
LS2_x	→	Entrée E3,

« x » donnant le numéro de module.

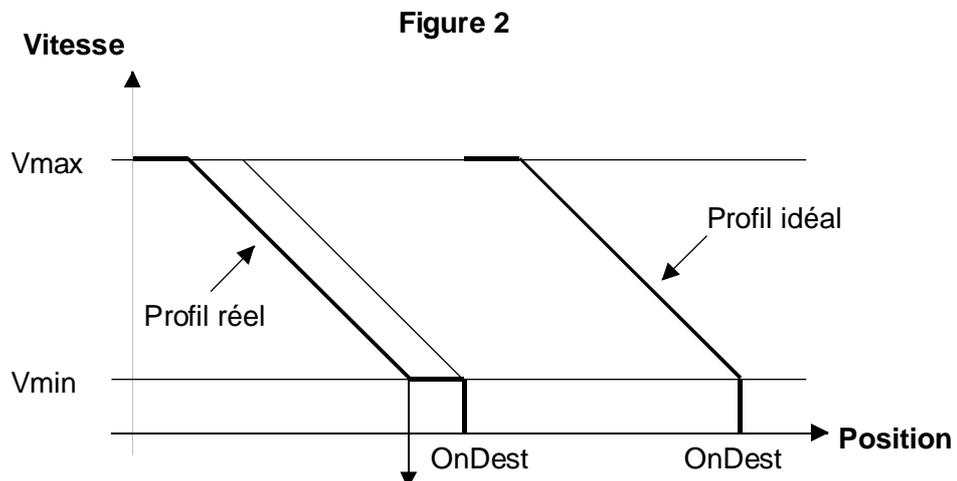
Les sorties A2 et A3 n'ont pas d'usage particulier et peuvent être activées par les commandes « SetOut2 – SetOut3 » et « ResOut2 – ResOut3 » du bloc de fonctions EXEC (Cf. Annexe A).

A noter qu'il est impossible de relire l'état logique de ces sorties.

5.3 Complément d'information sur le profil de vitesse



Nombre de pas en accélération = Nombre de pas en freinage



Déplacement achevé en effectuant quelques pas à la vitesse V_{min}

L'équation ci-dessous permet de calculer le temps maximal d'exécution d'un déplacement :

$$\text{Temps maxi} = \frac{V_{max}}{V_{min}} \times T_{acc}$$

avec :

V_{max} = vitesse maximale atteinte au cours du déplacement

V_{min} = vitesse minimale programmée (fréquence Start/Stop)

T_{acc} = temps d'accélération : $n * 250 \mu s$ ($1 \leq n \leq 255$)

Nota : Cette formule correspond au cas de figure le plus défavorable. Il suffit de modifier légèrement les paramètres V_{max} , V_{min} et T_{acc} pour optimiser le déplacement et son temps d'exécution.

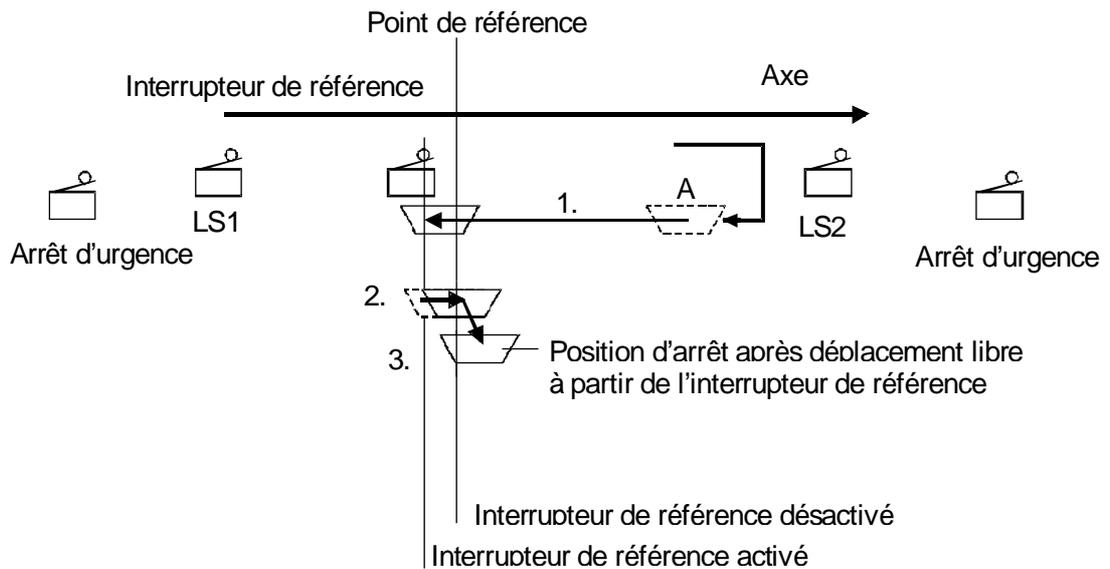
5.4 Complément d'information sur la fonction Retour à l'origine (bloc de fonctions « HOME »)

La fonction Retour à l'origine peut être exécutée de façon indépendante à l'aide du bloc HOME (Cf. § 7.1.2).

Neuf paramètres servent à définir le parcours à effectuer jusqu'au point de référence. Précisons qu'il faut d'abord initialiser l'axe à référencer avec le bloc INIT.

- La recherche de l'interrupteur de référence s'effectue aux fréquences minimale et maximale définies respectivement par les paramètres 6 et 7 et à l'accélération donnée par le paramètre 8, dans le sens indiqué par le paramètre 3. Si, à défaut d'un interrupteur de référence, l'axe rencontre un fin de course (matériel ou logiciel), le sens de recherche est inversé.
- Si l'interrupteur de référence est trouvé, le déplacement libre commence, dans le sens défini par le paramètre 4 et à la fréquence donnée par le paramètre 6 (Vmin).
- Après désactivation de l'interrupteur de référence, l'axe s'arrête et la position du point de référence (paramètre 2) est chargée sous forme de position absolue.
- Le module est configuré avec ses paramètres initiaux (bloc INIT) et le traitement du bloc HOME est interrompu.

Cette démarche est illustrée sur la figure de la page suivante.



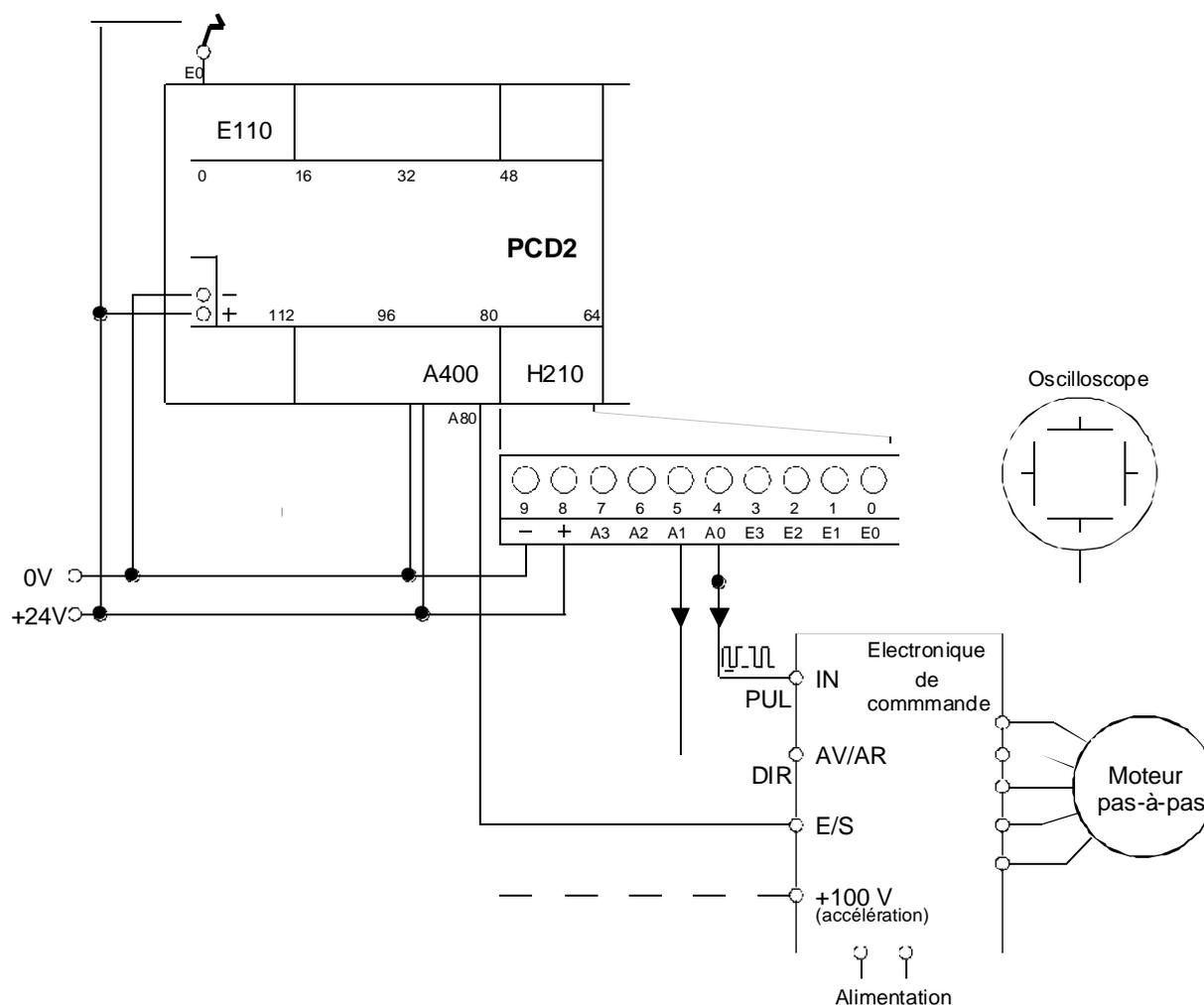
Attention :

- Sur un même PCD, il est possible de référencer plusieurs modules à la fois.
- Utiliser des contacts repos pour les interrupteurs.
- Durant l'exécution de la fonction Retour à l'origine (HOME), les fins de course sont désactivés par l'électronique interne; ils servent uniquement d'inverseurs de sens.
- L'arrêt d'urgence n'est actif que s'il est configuré par le bloc INIT.
- Si l'axe ne rencontre pas d'interrupteur de référence, l'indicateur d'erreur « fHomeErr_x » (avec x = numéro de module) est positionné, la position absolue est forcée à 0 et le bloc HOME est interrompu (Cf. § 8.2.2).

Remarque : si l'axe ne rencontre pas d'interrupteur de référence, il se produit au maximum 2 changements de direction lorsque l'axe atteint les interrupteurs de fin de course.

- La fonction Retour à l'origine (homing procedure) ne peut commencer que si l'indicateur « fEndHome_x » est mis à 0 par le programme utilisateur. Cet indicateur sera positionné automatiquement (haut) à la fin de la procédure.
- Le bloc HOME n'est normalement terminé qu'après une exécution correcte de la fonction ou la détection d'une erreur. Toutefois, il est également possible de l'annuler par une temporisation (paramètre 9) correspondant au délai d'expiration (en secondes) du bloc HOME. Ce cas de figure entraîne le positionnement de l'indicateur d'erreur « fHomeErr_x » ainsi que le chargement du registre de diagnostic « rDiag » avec le code 9 (troisième octet). Le contrôle du paramétrage s'effectue comme exposé au chapitre 8.

6. Prise de contact



Configuration minimale requise pour piloter un moteur pas-à-pas à l'aide du PCD2.H210 :

- PCD1 ou PCD2 équipé d'au moins
 - 1 module PCD2.H210
 - 1 module PCD2.E110
 - 1 module PCD2.A400
- Electronique de commande adaptée
- Moteur pas-à-pas, éventuellement doté d'un mobile
- Alimentation
- Oscilloscope (facultatif)

Les paramètres de la commande d'axe 'Vmin', 'Vmax' and 'Acc' (accélération) doivent être adaptés aux moteurs utilisés et ne peuvent par conséquent **pas** être pris tels qu'ils sont dans les exemples ci-après.

6.1 Bien démarrer avec la programmation en liste d'instructions (IL)

Pour mettre en service un moteur pas-à-pas avec un maximum de facilité, nous vous proposons le petit programme suivant.



Un programme utilisateur bien écrit doit normalement être exempt de boucle d'attente. Ce n'est toutefois pas la démarche retenue pour cet exemple qui vise à mieux illustrer les principales commandes d'un PCD2.H210. Dans la pratique, il convient de toujours choisir une structure GRAFTEC ou dans le futur FUPLA pour des programmes de ce type.

Caractéristiques du moteur :

Vmin (fréquence démarrage-arrêt) : 40 Hz
 Vmax : 1 000 Hz
 Accélération : 2 000 Hz/s
 Pas/tour : 48

Exemple de tâche : Après démarrage du PCD, l'activation de l'entrée E0 doit permettre au moteur pas-à-pas d'effectuer un déplacement de 4 800 pas.

Chaque paramètre de la commande d'axe ainsi que l'adresse de base du module doivent être configurés conformément aux indications du paragraphe 7.1.

Le programme utilisateur se présente alors comme suit :

(Vous trouverez un exemple similaire ("exemple1.src") mais avec des mouvements d'aller et retour sur la disquette « Blocs de fonctions FB ».)

Les blocs de fonctions « FB » (IL pour PG4) se trouvent sur la disquette PCD9.H21E. Suivez les indications du fichier « README.TXT » pour installer correctement ces blocs de fonctions sur votre PC.

Le nombre de modules (1) ainsi que l'adresse du module PCD2.H210 (adresse 64) doivent être enregistrés dans le fichier 2D2H210_B.MBA :

```
NbrModules EQU 1 ; nb. de modules H210 utilisés (0 à 16)
BA_1 EQU 64 ; adresse de base du module 1
```

Ce fichier (2D2H210_B.MBA) doit se trouver dans le répertoire du projet de cet exemple, c.-à-d. que ce fichier doit être copié manuellement de la disquette vers l'actuel répertoire du projet.

```

$include d2h210_b.equ
$group H210

        xob          16

        cfb          init          ; initialisation
          k          1             ; numéro du module
          0          0             ; plage de fréquence 0
          0          0             ; arrêt d'urgence : non
          0          0             ; fins de course : non
          4          4             ; Vmin
          105        105          ; Vmax
          77         77           ; accélération

        ld   r       1             ; nombre de pas
          4800

        exob
;-----

        cob          0
          0

st1:    sth   i     0             ; démarrage ?
        jr   l     st1

        cfb          exec          ; chargem. déplacement relatif
          k          1             ; numéro du module
          LdDestRel  ; commande : charg. dépl. rel.
          r          1             ; nombre de pas

        cfb          exec          ; démarrage du déplacement
          k          1             ; numéro du module
          start      ; commande : démarrage dépl.
          rNotused   ; registre vide

wait:   sth   l     OnDest_1     ; terminé ?
        jr   l     wait

st2:    sth   i     0             ; démarrage ?
        jr   h     st2

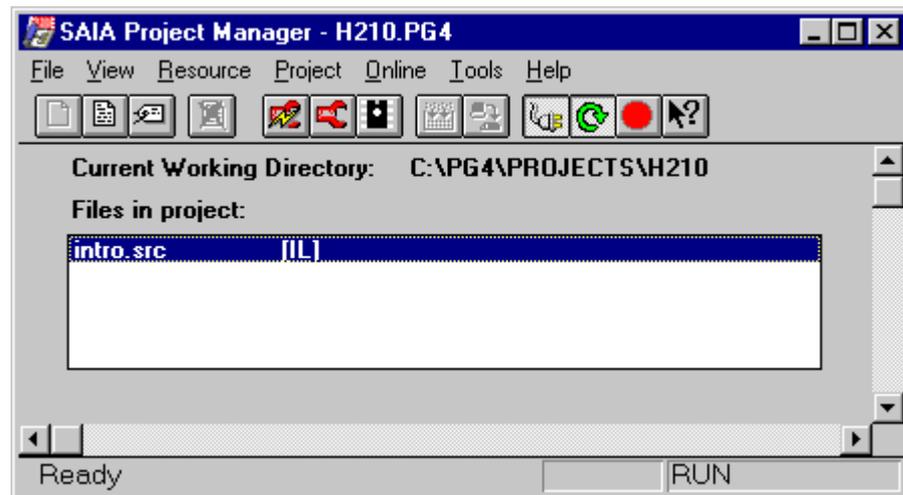
        ecob

$endgroup
;-----

```

Comme déjà mentionné, un programme utilisateur bien écrit doit normalement être exempt de boucle d'attente. Ce n'est toutefois pas la démarche retenue pour cet exemple qui vise à mieux illustrer les principales commandes d'un PCD2.H210. Dans la pratique, il convient de toujours choisir une structure GRAFTEC pour des programmes de ce type (CF chapitre 9).

Ce programme, baptisé « intro.src », doit figurer dans le projet « H210 ».



La commande d'exécution « Project – Build » traite le programme et le télécharge dans le PCD. La mise en service s'effectue à l'aide du programme de mise au point.

S'il importe de visualiser en permanence la position (par exemple, sur un afficheur PCD2.F5x0), il est plus simple d'ajouter un nouveau bloc d'organisation cyclique COB au COB 0 existant et de l'éditer comme suit :

```

cob          0
              0
. . .
ecob

;-----
$group H210

cob          1
              0

cob          k      exec      ; lecture de la position
              1      RdPosition ; module n° 1
              r      99      ; commande: RdPosition
              r      99      ; registre avec résultat

dsp          r      99      ; affichage du registre

ecob

$endgroup
;-----

```

Dans ce cas, il faut insérer une instruction de transition au bloc d'organisation cyclique suivant (« NCOB ») à chaque boucle d'attente du COB 0 ; sinon, le programme risque de rester bloqué sans jamais pouvoir atteindre COB 1. En revanche, force est de constater que cette démarche donne lieu à une programmation maladroite et alambiquée.

6.2 Bien démarrer avec la programmation en FUPLA

(En préparation)

Notes personnelles :

7. Programmation

Les outils de programmation du logiciel PG4 permettent de créer le programme utilisateur chargé de gérer les modules de comptage et de positionnement PCD2.H210.

Il est recommandé de ne plus utiliser le logiciel précédent, PG3, dans la mesure où celui-ci ne comprend pas les nouvelles commandes en langage assembleur. Cependant, des blocs de fonctions spécifiques pour le logiciel PG3 sont également disponibles si nécessaire.

Il existe deux modes de programmation :

- en liste d'instructions (IL), à l'aide de blocs de fonctions,
- en FUPLA, à l'aide de boîtes de fonctions (en préparation).

Les bibliothèques de fonctions correspondantes sont disponibles sur disquette référencée PCD9.H21E.

La commande d'axe faisant toujours appel à des tâches séquentielles, il est préférable d'écrire les programmes utilisateur en GRAFTEC, chaque étape et transition pouvant être à leur tour éditées en langage IL (blocs de fonctions) ou FUPLA (boîtes de fonctions).

Le BLOCTEC ou le FUPLA sont également deux autres possibilités de programmation.

7.1 Programmer en liste d'instructions IL avec les blocs de fonctions « FB »

7.1.1 Constitution de la fourniture (paquet IL)

Référence de commande de la disquette : PCD9.H21E.

Cette disquette contient les répertoires suivants :

- APPSDIR toutes les aides « helps »
- FB les fichiers .SRC et .EQU du H210
- FBOX les boîtes de fonctions (FBoxes) du H210
- PG3_FB tous les fichiers pour les blocs de fonctions PG3
- PG4_FB les exemples et le fichier .MBA
- Readme les informations générales

La fourniture est utilisable à partir de la version V2.0.70 du programme PG4 de SAIA. Consultez le fichier « Readme » pour toutes les autres versions du PG4. (La fourniture comporte également les blocs de fonctions utilisables avec l'ancien programme PG3)

Les boîtes de fonctions pour FUPLA ne sont pas encore disponibles.

Procédure d'installation sous PG4

Pour simplifier votre tâche, utilisez les fichiers d'installation « Setup Extra Files » du PG4.

- 1) Introduisez la disquette PCD9.H21E dans le lecteur A.
- 2) Cliquez sur le bouton « Start », pointez sur « Programs » et choisissez « SAIA PG4 », puis « Setup Extra Files ».

Les blocs de fonctions « FB » et le fichier d'aide « Help » s'installent dans le répertoire « PG4 » du disque dur.

L'installation porte sur les fichiers :

- D2H210_B.SRC Code source des FB (accessible en lecture seule)
- D2H210_B.EQU Définition des FB (accessible en lecture seule)

Ces 2 fichiers, situés sous A:\FB, sont copiés dans le répertoire ... \PG4\FB du PG4.

- FB_LIB.HLP Bibliothèque des blocs de fonctions
- D2H210_B.HLP Blocs de fonctions d'aide (help)

Ces 2 fichiers, situés sous A:\APPSDIR, sont copiés dans le répertoire ... \PG4 du PG4.

- D2H210_B.MBA Définition des adresses de base du module

Ce fichier, situé sous A:\PG4_FB, doit être copié **manuellement** dans le répertoire du projet en cours.

Le fichier **D2H210_b.mba** est représenté ci-dessous.

```

;
; Fichier modifiable par l'utilisateur
;
; Adresses de base définies par l'utilisateur
; -----
$group H210
NbrModules      EQU      1          ; Nb de PCD2.H210 utilisés (0 à 16)
;
; Adresses de base des modules(ne définir que les modules utilisés)

BA_1             EQU      32         ; Adresse de base du module 1
BA_2             EQU      0          ; Adresse de base du module 2
BA_3             EQU      0          ; Adresse de base du module 3
BA_4             EQU      0          ; Adresse de base du module 4
BA_5             EQU      0          ; Adresse de base du module 5
BA_6             EQU      0          ; Adresse de base du module 6
BA_7             EQU      0          ; Adresse de base du module 7
BA_8             EQU      0          ; Adresse de base du module 8
BA_9             EQU      0          ; Adresse de base du module 9
BA_10            EQU      0          ; Adresse de base du module 10
BA_11            EQU      0          ; Adresse de base du module 11
BA_12            EQU      0          ; Adresse de base du module 12
BA_13            EQU      0          ; Adresse de base du module 13
BA_14            EQU      0          ; Adresse de base du module 14
BA_15            EQU      0          ; Adresse de base du module 15
BA_16            EQU      0          ; Adresse de base du module 16
$endgroup

```

Il faut indiquer le nombre de modules PCD2.H210 mis en œuvre ainsi que leur adresse de base respective.

Ce fichier d'extension « .mba » (pour *module base address*) ne figurant pas dans la liste des fichiers du Gestionnaire de projet, il faut passer par un éditeur de texte (par ex., SEDIT32) pour le modifier.

Les modules doivent être numérotés dans l'ordre croissant en commençant avec « BA_1 ». Lorsque, par exemple, 3 modules sont utilisés dans un projet, ils seront numérotés « BA_1, BA_2 et BA_3 ». L'emplacement des modules dans le PCD2 est laissé au libre choix de l'utilisateur.

Exemple :

```

NbrModules      EQU      3          ; Nb de PCD2.H210 utilisés (0 à 16)
;
; Adresses de base des modules(ne définir que les modules utilisés)

BA_1             EQU      64         ; Adresse de base du module 1
BA_2             EQU      208        ; Adresse de base du module 2
BA_3             EQU      112        ; Adresse de base du module 3
BA_4             EQU      0          ; Adresse de base du module 4
BA_5             EQU      0          ; Adresse de base du module 5

```

Les adresses de base des registres, indicateurs et blocs de fonctions sont automatiquement affectées et consultables dans la liste des ressources, par la commande « View », puis « Resource List ».

Le projet à créer doit s'intituler « TEST-H2 » et le module de programme utilisateur, « move-01.sfc ».

Les fichiers sont organisés de la manière suivante :

```
C:\PG4 \FB          \D2H210_b.equ
          \D2H210_b.src
          \...
          \FBOX      \...
          \GALEP3    \...
          \PROJECTS  \FUP_E          (exemple démo PG4)
          \          \GRAF_E        (exemple démo PG4)
          \          \TEST-H2       \D2H210_b.mba
          \          \move-01.sfc
          \...
          \D2H210_b.hlp
```

Le programme utilisateur de la partie H210 est structuré comme suit :

```
$include D2H210_b.equ
$group H210

XOB      16

PCD-Code

ecob
$endgroup
```

Si le programme est édité en GRAFTEC, les directives assembleur \$include et \$group sont placées dans la première étape (ST), qui constitue normalement l'étape initiale (IST). La directive \$endgroup met un terme à la dernière transition (TR).

Une fois l'installation correctement effectuée, le programme utilisateur édité et tous les paramètres définis, le programme peut être traité et téléchargé dans le PCD par la commande « Project – Build » du Gestionnaire de projet.



7.1.2 Description des blocs de fonctions

La bibliothèque est constituée des trois principaux blocs de fonctions « FB » suivants :

- INIT Initialisation Bloc de 7 paramètres
- EXEC Exécution Bloc de 3 paramètres
- HOME Retour à l'origine Bloc de 9 paramètres

L'appel du bloc de fonctions INIT revêt toujours la forme suivante :

```
CFB      init
      k    1      ; Param. 1 : n° de module      (k 1 à k 16)
          1      ; Param. 2 : plage de fréquence (0 à 3)
          0      ; Param. 3 : arrêt d'urgence   (0=HS/1=ES)
          0      ; Param. 4 : fin de course    (0=HS/1=ES)
          1      ; Param. 5 : fréq. mini (Vmin) (1 à 255)
          100    ; Param. 6 : fréq. maxi (Vmax) (1 à 255)
          20     ; Param. 7 : accélération     (1 à 255)
```

En règle générale, l'appel du bloc de fonctions EXEC revêt la forme suivante :

```
CFB      exec
      k    1      ; Param. 1 : n° de module      (k 1 à k 16)
          LdDestRel ; Param. 2 : commande
      r    777    ; Param. 3 : valeur (du registre source)
```

```
CFB      exec
      k    1      ; Param. 1 : n° de module      (k 1 à k 16)
          start    ; Param. 2 : commande
          rNotUsed ; Param. 3 : inutilisé
```

```
CFB      exec
      k    1      ; Param. 1 : n° de module      (k 1 à k 16)
          RdPosition ; Param. 2 : commande
      r    1000   ; Param. 3 : valeur (dans registre cible)
```

Il faut toujours définir trois paramètres, même dans le cas où il en suffit de deux; le troisième est alors déclaré inutilisé (« rNotUsed ») ou consiste en n'importe quel registre.

Toutes les commandes du bloc de fonctions EXEC sont énumérées en page suivante et détaillées en Annexe A.

Commandes du bloc de fonctions EXEC :

```
LdVmin      ; chargement de la fréquence de Start/Stop (Vmin)
LdVmax      ; chargement de la fréquence maxi (Vmax)
LdAcc       ; chargement de l'accélération

LdDestRel   ; chargement du déplacement en mode relatif

LdDestAbs   ; chargement du déplacement en mode absolu

Start       ; initialisation et démarrage du déplacement
Stop        ; arrêt du déplacement
Continue    ; poursuite du déplacement après interruption
```

```

SetOut2          ; mise à 1 de la sortie 2
ResOut2          ; mise à 0 de la sortie 2
SetOut3          ; mise à 1 de la sortie 3
ResOut3          ; mise à 0 de la sortie 3

RdPosition       ; lecture de la position actuelle
RdIdent          ; lecture de l'identification du module
    
```

Il n'y pas de commande FB 'SetZero' pour définir une position quelconque à zéro. On peut dans ce cas utiliser le registre 'rPosAbs_x' dans lequel on charge la position 0 avec

```

Ld      rPosAbs_x
        0
    
```

Attention : l'axe doit être à l'arrêt pendant cette opération.

L'appel du bloc de fonctions HOME revêt toujours la forme suivante :

CFB	HOME		
k	1	; Param. 1 : n° de module	(k 1 à k 16)
r	888	; Param. 2 : point de référence	(REF)
	1	; Param. 3 : sens de la recherche	(0 = nég./1 = pos.)
	0	; Param. 4 : sens du déplacement libre à partir de REF	(0 = nég./1 = pos.)
	2	; Param. 5 : plage de fréquence	(0 à 3)
	10	; Param. 6 : Vmin	(1 à 255)
	100	; Param. 7 : Vmax	(1 à 255)
	20	; Param. 8 : accélération	(1 à 255)
	30	; Param. 9 : tempo (en s)	(1 à 65535 0 = pas de tempo)

Indicateurs généraux :

```

rDiag          ; erreur de paramètre
fPar_Err
    
```

Indicateurs spécifiques au module :

```

fHomeErr_x     ; erreur Retour à l'origine
fEndHome_x     ; bas pendant la fonction Retour à l'origine

rPosAbs_x      ; position absolue
EmergTrig_x    ; déclenchement arrêt d'urgence
OnDest_x       ; cible atteinte
LSxTrig_x      ; déclenchement fin de course LS1 ou LS2
Emerg_x        ; arrêt d'urgence
LS1_x          ; fin de course LS1
REF_x          ; interrupteur de référence REF
LS2_x          ; fin de course LS2
    
```

avec « _x » = numéro de module

Les adresses effectives des registres et indicateurs figurent dans le fichier Project.map (à des fins de mise au point).

7.1.3 Définition des paramètres

Les valeurs de fréquence minimale (Vmin), maximale (Vmax) et d'accélération, comprises entre 1 et 255, peuvent être choisies parmi les quatre plages (0 à 3) des deux tables ci-après.

Précisons que ces trois paramètres doivent appartenir à la même plage.

Fréquences minimale « Vmin » et maximale « Vmax » (en Hz)

Valeur	Plage 0	Plage 1	Plage 2	Plage 3
1 (résolution)	9,5	19	38	76
2	19	38	75	152
3	28	57	114	229
4	38	76	153	305
5	48	95	191	381
6	57	114	229	458
8	76	153	305	610
10	95	191	381	763
15	143	286	572	1144
20	190	381	763	1526
25	238	477	954	1907
30	286	572	1144	2289
40	381	763	1526	3052
50	477	964	1907	3814
60	572	1144	2289	4577
70	667	1335	2670	5340
80	763	1526	3052	6103
90	858	1717	3433	3866
100	954	1907	3815	7629
110	1049	2098	4196	8392
120	1144	2289	4577	9155
130	1240	2480	4959	9918
140	1335	2670	5340	10680
150	1430	2861	5722	11443
160	1526	3052	6103	12206
170	1621	3243	6485	12969
180	1716	3433	6866	13732
190	1812	3624	7248	14495
200	1907	3815	7629	15258
210	2002	4006	8010	16021
220	2098	4196	8392	16784
230	2193	4387	8773	17546
240	2288	4578	9155	18309
250	2384	4769	9536	19072
255	2431	4864	9727	19454

Exemple pour des valeurs non indiquées dans la table ci-dessus :

$$\text{par ex. } 1000\text{Hz : valeur} = \frac{1000}{9,5} = 105 \rightarrow \text{plage 0}$$

Accélération (en Hz/s)

Valeur	Plage 0	Plage 1	Plage 2	Plage 3
1 (résolution)	152939	305911	611821	1223642
2	76470	152955	305911	611821
3	50980	101970	203940	407881
4	38235	76478	152955	305911
5	30588	61182	122364	244728
6	25489	50985	101970	203940
8	19117	38239	76478	152955
10	15294	30991	61162	122364
15	10196	20394	40788	81776
20	7647	15296	30591	61162
25	6118	12236	24473	48946
30	5098	10197	20394	40788
40	3823	7648	15296	30591
50	3059	6118	12236	24473
60	2549	5099	10197	20394
70	2185	4370	8740	17481
80	1912	3824	7648	15296
90	1699	3399	6798	13596
100	1529	3059	6118	12236
110	1390	2781	5562	1124
120	1274	2549	5099	10197
130	1176	2353	4706	9413
140	1092	2185	4370	8740
150	1020	2039	4079	8158
160	956	1920	3824	7648
170	900	1800	3599	7198
180	850	1700	3399	6798
190	805	1610	3220	6440
200	765	1530	3059	6118
210	728	1457	2913	5826
220	695	1391	2781	5562
230	665	1330	2660	5320
240	637	1275	2549	5099
250	612	1224	2447	4895
255	600	1200	2400	4800

Exemple : Déterminons les valeurs de Vmin, Vmax et d'accélération correspondant à :
 Vmin = 100 Hz,
 Vmax = 3000 Hz,
 Accélération = 6000 Hz/s

La plage choisie est toujours celle offrant la valeur la plus proche, soit dans notre cas la plage 1. On obtient ainsi :

Valeur de Vmin → 5 (95 Hz)
 Valeur de Vmax → entre 150 et 160 = env. 157
 Valeur de l'accélération → entre 50 et 60 = env. 51

7.2 Programmer en FUPLA avec les boîtes de fonctions « FBoxes »

(En préparation)

7.3 Programmer en GRAFTEC avec les boîtes de fonctions

(En préparation)

8. Typologie des erreurs et diagnostic

8.1 Erreurs de définition

Les erreurs de définition du fichier « D2H210_b.mba » sont repérées au cours de l'assemblage :

- Nombre de modules (« NbrModules ») inférieur à 1 → Pas d'assemblage du code et affichage de l'avertissement suivant dans la fenêtre « Make » :

"Remark : No H210 used (NbrModules = 0 in D2H210_B.MBA)"

- Nombre de modules (« NbrModules ») supérieur à 16 → Pas d'assemblage du code et affichage du message d'erreur suivant dans la fenêtre « Make » :

"Error : more than 16 Modules H210 defined (NbrModules = 0...16)"

- Code de commande du bloc de fonctions EXEC incorrect (par ex., « RdIdenti » au lieu de « RdIdent ») → Signalisation d'erreur par l'assembleur :

"Symbol not defined 'H210.RdIdenti'"

(provoquant l'impression « H210 » à partir de « \$group H210 »)

- Absence de directive « \$group H210 » → Signalisation d'erreur par l'assembleur pour chaque commande et registre/indicateur utilisé dans le programme :

"Symbol not defined"

8.2 Erreurs de traitement

8.2.1 Paramètres incorrects

Dans le bloc de fonctions EXEC, seul le code de la commande (paramètre 2) est contrôlé. A l'inverse, les paramètres 1 (n° de module) et 3 (registre source/cible) sont exclus du contrôle pour éviter d'allonger les temps d'exécution.

Dans les blocs de fonctions INIT et HOME, tous les paramètres sont vérifiés pour s'assurer que leurs valeurs se situent bien dans la plage autorisée (par ex., 0 à 3 pour la plage de fréquence). Si un paramètre est hors bornes, il est forcé à sa valeur minimale (sauf l'accélération, forcée à 255), l'indicateur d'erreur « fPar_Err » est positionné et le registre de diagnostic « rDiag » est chargé avec le code d'erreur correspondant.

L'indicateur fPar_Err n'est pas remis à 0 dans le bloc de fonctions, mais dans le bloc d'organisation des exceptions XOB 16 ou dans l'étape INIT.

Codage de l'erreur :

```

rDiag  bit 31 ..... 24 23 ..... 16 15 ..... 8 7 ..... 0
        \ Réservé /   \ N° bloc /   \ N° param./   \ N° module/
        (avec  INIT   = Bloc 1,
              EXEC   = Bloc 2,
              HOME   = Bloc 3)

```

Exemple : Si la plage de fréquence (paramètre 5) du bloc de fonctions HOME du module n° 2 est incorrecte (c'est-à-dire supérieure à 3), le registre rDiag prend la valeur 00 03 05 02 hexa.

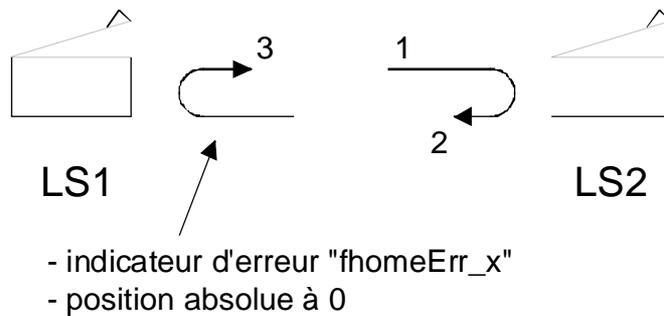
Le registre rDiag est écrasé à chaque apparition de paramètre incorrect et contient toujours la dernière erreur. Il doit donc être lu dès que l'indicateur fPar_Err signale une erreur. Les adresses absolues de rDiag et de fPar_Err sont consultables dans le fichier « project.MAP » ; cela permet, à la mise en service, de localiser une erreur avec le programme de mise au point, en procédant comme suit :

- Exécuter le programme jusqu'à ce que fPar_Err passe à l'état haut,
- Afficher rDiag en hexa,
- Effacer fPar_Err.

8.2.2 Erreur de Retour à l'origine

Si l'axe ne rencontre pas de point de référence (en raison, par exemple, d'un défaut de l'interrupteur REF), l'indicateur d'erreur « fHomeErr_x » est positionné, le déplacement s'arrête, la position absolue est forcée à 0 et le bloc de fonctions HOME est interrompu.

Absence de l'interrupteur de référence ou mauvais connexion :



Si le bloc de fonctions HOME est interrompu à l'échéance de la temporisation prédéfinie par le paramètre 9, le registre de diagnostic « rDiag » est chargé avec le code 9 correspondant.

L'indicateur fHomeErr_x est défini pour chaque module (« x » étant le n° de module) et remis à 0 au lancement de chaque bloc de fonctions HOME. Il doit être interrogé après chaque appel du bloc de fonctions HOME pour s'assurer que l'axe est correctement référencé.

Exemple :

```
CFB      Home          ; Retour à l'origine de l'axe 3
      K 3
      .
      .
      .

STH      fHomeErr_3    ; Interrogation indicateur
                        ; erreur module 3
CFB      H Errorhandl  ; Appel bloc (spécifique utili-
                        ; sateur), si fHomeErr_3 est
                        ; à l'état haut

CFB      Exec          ; Déplacement 1
...

```

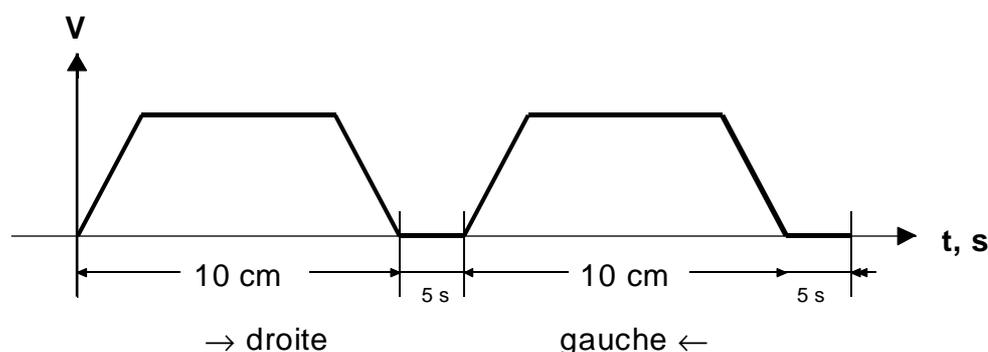
Notes personnelles :

9. Exemples de programmation

9.1 Exemple utilisateur en GRAFTEC et en liste d'instructions (IL)

En s'inspirant de l'exemple du paragraphe 6.1 « Bien démarrer avec la programmation en liste d'instructions » dans lequel l'entrée 0 du PCD (« Démarrage ») est activée, nous supposons qu'un mobile piloté par un moteur pas-à-pas effectue, depuis sa position de départ, un déplacement de 10 cm vers la droite, puis marque une pause de 5 secondes pour finalement revenir en position de départ.

Si l'entrée Démarrage reste activée, cette séquence reprendra depuis le début après une pause de 5 secondes.



Voyons à présent la variante suivante : le déplacement du mobile doit s'arrêter avec l'entrée 1 du PCD (« Arrêt »), puis reprendre avec l'entrée 2 (« Poursuite ») tout en conservant les rampes d'accélération et de freinage.

Quelques caractéristiques techniques de notre exemple :

Tours/révolution :	48
Pas de la vis :	1 mm/U
Fréquence Start/Stop (V_{min}) :	40 Hz
Fréquence maxi (V_{max}) :	1 000 Hz
Accélération :	2 000 Hz/s

Ce projet s'intitule « TEST-H2 » et le programme utilisateur, édité en GRAFTEC, est baptisé « Move-01.sfc ».

Les paramètres de la commande d'axe ' V_{min} ', ' V_{max} ' and ' Acc ' (accélération) doivent être adaptés aux moteurs utilisés et ne peuvent par conséquent **pas** être pris tels qu'ils sont dans les exemples ci-après.

Démarche :

- Copiez le fichier « D2H210_b.mba » dans le répertoire du projet TEST-H2 et définissez l'adresse de base du PCD2.H210 (64, par ex.).
- Paramétrez les fréquences mini et maxi ainsi que l'accélération en vous aidant du paragraphe 7.1.3 pour choisir les valeurs suivantes :

Vmin	→	4
Vmax	→	≈ 105
Accélération	→	≈ 75

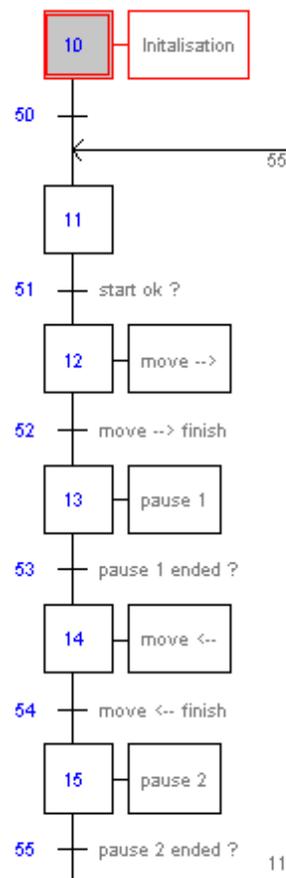
- Définissez le nombre de pas, soit :

$$100 \text{ mm} \times 48 \text{ pas} = 4\,800 \text{ pas}$$

En mode relatif, il faut alors saisir 4800 pour le mouvement d'aller (vers la droite), puis -4800 pour le mouvement de retour (vers la gauche).

En mode absolu, il faut aussi saisir 4800 pour l'aller, mais cette fois 0 (zéro) pour le retour.

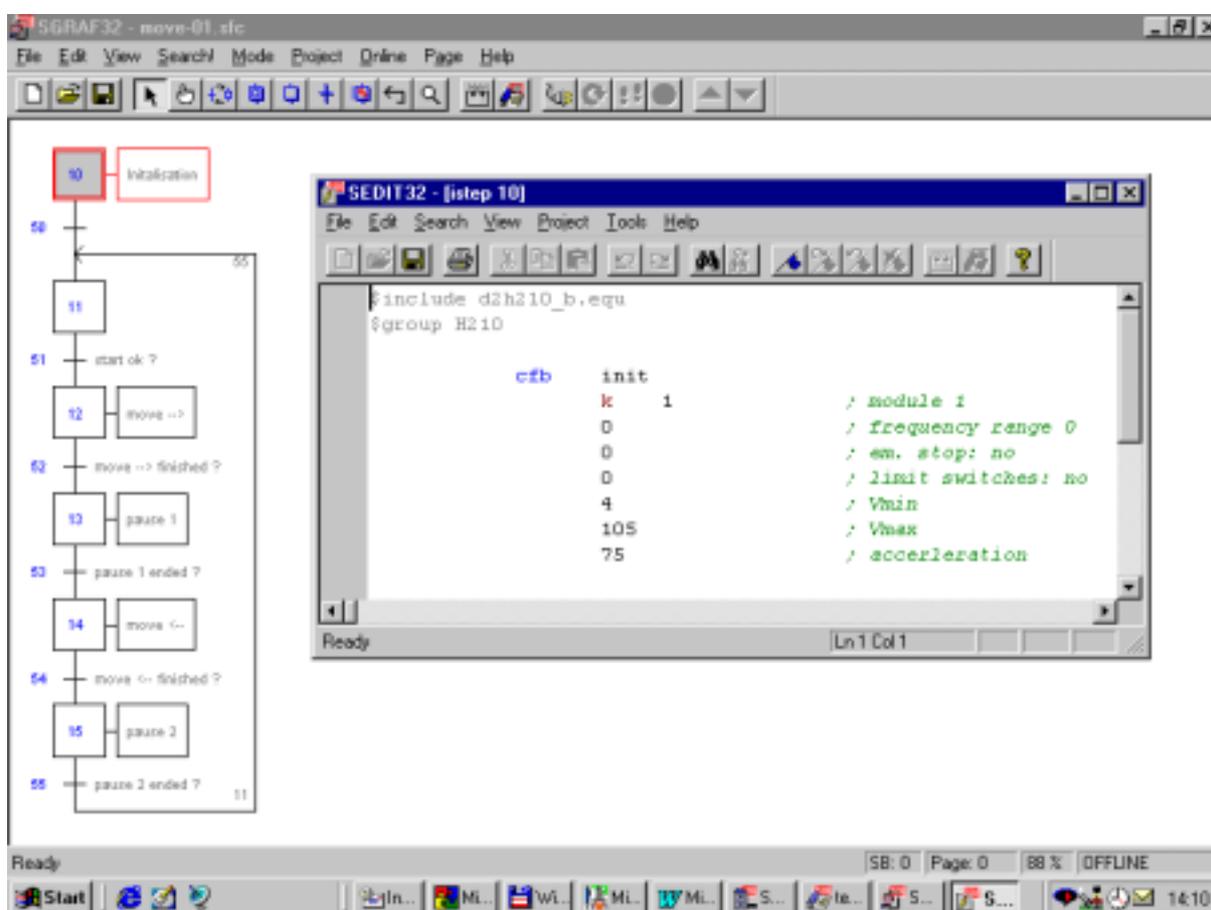
- Réalisez le GRAFTEC correspondant en incluant des commentaires :



Codez chaque étape et chaque transition à l'aide de l'éditeur SEDITWIN.



(Pour éditer en FUPLA à l'aide de l'éditeur SFUP, reportez-vous au paragraphe 9.2).



Le code de chaque étape et transition s'édite de la même façon.

Il est possible, par exemple, de choisir le fichier Move-01.sfc dans l'éditeur de bloc notes et de l'imprimer (Cf. pages suivantes). L'impression du GRAFTEC ainsi que celle du fichier de code constituent la documentation du programme.

```

SB      0
;-----
IST     10          ; Initialisation
        O 50
$include d2h210_b.equ
$group H210

cfb          init
k         1        ; module n° 1
          0        ; plage de fréquence : 0
          0        ; arrêt d'urgence : non
          0        ; fins de course : non
          4        ; Vmin
          105       ; Vmax
          75        ; accélération

EST      ;10
;-----
ST       11
        I 50
        I 55          ;pause 2 terminée ?
        O 51          ;démarrage ok ?
EST      ;11
;-----
ST       12          ; déplacement -->
        I 51          ; démarrage ok ?
        O 52          ; déplacement --> terminé ?
ld      r         1
          4800

cfb          exec
k         1        ; module n° 1
          lddestrel
r         1

cfb          exec
k         1        ; module n° 1
          start
          rnotused
EST      ;12
;-----
ST       13          ; pause 1
        I 52          ; déplacement --> terminé ?
        O 53          ; pause 1 terminée ?
ld      t         1
          50          ; 5 s
EST      ;13
;-----
ST       14          ; déplacement <--
        I 53          ; pause 1 terminée ?
        O 54          ; déplacement <-- terminé ?
ld      r         1
          -4800

cfb          exec
k         1        ; module n° 1
          lddestrel
r         1

cfb          exec
k         1        ; module n° 1
          start
          rnotused
EST      ;14

```

```

;-----
ST    15          ; pause 2
      I 54          ; déplacement <-- terminé ?
      O 55          ; pause 2 terminée ?
ld    t          1
      50          ; 5 s
EST   ;15
;-----
TR    50
      I 10          ; Initialisation
      O 11
ETR   ;50
;-----
TR    51          ; démarrage ok ?
      I 11
      O 12          ; déplacement -->
sth   i          0
ETR   ;51
;-----
TR    52          ; déplacement --> terminé ?
      I 12          ; déplacement -->
      O 13          ; pause 1
cfb   exec
      k 1          ; module 1
      rdposition
      r 0
dsp   r 0        ; affichage
sth   ondest_1
ETR   ;52
;-----
TR    53          ; pause 1 terminée ?
      I 13          ; pause 1
      O 14          ; déplacement <--
cfb   exec
      k 1          ; module 1
      rdposition
      r 0
dsp   r 0        ; affichage
stl   t 1
ETR   ;53
;-----
TR    54          ; déplacement <-- terminé ?
      I 14          ; déplacement <--
      O 15          ; pause 2
cfb   exec
      k 1          ; module n° 1
      rdposition
      r 0
dsp   r 0        ; affichage
sth   ondest_1
ETR   ;54

```

```

;-----
TR    55          ; pause 2 terminée ?
      I 15          ; pause 2
      O 11

cfb   exec
      k 1          ; module n° 1
      rdposition
      r 0

dsp   r 0          ; affichage

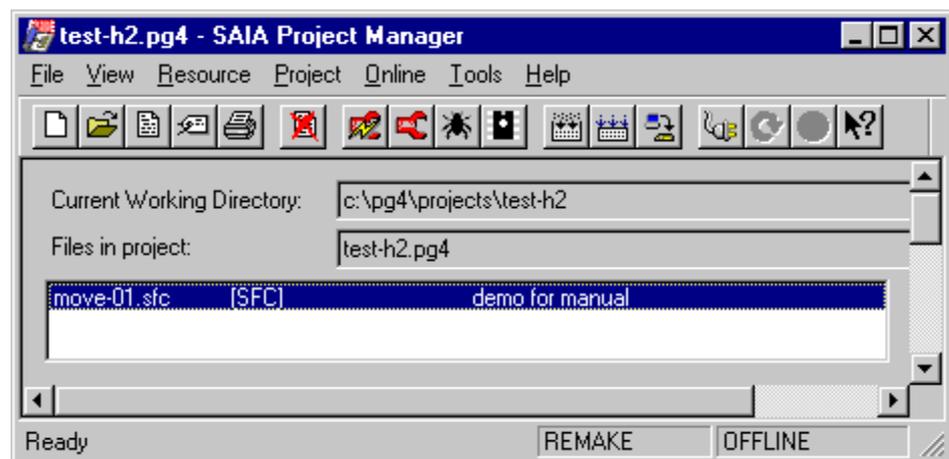
stl   t 1

$endgroup
ETR   ;55

ESB   ;0

```

La commande « Project – Build » du gestionnaire de projet permet de traiter le programme et de le télécharger dans le PCD.



La présence d'un module afficheur PCD2.F5x0 permet non seulement de visualiser en direct le déplacement de l'axe, mais aussi de lire l'état du compteur.

Des exemples de programmation similaires se trouvent également sur la disquette PCD9.H21E.

Variante avec arrêt suivi d'une reprise du mouvement :

Si, comme indiqué dans le descriptif de la tâche, il faut arrêter le mouvement avec l'entrée 1 du PCD, puis le poursuivre avec l'entrée 2, les transitions 52 (aller) et 54 (retour) doivent être programmées comme suit :

```

;-----
TR    52                ; déplacement --> terminé ?
      I 12              ; déplacement -->
      O 13              ; pause 1

cfb   exec
      k    1            ; module n° 1
      rdposition
      r    0

dsp   r    0            ; affichage

sth   i    1            ; arrêt
dyn   f    1
set   f    10           ; indicateur arrêt-mémorisation
cfb   h    exec
      k    1            ; module n° 1
      stop
      rnotused

sth   i    2            ; poursuite
dyn   f    2
res   f    10           ; indicateur arrêt-mémorisation
cfb   h    exec
      k    1            ; module n° 1
      continue
      rnotused

sth   ondest_1
anl   f    10           ; indicateur arrêt-mémorisation
ETR   ;52

```

En principe, les transitions 52 et 54 de cet exemple ne font qu'attendre la fin du déplacement (sth ondest_1). Le GRAFTEC présente à cet égard une particularité : après chaque transition **non satisfaite**, le programme revient au COB appelant et, au cours du cycle de programme suivant, re-traite **dans sa totalité** la transition non satisfaite. Cela permet de rafraîchir en permanence l'affichage.

S'il faut ensuite interrompre la sortie d'impulsions au cours d'un mouvement, cela doit également avoir lieu dans la transition en attente de la fin de ce mouvement. Si la commande Arrêt est activée, l'indicateur « ondest_x » passe aussi à l'état haut et la transition est considérée comme satisfaite. La commande Arrêt est donc stockée entre-temps dans un indicateur d'arrêt-mémorisation dont l'état logique indique si le parcours a bel et bien été effectué dans sa totalité. Cet indicateur est remis à 0 par la fonction Poursuite (« Continue »).

Si besoin, voilà une bonne façon d'interrompre parfaitement un déplacement, c'est-à-dire avec des rampes de freinage et d'accélération et sans perte de pas.

Mise en œuvre des interrupteurs de référence et des fins de course

Cette exemple se trouve sur la disquette PCD9.H21E.

```
; *****  
; * Exemple de programme pour FB's H210 PG4 *  
; *****  
;  
; FileName :      H210_Ex4.SRC  
; Autor :        Nguyen T.D.  03.01.99  
;  
; Système requis:  
; 1 module d'entrée à l'adresse 0,  
; 1 module H210 avec l'adresse de base définie dans le  
;   fichier D2H210_B.MBA  
;  
; Description:  
;  
; - Ce programme décrit la façon de travailler avec la  
;   commande Retour à l'origine (HOMING Procedure).  
;   (Consultez la description détaillée dans ce manuel)  
;   Si l'entrée I3 est H, le programme exécute la commande  
;   Retour à l'origine (HOMING Procedure).  
;   Si il n'existe pas d'interrupteurs LS, utilisez pour la  
;   simulation : E1=I5 = LS1, E2=I6 = LSRef, E3=I7 = LS2 ;  
;   si ils sont disponibles, raccordez-les correctement.  
; - Ce programme doit être linker soit avec l'exemple  
;   H210_Ex2.SFC ou avec H210_Ex3.SFC. Pour démarrer le  
;   programme utilisez les entrées du programme respectif  
;   (c.-à-d. I0 = start, I1 = stop, I2 = continue) ;  
;   Pour plus de détails, consultez le chapitre 9 du manuel.  
; - L'utilisateur doit adapter les paramètres du bloc de  
;   fonctions « FB INIT » dans l'étape initiale (IST) de  
;   l'exemple H210_Ex2.SFC ou H210_Ex3.SFC comme suit:  
;   Set Emergency stop = 1, et Limit switch = 1).  
; - La plage de travail est limitée au moyen de LS1 et LS2,  
;   en dehors de cette plage l'indicateur LS1Trigl est haut  
;   et la sortie A3 du module est mis à 1.  
;   La sortie A3 est également mise à 1 lorsque l'arrêt  
;   d'urgence est activé (E0 = 0).  
;   Dans la pratique, une sortie normale peut être activée  
;   dans le programme utilisateur pour enclencher une  
;   alarme ou un avertissement ainsi que pour d'autres  
;   intentions dans de telles situations.  
; - L'entrée d'arrêt d'urgence est simulée avec E0 = I4.  
;
```

```

$include D2H210_B.EQU ; pour PG4 V2.0 le chemin d'accès
                        ; complet n'est plus nécessaire
$group H210

XOB          16          ; démarrage bloc de fonctions

CFB          Exec        ; FB Exec
              K 1         ; n° du module
              ResOut3     ; commande: mise à 0 sortie A3
              rNotUsed    ; registre non utilisé

EXOB

;-----

COB          0           ; bloc cyclique
              0

LD           R 100       ; Charger position Offset pour REF
              0

STH          I 3         ; démarrage Retour à l'origine
DYN          F 1000
CFB          H HOME      ; FB Home
              K 1         ; n° du module
              R 100       ; registre offset
              1           ; sens de recherche (0=Neg.,1=Pos.)
              0           ; sens déplac. libre(0=Neg.,1=Pos.)
              1           ; plage de fréquence
              16          ; Vmin
              80          ; Vmax
              100         ; accélération
              400         ; timeout (délai maxi en secondes
                        ; avant arrêt)

STH          LSxTrig_1   ; LS1 ou LS2 dépassé ?
ORH          EmergTrig_1 ; ou arrêt d'urgence activé ?
CFB          H Exec      ; FB Exec
              K 1         ; n° du module
              SetOut3     ; commande: mise à 1 sortie A3
              rNotUsed    ; registre non utilisé

ECOB

$endgroup
;-----

```

Notes personnelles :

9.2 Exemple utilisateur en FUPLA

(En préparation)

9.3 Exemple utilisateur en GRAFTEC et en FUPLA

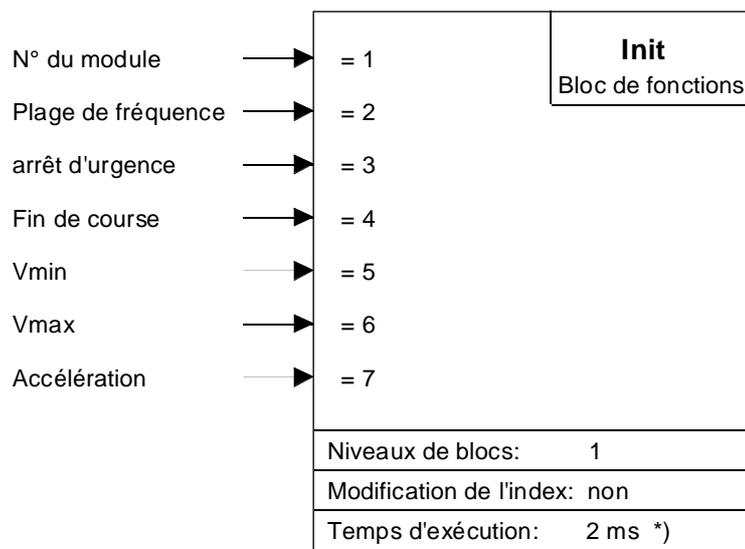
(En préparation)

Annexe A Récapitulatif des blocs de fonctions et des commandes PCD2.H210 en langage IL

Bloc de fonctions « INIT »

INIT

Initialisation du module PCD2.H210



*) mesuré avec PCD2.M120

Description :

Ce bloc définit le paramétrage du PCD2.H210 et lit son adresse de base du fichier D2H210_b.mba.

Les valeurs de fréquence mini (Vmin), fréquence maxi (Vmax) et d'accélération sont modifiables à tout moment et ne sont pas considérées comme des valeurs maximales.

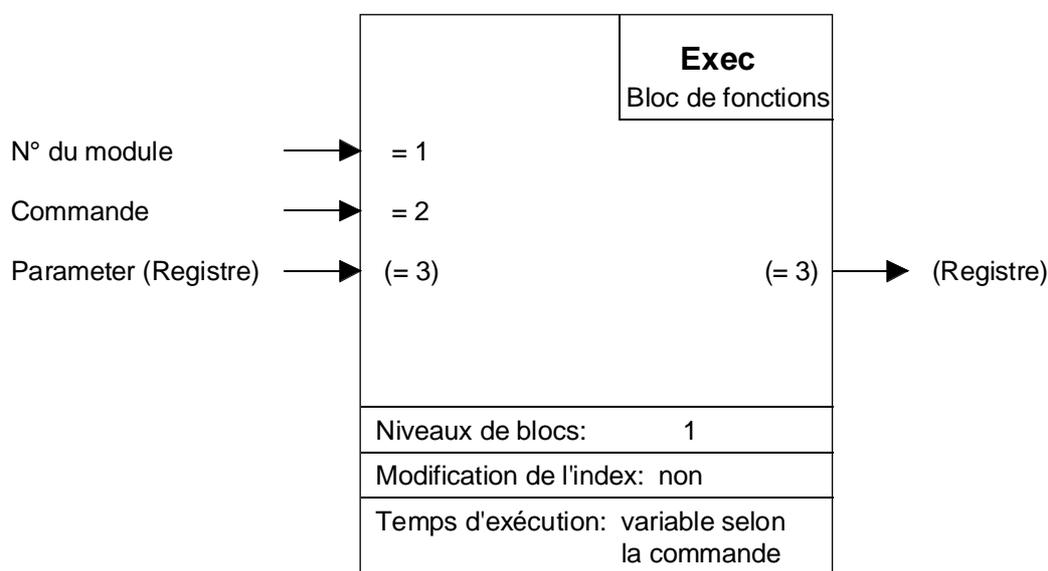
Le paramètre 1 s'exprime sous forme de constante K ; tous les autres paramètres sont au format entier.

Par.	Désignation	Type	Format	Plage	Remarques
= 1	N° de module	K	K n	K 1 à K 16	
= 2	Plage de fréquence		Entier	0 à 3	0 = 9.5 à 2 431 Hz 1 = 19 à 4 864 Hz 2 = 38 à 9 727 Hz 3 = 76 à 19 454 Hz
= 3	Config. arrêt d'urgence		Entier	0 / 1	0 = HS / 1 = ES
= 4	Config. fin de course		Entier	0 / 1	0 = HS / 1 = ES
= 5	Fréquence Start/Stop (Vmin)		Entier	1 à Vmax	Contrôle de saisie
= 6	Fréquence après rampe d'accélération (Vmax)		Entier	Vmin à 255	Contrôle de saisie
= 7	Accélération		Entier	255 à 1	Contrôle de saisie

Bloc de fonctions « EXEC »

EXEC

Exécution d'une commande PCD2.H210



Description :

Ce bloc de fonctions permet d'envoyer des commandes au PCD2.H210.

Le numéro de module (paramètre 1) doit être une constante K (dans la plage K 1 à K 16).

L'adresse de base est définie dans le fichier D2H210_b.mba.

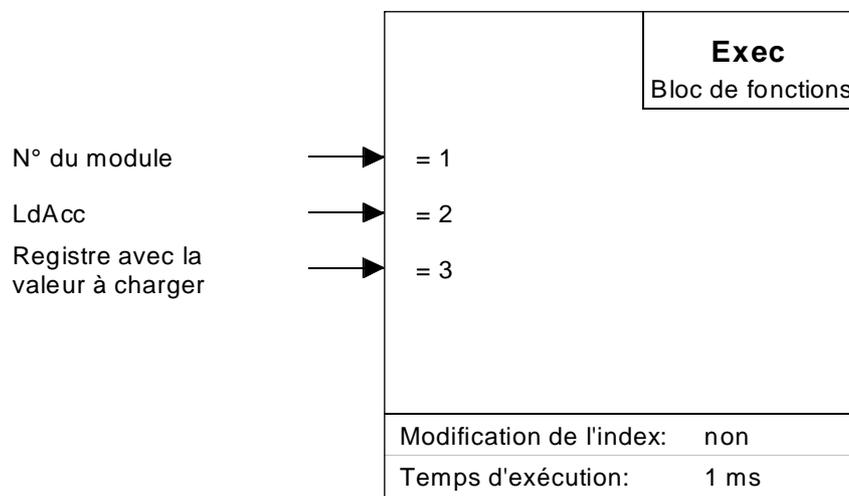
Ce bloc de fonctions accepte jusqu'à 16 modules PCD2.H210 par automate PCD.

Il est constitué de plusieurs commandes (paramètre 2), dont la description fait l'objet des pages suivantes.

Le paramétrage d'une commande (par ex., la valeur de l'accélération pour la commande « LdAcc ») est transféré dans un registre (paramètre 3). Si la commande n'a pas de paramètre (comme c'est le cas, par exemple, de la commande de démarrage « Start »), on utilise n'importe quel registre ou le registre « rNotUsed ».

Commandes du bloc de fonctions « EXEC »

LdAcc Chargement de l'accélération



Description :

Cette commande charge dans le registre d'entrée la valeur de l'accélération, qui s'appliquera dès la prochaine commande de démarrage « Start ». Rappelons que les rampes d'accélération et de freinage sont symétriques et que l'accélération dépend du choix de la plage de fréquence (Cf. table du paragraphe 7.1.3).

Calcul de l'accélération effective :

$$\text{Accél. effective (plage de fréq. 3)} = \frac{76.30 \text{ [pas/s]} * 16\,000 \text{ [1/s]}}{\text{Valeur de l'accélération}} \text{ [pas/s}^2\text{]}$$

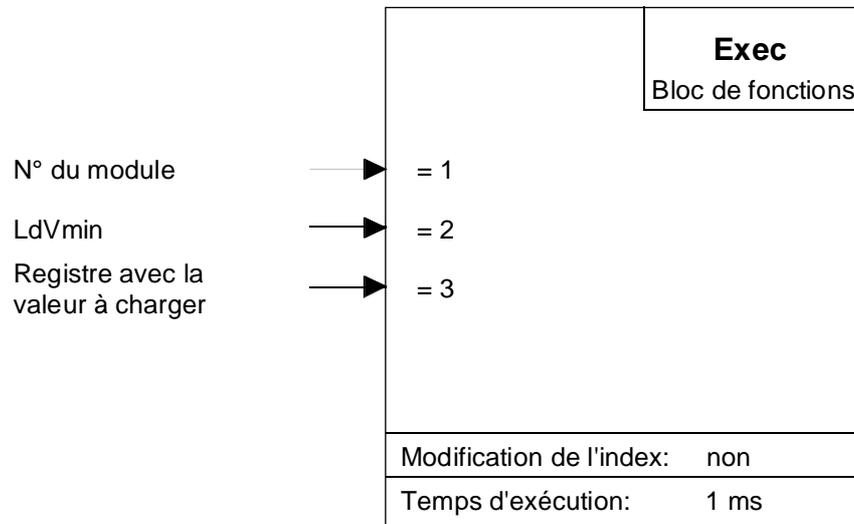
et ainsi de suite pour 38.15 et 19.08 [pas/s] jusqu'à :

$$\text{Accél. effective (plage de fréq. 0)} = \frac{9.54 \text{ [pas/s]} * 16\,000 \text{ [1/s]}}{\text{Valeur de l'accélération}} \text{ [pas/s}^2\text{]}$$

Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Plage	Remarques
= 1	N° du module	K		1 à 16	
= 2	Commande: LdAcc				
= 3	Accélération	R	Entier	255 à 1	255 = mini et 1 = maxi Les valeurs hors bornes ne sont pas prises en compte.

LdVmin Chargement de la fréquence Start/Stop (Vmin)



Description :

Cette commande charge dans le registre d'entrée la valeur de la fréquence Start/Stop (Vmin), qui s'appliquera dès la prochaine commande de démarrage « Start ».

La progression de la fréquence (c'est-à-dire les différents paliers permettant de passer d'un étage de fréquence à l'autre) dépend de la plage retenue (Cf. § 7.1.3), ce qui donne :

Plage 0 (9.5 à 2 431 Hz)	→	Résolution :	9.54 Hz
Plage 1 (19 à 4 864 Hz)	→	Résolution :	19.08 Hz
Plage 2 (38 à 9 727 Hz)	→	Résolution :	38.15 Hz
Plage 3 (76 à 19 454 Hz)	→	Résolution :	76.30 Hz

Calcul de la fréquence de démarrage-arrêt effective :

Plage 0 : Fréquence Start/Stop = 9.54 Hz * valeur de Vmin

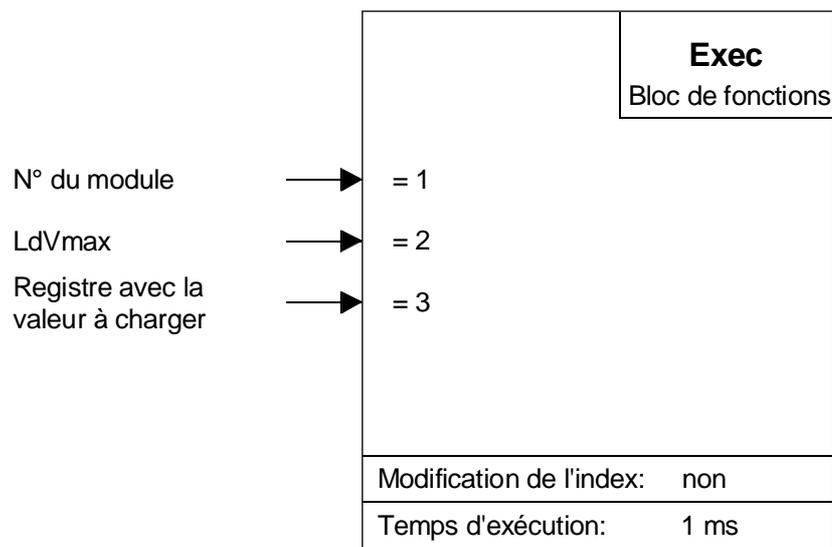
Et ainsi de suite pour 19.08 Hz et 38.15 Hz jusqu'à :

Plage 3 : Fréquence Start/Stop = 76.30 Hz * valeur de Vmin

Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Plage	Remarque
= 1	N° du module	K		1 à 16	
= 2	Commande: LdVmin				
= 3	Fréquence Start/Stop	R	Entier	1 à Vmax	Les valeurs hors bornes ne sont pas prises en compte.

LdVmax Chargement de la fréquence maximale (Vmax)



Description :

Cette commande charge dans le registre d'entrée la valeur de la fréquence maximale (Vmax) qui peut être atteinte après une rampe d'accélération. Cette valeur s'appliquera dès la prochaine commande de démarrage « Start ».

La progression de la fréquence (c'est-à-dire les différents paliers permettant de passer d'un étage de fréquence à l'autre) dépend de la plage retenue (Cf. § 7.1.3), ce qui donne :

Plage 0 (9.5 à 2 431 Hz)	→	Résolution : 9.54 Hz
Plage 1 (19 à 4 864 Hz)	→	Résolution : 19.08 Hz
Plage 2 (38 à 9 727 Hz)	→	Résolution : 38.15 Hz
Plage 3 (76 à 19 454 Hz)	→	Résolution : 76.30 Hz

Calcul de la fréquence maximale effective :

Plage 0 : Fréquence maxi = 9.54 Hz * valeur de Vmax

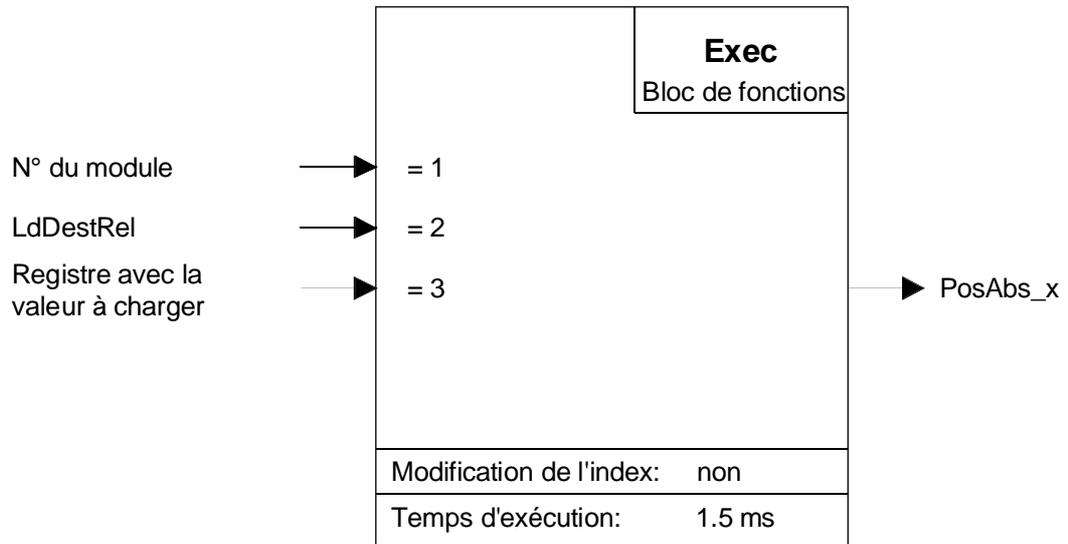
Et ainsi de suite pour 19.08 Hz et 38.15 Hz jusqu'à :

Plage 3 : Fréquence maxi = 76.30 Hz * valeur de Vmax

Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Plage	Remarque
= 1	N° du module	K		1 à 16	
= 2	Commande: LdVmax				
= 3	Fréquence maximale	R	Entier	Vmin à 255	Les valeurs hors bornes ne sont pas prises en compte.

LdDestRel Chargement du déplacement en mode relatif



Description :

Cette commande charge dans le registre d'entrée le nombre de pas nécessaires au déplacement suivant. Cette valeur, toujours au format entier (positive ou négative), s'applique dès la prochaine commande de démarrage « Start ».

Cette valeur ne doit pas dépasser 2^{24} et ne fonctionne qu'en mode relatif.

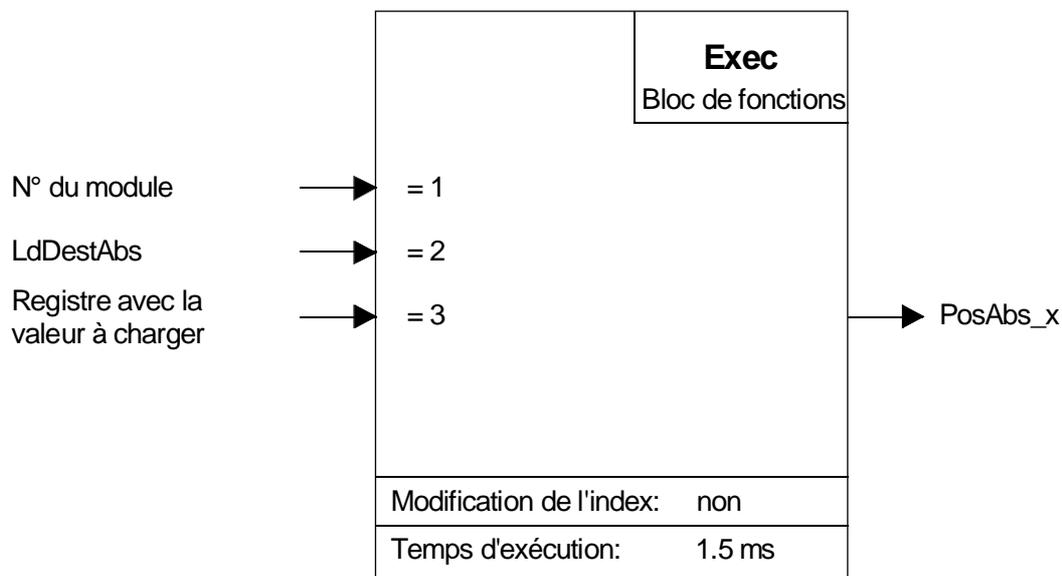
Précisons que LdDestRel se charge également de calculer la position absolue (registre « PosAbs_x ») à atteindre après exécution du déplacement.

Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Valeur	Remarques
= 1	N° du module	K		1 à 16	
= 2	Commande: LdDestRel				
= 3	Cible (relative)	R	Entier	24 bits	$\pm 16\,777\,215$
PosAbs_x	Position absolue	R	Entier	32 bits	-2^{31} à $+2^{31}-1$

LdDestAbs

Chargement du déplacement en mode absolu



Description :

Cette commande charge dans le registre d'entrée la cible du déplacement suivant, cette valeur s'appliquant dès la prochaine commande de démarrage « Start ».

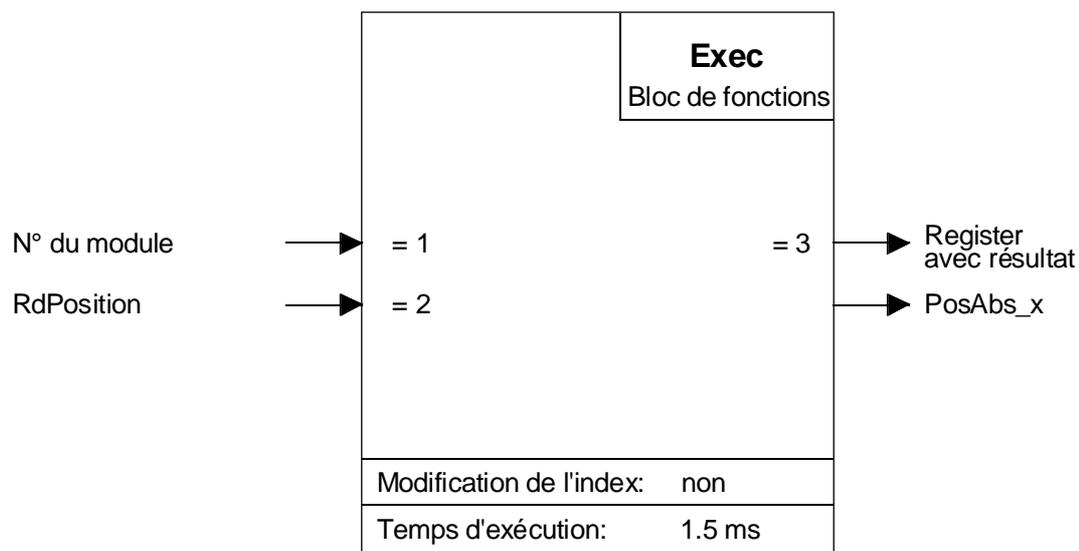
Elle calcule le déplacement relatif à effectuer pour atteindre la position cible; la valeur correspondante s'exprime toujours sous forme de nombre entier positif. La valeur de ce déplacement relatif ne doit pas dépasser 2^{24} .

Précisons que LdDestAbs se charge également de calculer la position absolue à atteindre (registre « PosAbs_x ») après exécution du déplacement.

Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Valeur	Remarques
= 1	N° du module	K		1 à 16	
= 2	Commande: LdDestAbs				
= 3	Cible (absolue)	R	Entier	32 bits	-2^{31} à $+2^{31}-1$, mais 24 bits maxi en déplacement relatif.
PosAbs_x	Position absolue	R	Entier	32 bits	-2^{31} à $+2^{31}-1$

RdPosition Lecture de la position réelle (actuelle)



Description :

Cette commande lit l'état du compteur du module pour connaître le nombre d'impulsions restant à générer avant la fin du déplacement. Le résultat de cette lecture est toujours positif.

Précisons que RdPosition se charge également de calculer la position réelle absolue, rafraîchie dans le registre « PosAbs_x ».

Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Valeur	Remarques
= 1	N° du module	K		1 à 16	
= 2	Commande: RdPosition				
= 3	Registre PCD avec le nombre d'impulsions restant à générer	R	Entier	24 bits	0 à 16 777 215
PosAbs_x	Position absolue	R	Entier	32 bits	-2^{31} à $+2^{31}-1$

Start

Démarrage du déplacement

		Exec	
		Bloc de fonctions	
N° du module	→	= 1	
Start	→	= 2	
non utilisé	→	= 3	
		Modification de l'index:	non
		Temps d'exécutions:	0.5 ms

Description :

Cette commande transmet au registre de travail les paramètres du déplacement (fréquence Start/Stop (Vmin), fréquence maxi (Vmax), accélération et position cible), puis démarre le mouvement.

Stop

Arrêt du déplacement

		Exec	
		Bloc de fonctions	
N° du module	→	= 1	
Stop	→	= 2	
non utilisé	→	= 3	
		Modification de l'index:	non
		Temps d'exécution:	0.5 ms

Description :

Cette commande arrête le déplacement en cours, en lui appliquant la rampe de freinage normale, au terme de laquelle le moteur est à l'arrêt et l'indicateur « OnDest_x » passe à l'état haut.

L'interruption du déplacement peut prendre fin avec la commande de poursuite « Continue ».

Précisons qu'après une commande « Stop », le module n'est pas bloqué; il peut toujours effectuer un déplacement jusqu'à une nouvelle position à l'aide d'une autre commande Start.

Continue

Poursuite d'un déplacement interrompu

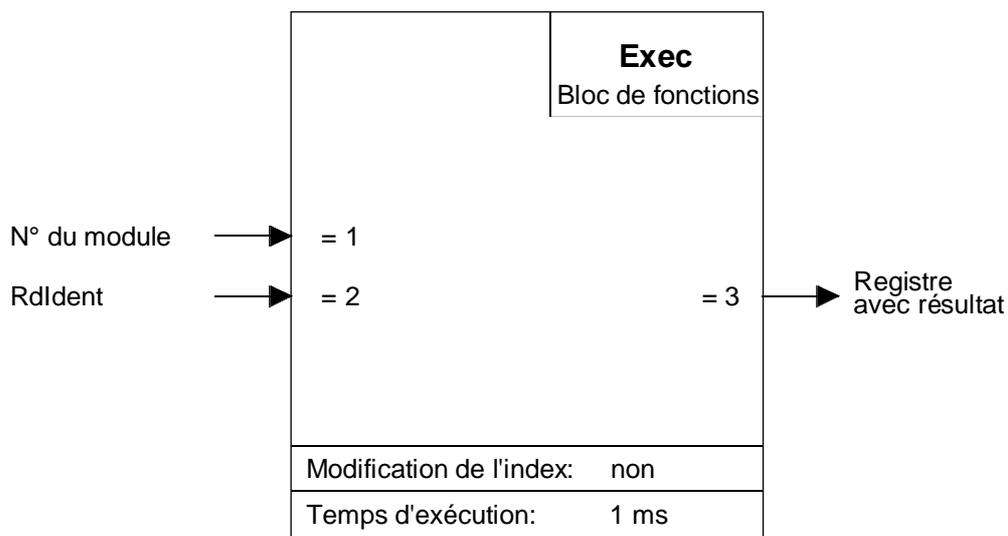
		Exec	
		Bloc de fonctions	
N° du module	→	= 1	
Continue	→	= 2	
non utilisé	→	= 3	
		Modification de l'index:	no
		Temps d'exécution:	2.6 ms

Description :

Cette commande permet de poursuivre un déplacement interrompu avec la commande d'arrêt « Stop ». Elle se charge tout d'abord de lire la position actuelle, de charger le parcours restant, puis de lancer le déplacement.

La commande Continue ne doit être émise que lorsque le moteur est à l'arrêt (l'indicateur « Ondest_x » étant à l'état haut). A défaut, la cible ne sera pas approchée correctement.

RdIdent Lecture de l'identification du module



Description :

Cette commande sert à vérifier le bon fonctionnement du PCD2.H210 et à connaître la version du circuit FPGA. Un fonctionnement correct du module donne la valeur 32xx (Cf. table ci-dessous). En cas de dysfonctionnement, (ou d'erreur d'adressage), on obtient 0.

Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Valeur	Remarque
= 1	N° du module	K		1 à 16	
= 2	Commande: RdIdent				
= 3	Identification du H210	R	Entier	12 bits	0 = défaut

Identification du H210 :

Valeur	Version du FPGA
3200	HD0
3201	HD1
3202	HD2
...	...
3215	HDF

SetOut2 - SetOut3

Mise à 1 des sorties A2 et A3

		Exec Bloc de fonctions
N° du module	→	= 1
SetOut2 - SetOut3	→	= 2
non utilisé	→	= 3
Modification de l'index:		non
Temps d'exécution:		0.6 ms

Description :

Ces deux commandes ont pour effet de mettre à 1 les sorties A2 et A3 respectivement. Celles-ci n'étant pas directement adressées par le bus du PCD1/2, elles ne peuvent être ni relues, ni effacées par les commandes « Clear Outputs » ou « Clear All Elements » du programme de mise au point.

ResOut2 – ResOut3

Mise à 0 des sorties A2 et A3

		Exec Bloc de fonctions
N° du module	→	= 1
ResOut2 - ResOut3	→	= 2
non utilisé	→	= 3
Modification de l'index:		non
Temps d'exécution:		0.6 ms

Description :

Ces deux commandes ont pour effet de mettre à 0 les sorties A2 et A3 respectivement. Celles-ci n'étant pas directement adressées par le bus du PCD1/2, elles ne peuvent être ni relues, ni effacées par les commandes « Clear Outputs » ou « Clear All Elements » du programme de mise au point.

Bloc de fonctions « HOME »

HOME Retour à l'origine (recherche du point de référence)

N° du module	→	= 1	Home Bloc de fonctions
Point de référence	→	= 2	
Sens de la recherche	→	= 3	
Sens du déplacem. libre	→	= 4	
Plage de fréquence	→	= 5	
Vmin	→	= 6	
Vmax	→	= 7	
Accélération	→	= 8	
Tempo (Timeout)	→	= 9	
Niveaux de blocs:			1
Modification de l'index:			non
Temps d'exécution:			tempo maxi.

Description :

Le bloc HOME définit le retour à l'origine et lance la fonction correspondante. Il n'est interrompu que lorsque l'interrupteur de référence est trouvé ou lorsque la temporisation arrive à expiration. Rappelons qu'il faut d'abord initialiser le PCD2.H210 (par le bloc INIT) avant de pouvoir exécuter le bloc HOME. Le flag 'OnDest_x' indique que la procédure est terminée. (Cf. § 5.4 pour un complément d'information).

Par.	Désignation	Type	Format	Plage	Remarques
= 1	N° de module	K	K n	K1 à K 16	
= 2	Point de référence *)	R	Entier	32 Bits	Position absolue
= 3	Sens de la recherche		Entier	0 / 1	0 = nég. / 1 = pos.
= 4	Sens du déplacement libre		Entier	0 / 1	0 = nég. / 1 = pos.
= 5	Plage de fréquence		Entier	0 à 3	Cf. bloc INIT
= 6	Fréquence mini (Vmin)		Entier	1 à Vmax	Contrôle de saisie
= 7	Fréquence maxi (Vmax)		Entier	Vmin à 255	Contrôle de saisie
= 8	Accélération		Entier	255 à 1	Contrôle de saisie
= 9	Tempo (délai maxi, en secondes, avant arrêt)		Entier	0 à 65535 [s]	0 = pas de tempo

*) Ce paramètre permet de définir un offset pour la position de référence.
(Voir page 7-6 pour définir un point quelconque comme zéro).

Notes personnelles :

Annexe B Récapitulatif des boîtes de fonctions et des commandes PCD2.H210 en langage FUPLA

(En préparation)

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 PCD2.H210

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 :