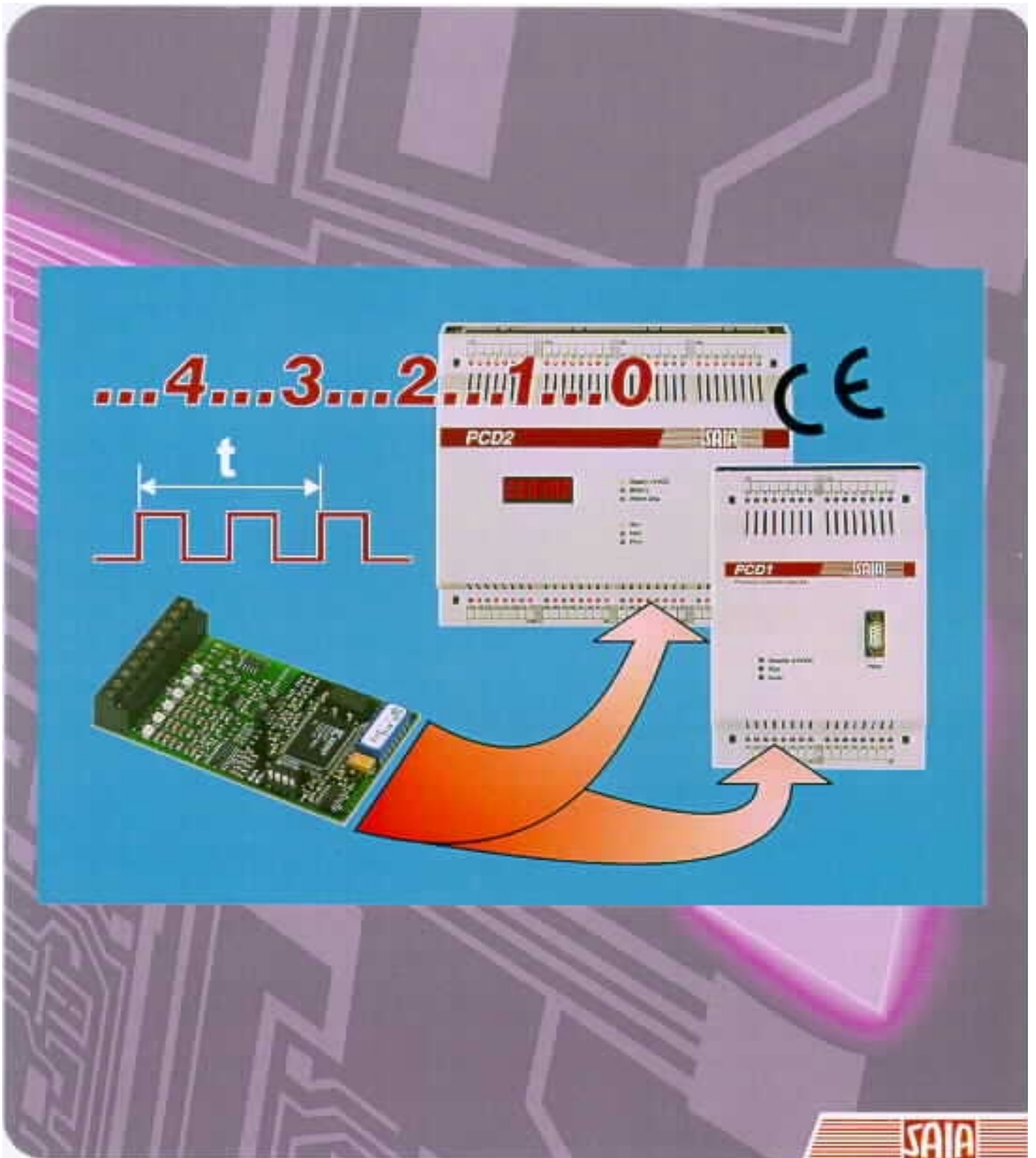


**SAIA® PCD**  
Process Control Devices

## PCD2.H110 Module universel de comptage et de mesure





**SAIA® Process Control Devices**

# **Module universel de comptage et de mesure**

## **PCD2.H110**

SAIA-Burgess Electronics SA 1999. Tous droits réservés  
Edition 26/755 F2 - 06.99

Sous réserve de modifications

# Mise à jour

---

Manuel : Module universel de comptage et de mesure PCD2.H110 - édition F2

Date	Chapitre	Page	Description

# Table des matières

---

	Page
<b>1. Présentation</b>	
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
<b>2. Caractéristiques techniques</b>	
2.1 Caractéristiques matérielles	2-1
2.2 Caractéristiques électriques	2-3
2.3 Caractéristiques de fonctionnement	2-4
<b>3. Constitution</b>	
<b>4. Repérage des bornes, câbles et voyants de signalisation d'état</b>	
<b>5. Prise de contact</b>	
5.1 Bien démarrer avec la programmation en liste d'instructions (IL)	5-2
5.2 Bien démarrer avec la programmation en FUPLA	5-5
<b>6. Programmation</b>	
6.1 Programmer en liste d'instructions IL avec les blocs de fonctions « FB »	6-2
6.1.1 Constitution de la fourniture (paquet IL)	6-2
6.1.2 Description des blocs de fonctions	6-5
6.2 Programmer en FUPLA avec les boîtes de fonctions « FBoxes »	6-7
6.3 Programmer en GRAFTEC avec les boîtes de fonctions	6-9

	Page
<b>7. Typologie des erreurs et diagnostic</b>	
7.1 Erreurs de définition	7-1
7.2 Erreurs de traitement	7-2
<b>8. Tâches de comptage et de positionnement</b>	
8.1 Schéma synoptique	8-1
8.2 Principe de fonctionnement	8-2
8.3 Configuration des entrées / sorties ValidationC, CCO, A et B ainsi que du mode de comptage	8-3
8.3.1 Configuration de l'entrée ValidationC	8-3
8.3.2 Configuration de la sortie CCO	8-4
8.3.3 Configuration des entrées A et B	8-5
8.3.4 Configuration du mode de comptage	8-6
8.4 Fonctions de comptage programmables	8-8
8.5 Méthode de programmation	8-9
8.6 Exemple d'application n° 1 : Comptage en GRAFTEC	8-13
8.7 Exemple d'application n° 2 : Positionnement avec codeur rotatif incrémental	8-18
8.8 Exemple d'application n° 3 : Mesure et comptage	8-26
<b>9. Tâches de mesure</b>	
9.1 Mesure de fréquence	9-1
9.1.1 Schéma synoptique	9-1
9.1.2 Principe de fonctionnement	9-2
9.1.3 Configuration de la mesure de fréquence	9-3
9.1.4 Méthode de programmation	9-4
9.1.5 Mesure de fréquence <u>et</u> de comptage	9-6
9.2 Mesure de durée de période	9-7
9.2.1 Schéma synoptique	9-7
9.2.2 Principe de fonctionnement	9-8
9.2.3 Configuration de la mesure de durée de période	9-9
9.2.4 Méthode de programmation	9-10

		Page
9.3	Mesure de durée d'impulsion	9-13
9.3.1	Schéma synoptique	9-13
9.3.2	Principe de fonctionnement	9-14
9.3.3	Configuration de la mesure de durée d'impulsion	9-15
9.3.4	Méthode de programmation	9-16

### **Annexe A      Récapitulatif des blocs de fonctions et des commandes PCD2.H110 en langage IL**


INIT	Initialisation du module PCD2.H110	A-1
EXEC	Exécution d'une commande PCD2.H110	A-3
LdCtPres	Chargement de la valeur du compteur de présélection	A-4
LdRegPres	Chargement de la valeur du registre de présélection	A-5
ModMsConf	Configuration du mode de mesure	A-6
LdMsVal	Chargement de la valeur de mesure	A-7
RdCt	Lecture du compteur	A-9
RdMsImp	Lecture de la mesure en nombre d'impulsions	A-10
RdMsUnit	Lecture de la mesure dans l'unité donnée	A-11
StartCt	Départ comptage	A-12
StartMs	Départ mesure	A-13
StopMs	Arrêt mesure	A-14
RdIdent	Lecture de l'identification du module	A-15

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

en préparation

**Notes personnelles :**





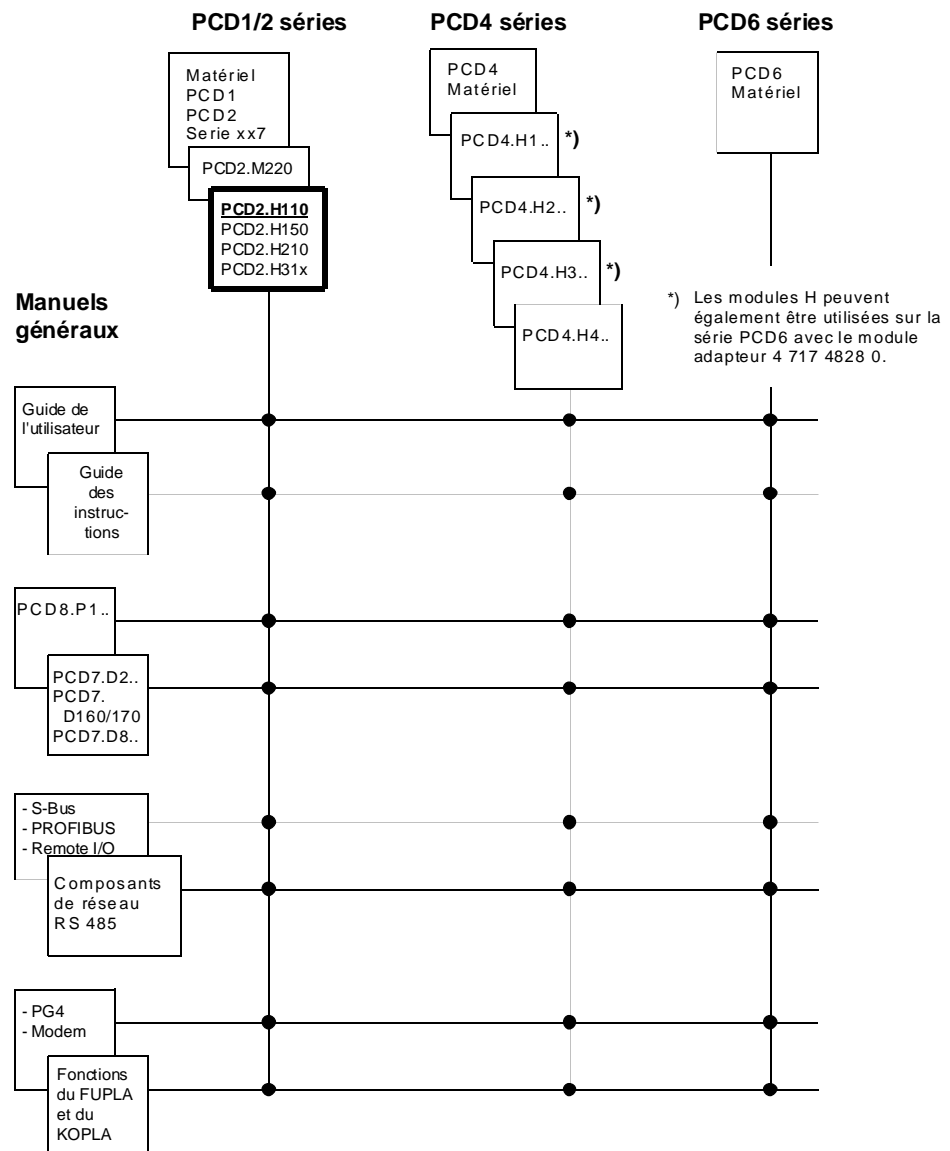
**Avis aux lecteurs :**

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

---

Les automates SAIA® PCD intègrent d'origine 1 600 registres de comptage de 31 bits : il faut toutefois préciser que ceux-ci ne conviennent qu'à des fréquences ne dépassant guère 20 Hz. Grâce aux entrées interruptives, cette limite peut être repoussée à 1 kHz, voire 20 kHz avec le module de comptage rapide PCD2.H100.

Le nouveau module PCD2.H110 hisse la barre encore plus haut : non content de porter la fréquence de comptage à 100 kHz, il autorise également la mesure précise de fréquences jusqu'à 100 kHz ainsi que de durées de périodes ou d'impulsions atteignant 1 heure.

Ses deux entrées de comptage, notées « IN-A » et « IN-B », garantissent la reconnaissance du sens de rotation de codeurs incrémentaux : de quoi ouvrir au H110 la voie de la commande d'axe, réservée néanmoins à des mouvements simples, non régulés : pour une commande optimale de servomoteurs à rampes de démarrage et de freinage, le module de positionnement PCD2.H3.. s'impose.

L'originalité de cette toute dernière innovation SAIA réside notamment dans son circuit programmable FPGA, gage de flexibilité et de performances, qui peut aussi se configurer pour s'adapter aux tâches métiers OEM, par l'intermédiaire d'une mémoire PROM embrochable. Dans cette optique, le module est doté de 4 entrées, de 4 sorties et de 2 x 4 LED assurant l'interfaçage avec le monde extérieur.

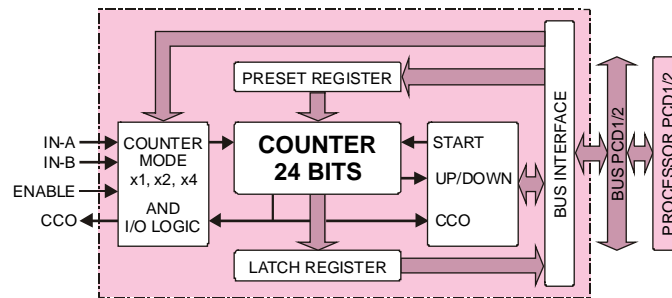
Une bibliothèque de blocs de fonctions et un manuel exhaustif regroupant les fonctions clés du PCD2.H110 sont proposés à l'utilisateur.

## 1.2 Fonctionnement et application

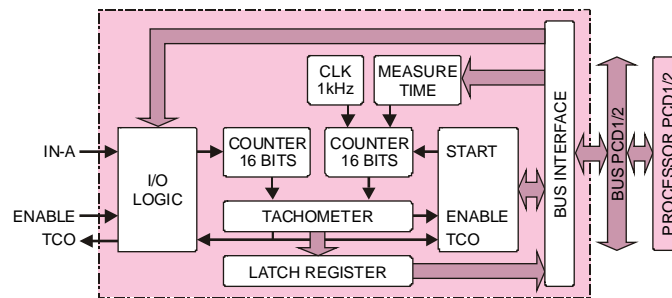
Ce module économique peut s'enficher dans n'importe quel connecteur d'E/S d'un PCD1 ou d'un PCD2.

Il assure plusieurs fonctions :

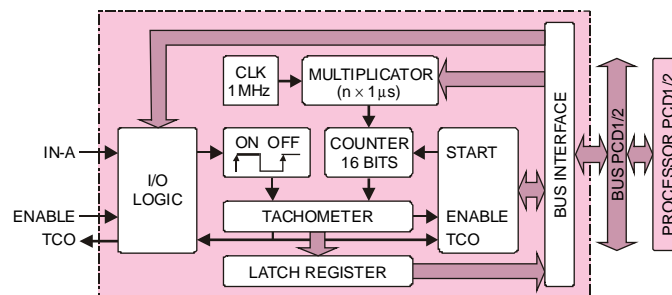
- Schéma synoptique de la fonction Comptage



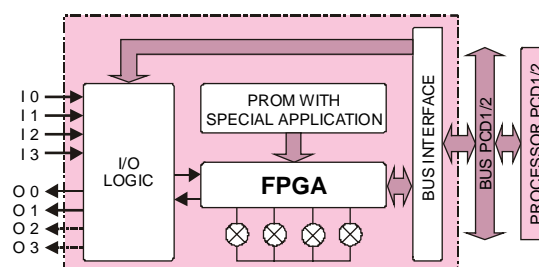
- Schéma synoptique de la fonction Mesure de fréquence



- Schéma synoptique de la fonction Mesure de durée de période ou d'impulsion



- Schéma synoptique de la version OEM



## 1.3 Points forts

---

- Enfichage et exploitation en parallèle de 16 modules PCD2.H110 dans le PCD2 ou 4 dans le PCD1.
- Possibilité de mixer comptage et mesure simultanément sur un même module.

### **En tant que module de Comptage**

- Fréquence de comptage maxi : 100 kHz
- Plage de comptage : 0 à 16 777 215 (24 bits)
- Valeur de présélection : 0 à 16 777 215 (24 bits)
- Comptage / décomptage jusqu'à la valeur de présélection
- 2 entrées TOR (IN-A et IN-B) assurant la reconnaissance du sens de rotation
- 1 sortie processus directe commandée par compteur (CCO)
- Sélection du mode de comptage

### **En tant que module de Mesure de fréquence**

- Plage de fréquence : 500 Hz à 100 kHz
- Plage de mesure : 0 à 65 535 (16 bits)
- Précision  $\geq 1$  %. (selon temps de mesure)
- Possibilité d'utiliser la sortie rapide (TCO) indiquant fin de mesure

### **En tant que module de Mesure de durée de période ou d'impulsion**

- Plage de fréquence : 0.27 mHz à 500 Hz
- Durée de période ou d'impulsion : 2 ms à 1 h
- Possibilité d'utiliser la sortie rapide (TCO) indiquant fin de course

## 1.4 Domaines d'utilisation

---

Les nouveaux modules ..H.. élargissent considérablement le champ d'application des petits automates PCD1 et PCD2. Citons quelques exemples d'applications types du ..H110 :

- Comptage rapide d'impulsions proportionnel à la quantité (articles, unités d'énergie, etc.) sans pénaliser pour autant la performance de l'unité centrale
- Commande d'axe non régulée de tout entraînement doté de codeurs incrémentaux
- Mesure avec la précision du quartz de la vitesse linéaire, de la vitesse de rotation, du débit, etc.

Applications :

- Automatisation de machines (manutention, assemblage...),
- Chargement-déchargement,
- Palettisation,
- Déplacement angulaire automatique (de caméras, phares automobiles, antennes...),
- Pilotage d'axes statiques (définition).

## 1.5 Programmation

---

Les blocs de fonctions « FB » (en liste d'instructions IL) ou boîtes de fonctions « FBox » (en langage FUPLA) du logiciel de programmation graphique PG4 sous Windows permettent de ne saisir que les paramètres indispensables au mode de comptage ou de mesure désiré. Ils font l'objet d'un manuel descriptif complet, illustré d'exemples pratiques.

L'emploi de l'ancien outil de programmation PG3 est aussi possible avec des blocs de fonctions spécifiques.

### Commandes d'initialisation

Bloc INIT	Choix du numéro de module
	Configuration du mode de comptage
	Valeur du compteur de présélection
	Valeur du registre de présélection
	Configuration de l'entrée Validation_Comptage
	Configuration de la sortie CCO
	Configuration de l'entrée IN-A
	Configuration de l'entrée IN-B
	Configuration du mode de mesure
	Valeur de la mesure
	Configuration de l'entrée Validation_Mesure
	Configuration de la sortie TCO

### Commandes d'exécution

Bloc EXEC	Choix du numéro de module
	Commande
	Registre cible (valeur de chargement ou résultat)

Commandes :

- LdCtPres	Chargement de la valeur du compteur de présélection
- LdRegPres	Chargement de la valeur du registre de présélection
- ModMsConf	Configuration du mode de mesure
- LdMsVal	Chargement de la valeur de mesure
- RdCt	Lecture du compteur
- RdMsImp	Lecture de la mesure en nombre d'impulsions
- RdMsUnit	Lecture de la mesure dans l'unité donnée (Hz ou ms)
- StartCt	Départ comptage
- StartMs	Départ mesure
- StopMs	Arrêt mesure
- RdIdent	Lecture de l'identification du module

**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	
Courant d'entrée (typique)	6.5 mA
Type de circuit	Sans séparation galvanique
Filtrage d'entrée	150 kHz

#### Sorties TOR

Nombre de sorties	2
Plage de courant	5 à 500 mA (Courant de fuite maxi : 1 mA) (Résistance de charge mini : 48 $\Omega$ dans la plage de tension 5 à 24 V)
Fréquence	$\leq$ 100 kHz
Plage de tension	5 à 32 V (alimentation externe)
Type de circuit	Sans séparation galvanique, ni protection contre les courts-circuits Commutation du « + »
Chute de potentiel (typique)	< 0.5 V pour 500 mA
Temps de réponse	< 1 $\mu$ s (plus long en cas de charge inductive en raison de la diode de protection)

**Alimentation**

Consommation de courant à partir du bus PCD1/2	5 VCC, 90 mA maxi
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.H110	Module universel de comptage rapide et de mesure
PCD9.H11E	Bibliothèque logicielle (software) avec les blocs de fonctions (FB)

**Voyants de signalisation d'état**

Nombre de voyants lumineux	6
Voyant 0 :	Etat entrée « A »
Voyant 1 :	Etat entrée « B »
Voyant 2 :	Etat entrée « EnableC » (Validation <u>C</u> omptage)
Voyant 3 :	Etat entrée « EnableM » (Validation <u>M</u> esure)
Voyant 4 :	Etat sortie « CCO »
Voyant 5 :	Etat sortie « TCO »

**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	90 mA maxi
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
------------	---

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

<b>Sorties TOR</b>	2 sorties TOR, repérées A0 et A1 (Cf. § 2.1)
--------------------	---

## 2.3 Caractéristiques de fonctionnement

---

Nombre de systèmes	1
Plage de comptage	0 à 16 777 215 (24 bits)
Fréquence de comptage	$\leq 100$ kHz
Protection des données	Toutes les données de ce module sont volatiles (possibilité d'utiliser des registres non volatiles du PCD)

### 3. Constitution

Module équipé

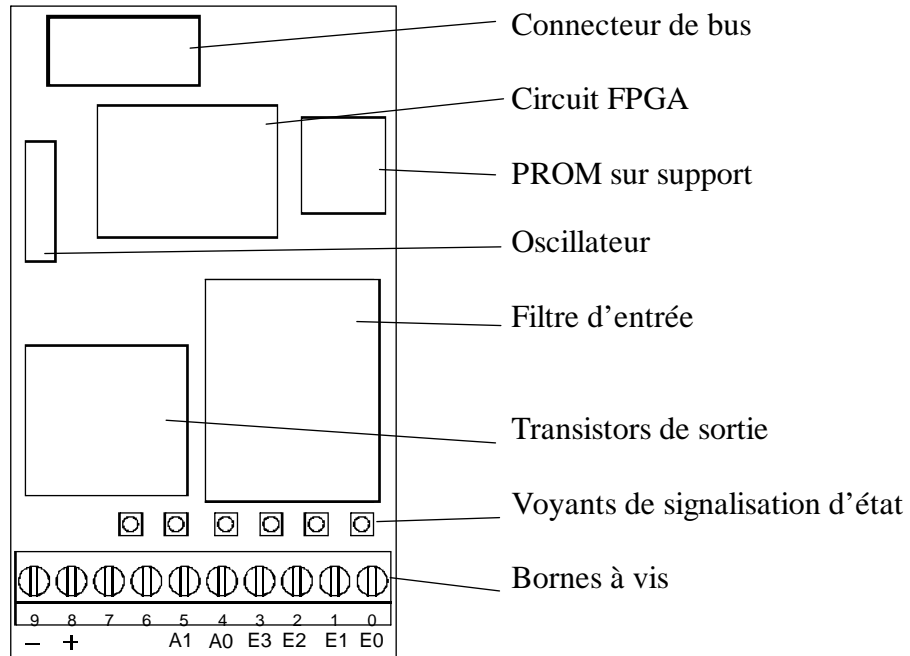
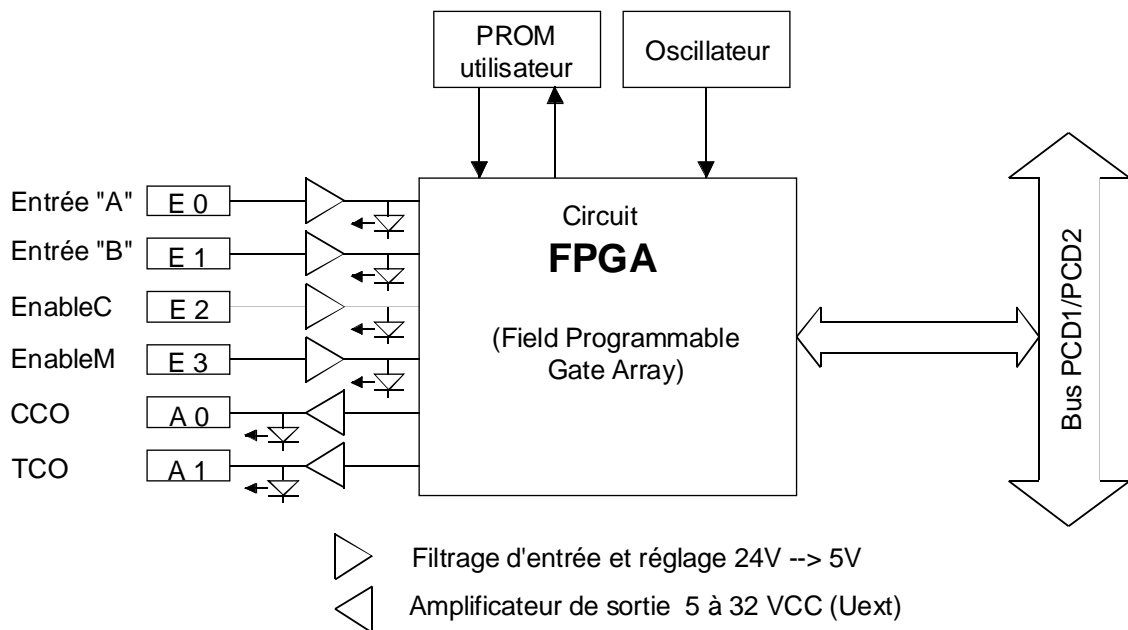


Schéma synoptique

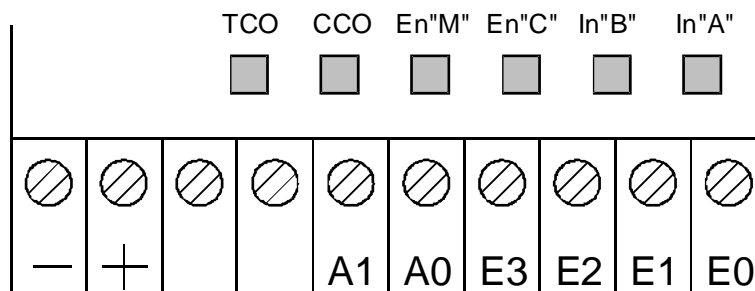


**Notes personnelles :**

## 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).



Bornes n° 9 8 7 6 5 4 3 2 1 0

Voir aussi les schémas synoptiques (Cf. § 1.2)

### Entrées 4

Borne 0 =	E0 :	Entrée « A » de Comptage et Mesure
Borne 1 =	E1 :	Entrée « B » de Comptage uniquement
Borne 2 =	E2 :	Entrée « EnableC » pour utilisation en tant que module de Comptage
Borne 3 =	E3 :	Entrée « EnableM » pour utilisation en tant que module de Mesure

### Sorties 2

Borne 4 =	A0 :	Sortie « CCO » pour la fonction Comptage (CCO = counter controlled output)
Borne 5 =	A1 :	Sortie « TCO » pour la fonction Mesure (TCO = timer controlled output)

### Alimentation

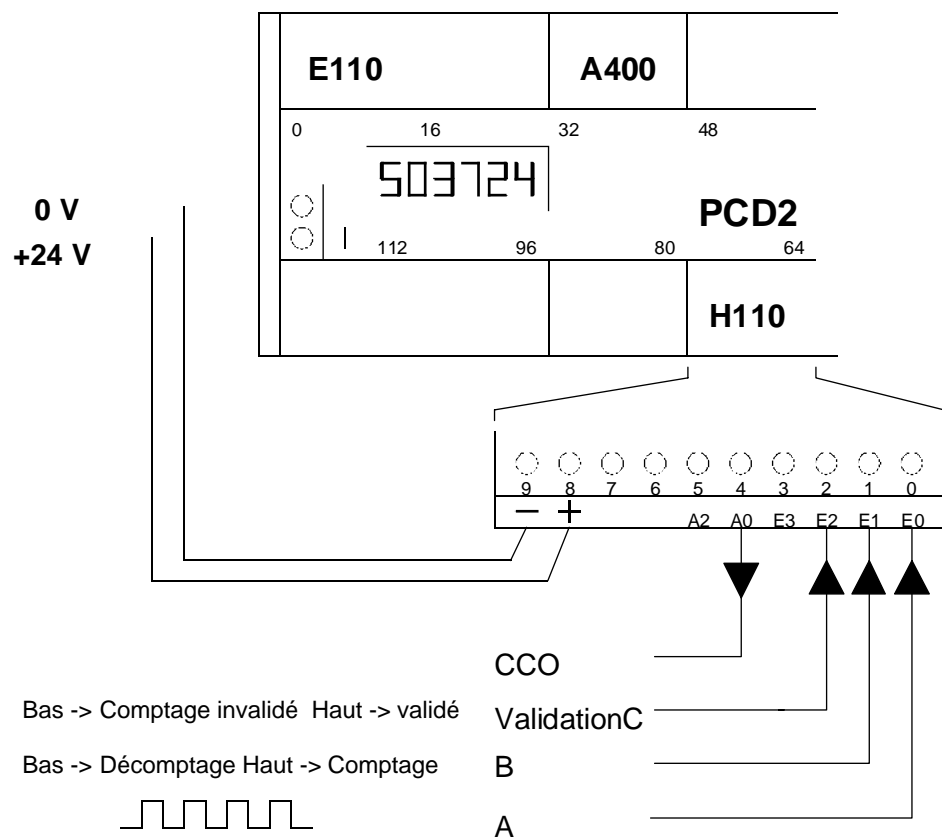
Borne 8 =	+	+ 24 VCC
Borne 9 =	-	Masse (GND)

**Voyants de signalisation d'état**

Voyant 0 =	Etat entrée « A »	"Status of input A"
Voyant 1 =	Etat entrée « B »	"Status of input B"
Voyant 2 =	Etat entrée « ValidationC »	"Status of input EnableC"
Voyant 3 =	Etat entrée « ValidationM »	"Status of input EnableM"
Voyant 4 =	Etat sortie « CCO »	"Status of output CCO"
Voyant 5 =	Etat sortie « TCO »	"Status of output TCO"



## 5. Prise de contact



Configuration minimale requise pour exploiter un PCD2.H110 en mode comptage/décomptage :

- PCD2 (ou PCD1) équipé d'au moins :
  - 1 module PCD2.H110
  - (1 module PCD2.F510/530)
  - (1 module PCD2.E110)
  - (1 module PCD2.A400)
- Contact antirebond
- Alimentation 24 VCC lissée

Un codeur incrémental (24V) peut être raccordé aux bornes d'entrée « A » et « B ». Selon le sens de rotation, les impulsions sont automatiquement comptées ou décomptées.

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

---

Pour mettre en service un PCD2.H110 en mode comptage/décomptage 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.H110. Dans la pratique, il convient de toujours choisir une structure GRAFTEC ou dans le futur FUPLA pour des programmes de ce type.

Exemple de tâche : Après mise sous tension du PCD, le compteur du PCD2.H110 est chargé à 1000. Les signaux de l'entrée « A » doivent alors incrémenter (+) le compteur, et les signaux de l'entrée « B » le décrémenter (-). Il faut pouvoir lire la valeur réelle du compteur sur l'afficheur ou dans le débogueur.

Chaque paramètre du comptage/décomptage ainsi que l'adresse de base du module doivent être configurés conformément aux indications du paragraphe 6.1.

Le programme utilisateur se présente alors comme ci-après :

(Des programmes détaillés et bien structurés figurent dans chaque chapitre ainsi que sur la disquette PCD9.H11E).

Les blocs de fonctions « FB » (IL pour PG4) se trouvent sur la disquette PCD9.H11E. Pour les installer correctement sur votre PC, suivez les instructions du fichier README.TXT de cette disquette.

Le nombre de modules (soit 1) ainsi que l'adresse de base du module PCD2.H110 (64) doivent être indiqués dans le fichier 2D2H110\_B.MBA :

```
NbrModules    EQU    1    ; Nb de modules H110 utilisés (0 à 16)

BA_1          EQU    64   ; Adresse de base du module 1
```

Ce fichier (2D2H110\_B.MBA) doit se trouver dans le répertoire du projet de cet exemple ; il faut donc le copier manuellement de la disquette au répertoire du projet en cours.

```

$include d2h110_b.equ
$group h110

xob      16

ld       r 999      ; registre PCD contenant la
          1000      ; valeur de départ du comptage
ld       r 998      ; registre PCD contenant la
          0         ; valeur du registre de comparaison

cfb      init      ; initialisation du H110
          k 1       ; par 1   n° de module
          0         ; par 2   mode de comptage = x1
          r 999     ; par 3   valeur de départ comptage
          r 998     ; par 4   valeur du reg. de comparaison
          0         ; par 5   ValidationC = statique-normale
          0         ; par 6   CCO = statique-normale
          0         ; par 7   entrée A = normale
          0         ; par 8   entrée B = normale
          0         ; par 9   inutilisé en comptage
          0         ; par 10  inutilisé en comptage
          0         ; par 11  inutilisé en comptage
          0         ; par 12  inutilisé en comptage

cfb      exec      ; départ comptage
          k 1       ; n° de module
          StartCt  ; commande : départ comptage
          r 0       ; registre vide

exob
; -----

cob      0         ; programme utilisateur réel
          0

cfb      exec      ; lecture compteur
          k 1       ; n° de module
          RdCt     ; commande : lecture compteur
          r 777    ; registre 777

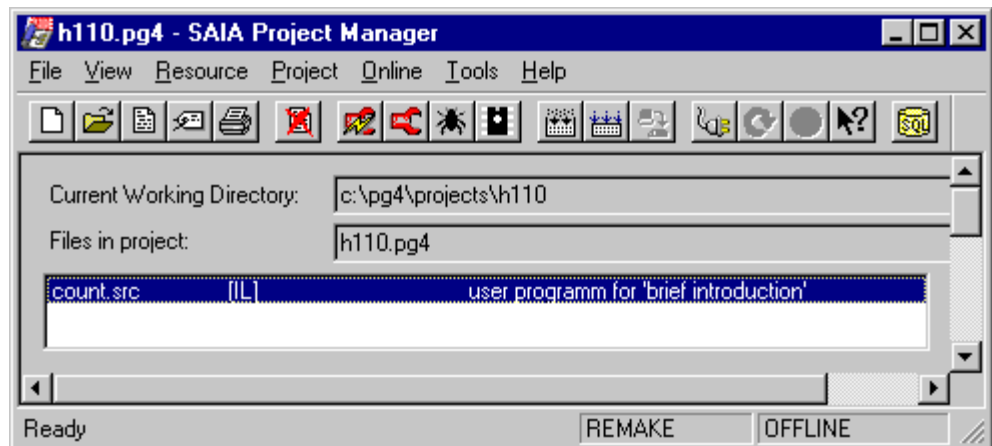
dsp      r 777     ; visualisation sur l'afficheur

$endgroup

ecob

```

Ce programme, baptisé « count.src », doit figuré dans le projet « h110 ».



La commande d'exécution « Project - Build » du Gestionnaire de projet assemble, lie et télécharge le programme dans le PCD, puis lance son exécution (« Run »).

La valeur du compteur apparaît alors immédiatement sur l'afficheur. En l'absence d'afficheur, la valeur du compteur contenue dans le registre R 777 du PCD peut être lue par le débogueur (commande « Display Register 777 <Space> Refresh <CR> »).

## **5.2 Bien démarrer avec la programmation en FUPLA**

---

(En préparation)

**Notes personnelles :**

## 6. 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.H...

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.H11E.

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.

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

---

### 6.1.1 Constitution de la fourniture (paquet IL)

Référence de commande de la disquette : PCD9.H11E.

Cette disquette contient les répertoires suivants :

- APPSDIR toutes les aides « helps »
- FB les fichiers .SRC et .EQU du H110
- FBOX les boîtes de fonctions (FBoxes) du H110
- 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.H11E 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 :

- D2H110\_B.SRC Code source des FB (accessible en lecture seule)
- D2H110\_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
- D2H110\_B.HLP Blocs de fonctions d'aide (help)

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

- D2H110\_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 **D2H110\_b.mba** est représenté ci-dessous.

```

;
; Fichier modifiable par l'utilisateur
;
; Adresses de base définies par l'utilisateur
; -----
$group H110
NbrModules      EQU      1          ; Nb de PCD2.H110 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.H110 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 par « 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.H110 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-H11 » et le module de programme utilisateur, « count-01.sfc ».

Les fichiers sont organisés de la manière suivante :

```
C:\PG4 \FB          \D2H110_b.equ
          \D2H110_b.src
          \...
          \FBOX      \...
          \GALEP3    \...
          \PROJECTS  \FUP_E          (exemple démo PG4)
                   \GRAF_E        (exemple démo PG4)
                   \TEST-H11      \D2H110_b.mba
                                   \count-01.sfc
          \...
          \D2H110_b.hlp
```

Le programme utilisateur de la partie H110 est structuré comme suit :

```
$include D2H110_b.equ
$group H110

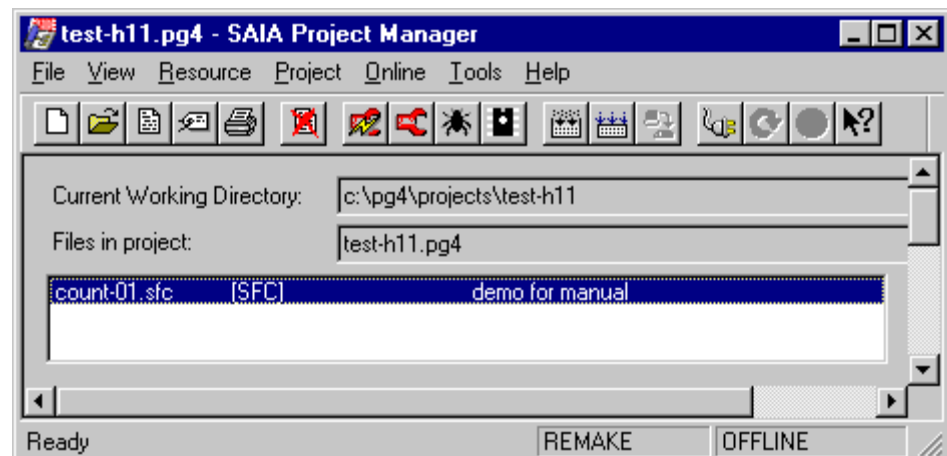
XOB      16

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.



### 6.1.2 Description des blocs de fonctions

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

- INIT            Initialisation            Bloc de 12 paramètres
- EXEC           Exécution                    Bloc de 3 paramètres

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

```
CFB        init
k        1            ; par. 1 : n° de module                    (k 1 à k 16)
          0            ; par. 2 : config. du mode de comptage    (0 à 4)
r        100          ; par. 3 : compteur présélection    (0 à 16777215)
r        101          ; par. 4 : registre de présélection (0 à 16777215)
          0            ; par. 5 : configuration de ValidationC    (0 à 3)
          0            ; par. 6 : configuration de CCO            (0 à 3)
          0            ; par. 7 : configuration de l'entrée A    (0 à 3)
          0            ; par. 8 : configuration de l'entrée B    (0, 1)
          0            ; par. 9 : config. du mode de mesure    (0 à 5)
          0            ; par. 10 : valeur de la mesure        (0 à 65535)
          0            ; par. 11 : configuration de ValidationM (0 à 3)
          0            ; par. 12 : configuration de TCO        (0 à 3)
```

L'appel du bloc de fonctions EXEC revêt généralement la forme suivante :

```
CFB        exec
k        1            ; par. 1 : n° de module                    (k 1 à k 16)
          RdCt        ; par. 2 : commande : lecture compteur
r        555          ; par. 3 : registre cible
```

```
CFB        exec
k        1            ; par. 1 : n° de module                    (k 1 à k 16)
          LdCtPres   ; par. 2 : commande : chargement compteur
r        0            ; par. 3 : inutilisé (registre vide)
```

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

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

Commandes du bloc de fonctions Exec (paramètre 2) :

```

LdCtPres      ; chargement valeur du compteur de présélection
LdRegPres     ; chargement valeur du registre de présélection
ModMsConf     ; configuration du mode de mesure
LdMsVal       ; chargement de la valeur de mesure
RdCt          ; lecture du compteur
RdMsImp       ; lecture de la mesure en nombre d'impulsions
RdMsUnit      ; lecture mesure dans l'unité donnée (Hz ou ms)
StartCt       ; départ comptage
StartMs       ; départ mesure
StopMs        ; arrêt mesure
RdIdent       ; lecture de l'identification du module
    
```

Indicateurs accessibles à l'utilisateur

Indicateur	Adresse	Description
Cstart_x	8 + BA_x	A l'état haut si le comptage est en cours. (l'entrée ValidationC est haute et la commande de départ comptage StartCt est exécutée)
CCO_x	9 + BA_x	Image de la sortie CCO. (invisible dans les modes dynamiques, car trop court)
UpDown_x	10 + BA_x	A l'état haut si le compteur s'incrémente.
EnableM_x	12 + BA_x	Image de l'entrée ValidationM
TCO_x	13 + BA_x	Image de la sortie TCO. (invisible dans les modes dynamiques, car trop court)
OverFlow_x	14 + BA_x	A l'état haut si dépassement dans le mode de mesure.
EndMes_x	15 + BA_x	A l'état haut si la mesure est terminée. Repassse à 0 à la fin d'une commande de lecture RdMsImp ou RdMsUnit.
fPar_Err	Cf. liste des ressources	A l'état haut si un paramètre du bloc INIT est hors plages.
fError_x	Cf. liste des ressources	A l'état haut, si division par zéro dans la conversion d'un résultat de mesure.
rDiag	Cf. liste des ressources	Cf. chapitre 7 « Typologie des erreurs et diagnostic ».

avec « \_x » = numéro de module

Les adresses effectives de ces indicateurs sont indiquées dans la liste des ressources (à des fins de mise au point - pour débogage).

## **6.2 Programmer en FUPLA avec les boîtes de fonctions « FBoxes »**

---

(En préparation)

**Notes personnelles :**

## **6.3 Programmer en GRAFTEC avec les boîtes de fonctions**

(En préparation)

**Notes personnelles :**



## 7. Typologie des erreurs et diagnostic

---

### 7.1 Erreurs de définition

---

Les erreurs de définition du fichier « D2H110\_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 H110 used (NbrModules = 0 in D2H110\_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 H110 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 'H110.RdIdenti'"**

(provoquant l'impression « H110 » à partir de « \$group H110 »)

- Absence de directive « \$group H110 » → Signalisation d'erreur par l'assembleur pour chaque commande et registre/indicateur utilisé dans le programme :

**"Symbol not defined"**

## 7.2 Erreurs de traitement

---

### 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 4 pour la fonction Comptage). Si un paramètre est hors bornes, il est forcé à sa valeur minimale, 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 initiale (IST).

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)
```

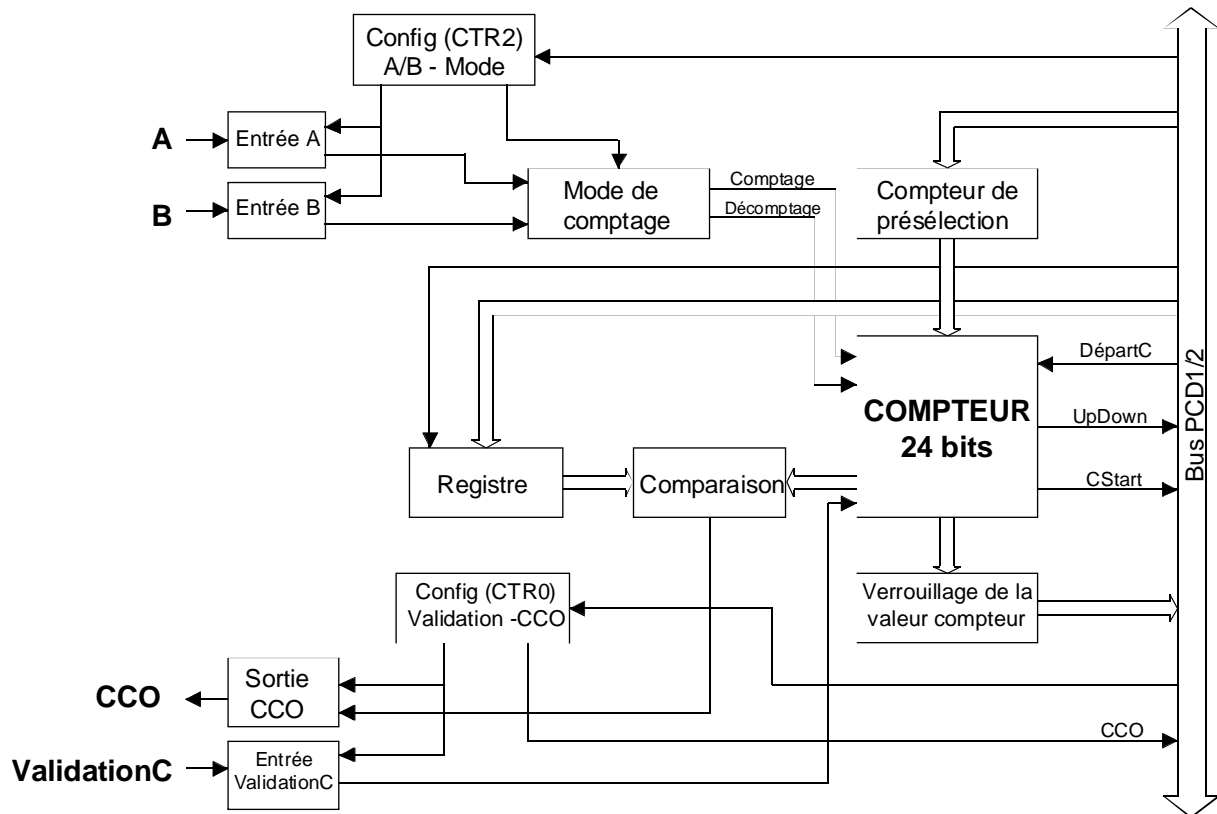
Exemple : Si la configuration de la sortie « CCO » (paramètre 6) du bloc de fonctions INIT du module n° 2 est incorrecte, le registre rDiag prend la valeur 00 01 06 02 hexa.

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. Tâches de comptage et de positionnement

### 8.1 Schéma synoptique



## 8.2 Principe de fonctionnement

---

Le cœur du module est un compteur 24 bits, qui reçoit sa valeur du compteur de présélection, lui-même chargé par le programme utilisateur à partir d'un registre PCD, sous la forme de 3 valeurs de 8 bits. C'est la commande Départ *Comptage* (*StartCt*) qui charge la valeur du compteur de présélection dans le compteur et lance le comptage.

Le chargement du registre suit la même procédure : le programme utilisateur charge la valeur de 24 bits d'un registre PCD dans le registre du H110, sous la forme de 3 valeurs de 8 bits. Après configuration des entrées A et B (Cf. § 8.3.3), de la sortie CCO (Cf. § 8.3.2) et de l'entrée ValidationC (Cf. § 8.3.1), et en cas de concordance du registre et du compteur, CCO peut être activée selon sa configuration pour permettre le pilotage du procédé et du programme utilisateur.

L'état des signaux CCO (*CCO\_x*), ValidationC (*CStart\_x*) et Comptage/décomptage (*UpDown\_x*) peut être lu par le programme utilisateur.

## 8.3 Configuration des entrées / sorties ValidationC, CCO, A et B ainsi que du mode de comptage

### 8.3.1 Configuration de l'entrée ValidationC (« EnableC »)

- Mode standard :

« Statique / normal » paramètre 5 du bloc INIT = 0

- L'entrée à l'état haut autorise le comptage,
- L'entrée à l'état bas arrête le comptage.

- Autres modes :

« Statique / inversé » paramètre 5 du bloc INIT = 1

- L'entrée à l'état bas autorise le comptage,
- L'entrée à l'état haut arrête le comptage.

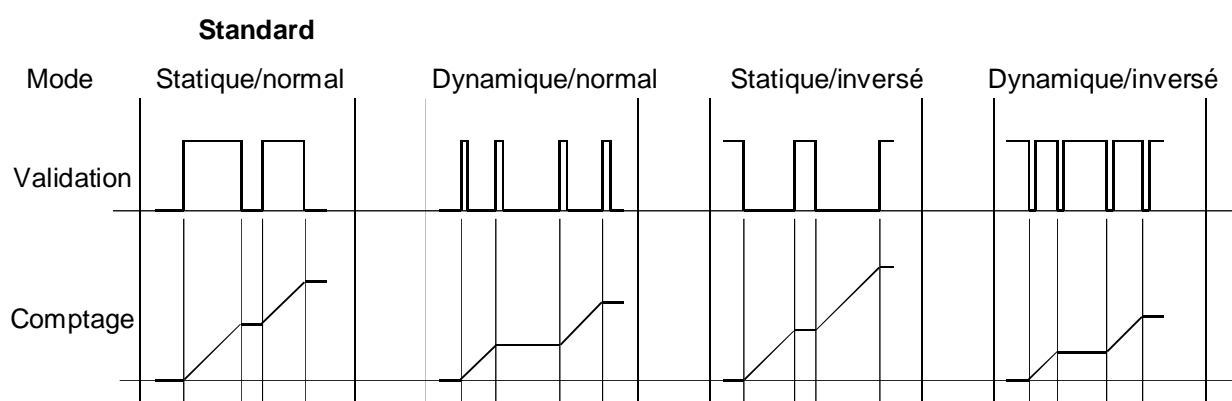
« Dynamique / normal » paramètre 5 du bloc INIT = 2

- L'entrée est à l'état bas : le premier front montant l'active, le second la désactive, et ainsi de suite...

« Dynamique / inversé » paramètre 5 du bloc INIT = 3

- L'entrée est à l'état haut : le premier front descendant l'active, le second la désactive, et ainsi de suite...

Le schéma ci-dessous illustre ces quatre modes de fonctionnement.



### 8.3.2 Configuration de la sortie CCO

« Statique / normal » paramètre 6 du bloc INIT = 0

CCO est activée par le programme utilisateur et passe ou reste à l'état bas. En cas de concordance registre/compteur, elle passe à l'état haut et y reste tant qu'elle n'a pas reçu de nouvelle commande d'activation du programme utilisateur.

« Statique / inversé » paramètre 6 du bloc INIT = 1

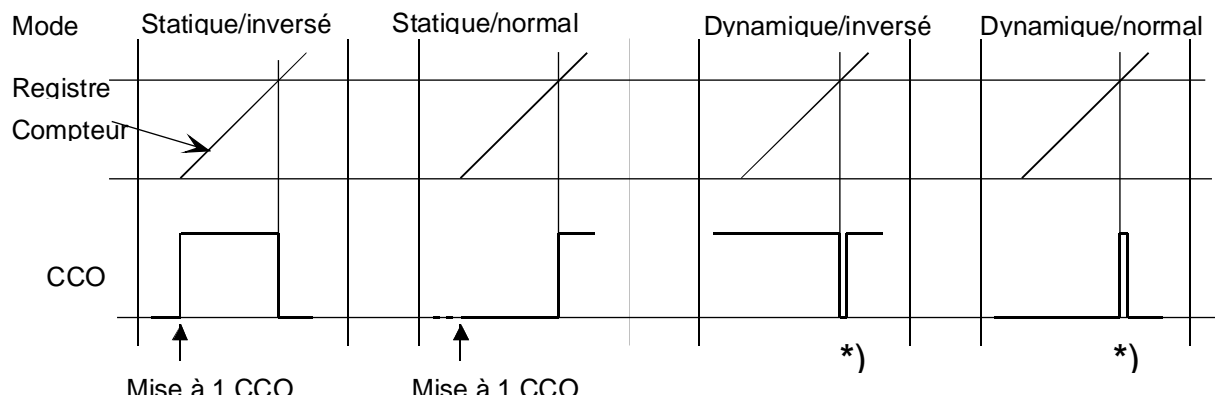
CCO est activée par le programme utilisateur et passe à l'état haut. En cas de concordance du registre et du compteur, elle repasse à l'état bas et y reste tant qu'elle n'a pas reçu de nouvelle commande d'activation du programme utilisateur.

« Dynamique / normal » paramètre 6 du bloc INIT = 2

CCO est activée par le programme utilisateur et passe ou reste à l'état bas. En cas de concordance registre/compteur, elle passe à l'état haut <sup>\*)</sup> pour une durée de 25 à 100  $\mu$ s. Puis à chaque nouvelle concordance registre/compteur, elle répète automatiquement ce processus, sans intervention du programme utilisateur.

« Dynamique / inversé » paramètre 6 du bloc INIT = 3

CCO est activée par le programme utilisateur et passe à l'état haut. En cas de concordance registre/compteur, elle repasse à l'état bas <sup>\*)</sup> pour une durée de 25 à 100  $\mu$ s. Puis à chaque nouvelle concordance registre/compteur, elle répète automatiquement ce processus, sans intervention du programme utilisateur.



\*) Le traitement de cette brève impulsion fait appel aux entrées interruptives du PCD1/PCD2 et au XOB 20 ou XOB 25.

### 8.3.3 Configuration des entrées A et B

Les entrées A et B peuvent être inversées individuellement.

	<b>Comptage</b>	<b>Mesure</b>	<b>Paramètre 7</b>
Entrée A	normal	normale	0
Entrée A	inversé	normale	1
Entrée A	normal	inversée	2
Entrée A	inversé	inversée	3

	<b>Comptage</b>		<b>Paramètre 8</b>
Entrée B	normal	-	0
Entrée B	inversé	-	1

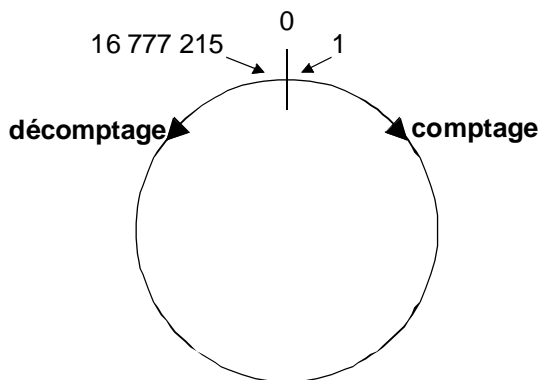
Si un module est utilisé en tant que compteur, l'inversion d'une seule entrée (A ou B) suffit à changer le sens de rotation du moteur.

### 8.3.4 Configuration du mode de comptage

**Mode « x1 »** paramètre 2 du bloc INIT = 0 ou 1

Conçu pour répondre à des applications de comptage simple (sans codeur rotatif incrémental) :

- Les signaux à compter sont appliqués à l'entrée A.
- - L'entrée B à l'état bas se traduit par un comptage décroissant, si le paramètre 8 est égal à 0 ;
- L'entrée B à l'état haut se traduit par un comptage croissant, si le paramètre 8 est égal à 0.
- Si le paramètre 8 est égal à 1, le comptage est inversé.



La plage de comptage s'étend de 0 à 16 777 215 (soit, 0 à  $2^{24} - 1$ )

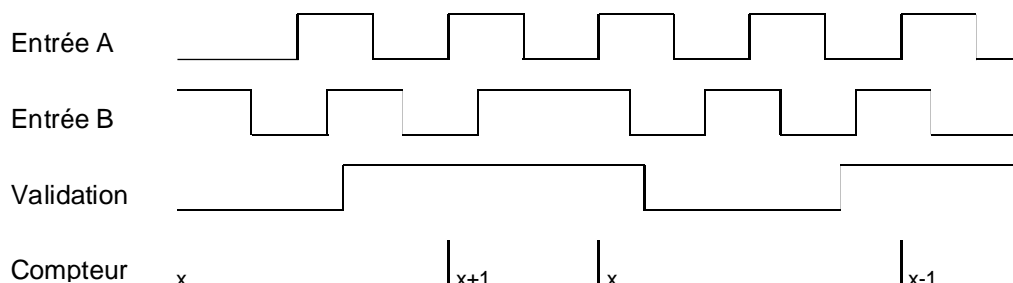
Comptage en partant de 0 → 0, 1, 2...

Décomptage en partant de 0 → 0, 16 777 215, 16 777 214...

Il n'y a ni valeur négative, ni dépassement de capacité.

**Mode « x1 »**

paramètre 2 du bloc INIT = 0 ou 1



Dans ce mode, seul le front montant du signal A est pris en compte. Le signal B, déphasé de 90°, détermine le sens de comptage.

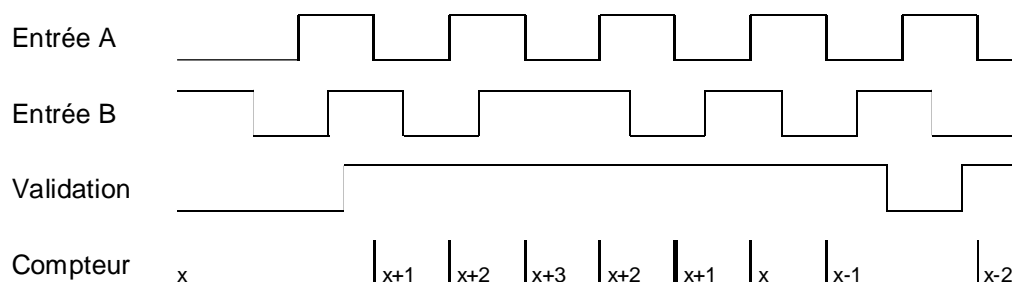
**Important :** En mode x1 les codeurs rotatifs incrémentaux **ne doivent pas** être utilisés parce que dans certaines situations le comptage pourrait être incorrect.



Pour des applications mettant en œuvre des codeurs incrémentaux biphasés, trois modes de comptage sont proposés :

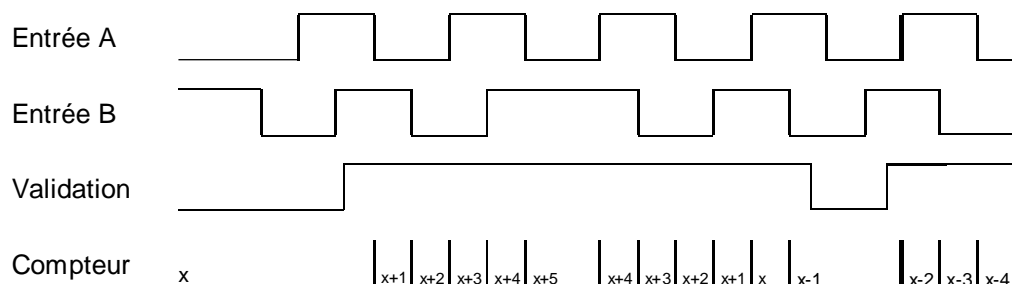
**Mode « x2 »** paramètre 2 du bloc INIT = 2

Dans ce mode, les fronts montant et descendant du signal A sont tous deux pris en compte. C'est encore le signal B, déphasé de 90°, qui dicte le sens de comptage.



**Mode « x4 »** paramètre 2 du bloc INIT = 4

Dans ce mode, les fronts montant et descendant du signal A et du signal B sont tous deux pris en compte. C'est toujours le signal B, déphasé de 90°, qui donne le sens de comptage.



**Mode « x3 »** paramètre 2 du bloc INIT = 3

Il est encore possible d'opter pour un mode x3. Toutefois, celui-ci n'ayant guère d'utilisation pratique, il n'est pas traité dans ce manuel.

## 8.4 Fonctions de comptage programmables

---

- Configuration de l'entrée ValidationC et de la sortie CCO.
  - Configuration du mode de comptage.
  - Configuration des entrées A et B.
  - Initialisation du compteur, c'est-à-dire adoption de la configuration de l'entrée ValidationC et de la sortie CCO, ainsi que du mode de comptage.
  - Chargement des valeurs de compteur dans un registre PCD.
  - Transfert des valeurs de compteur d'un registre PCD dans le compteur de présélection.
  - Transfert de la valeur du compteur de présélection dans le compteur.
  - Chargement de la valeur de présélection dans un registre PCD.
  - Transfert de la valeur de présélection d'un registre PCD dans le registre de présélection.
  - Lancement du comptage et activation de la sortie CCO.
- 
- Lecture de la valeur du compteur.
  - Lecture de l'état de la sortie CCO (Haut = activée, Bas = désactivée).
  - Lecture de l'état de ValidationC (Haut = activée, Bas = désactivée).
  - Lecture du sens de comptage (Haut = comptage, Bas = décomptage).

La programmation est détaillée en pages suivantes, étayée de quelques exemples concrets.

Nous avons choisi d'illustrer la **méthode de programmation** à l'aide d'un exemple non structuré, de préférence à un descriptif de plusieurs pages. Il s'agit d'un programme fonctionnel qui peut, notamment, servir à tester un PCD2.H110.

Le comptage étant toujours un processus séquentiel, constitué d'étapes

- de définition des caractéristiques du module et de la tâche de comptage,
- d'attente de la fin du comptage,
- d'analyse du comptage,

les programmes utilisateurs doivent être en toute logique programmés en GRAFTEC. Il va donc de soi que les trois exemples d'application proposés dans les chapitres suivants sont édités en GRAFTEC.

La programmation s'est effectuée sous progiciel PG4 (GRAFTEC et langage d'instructions IL).

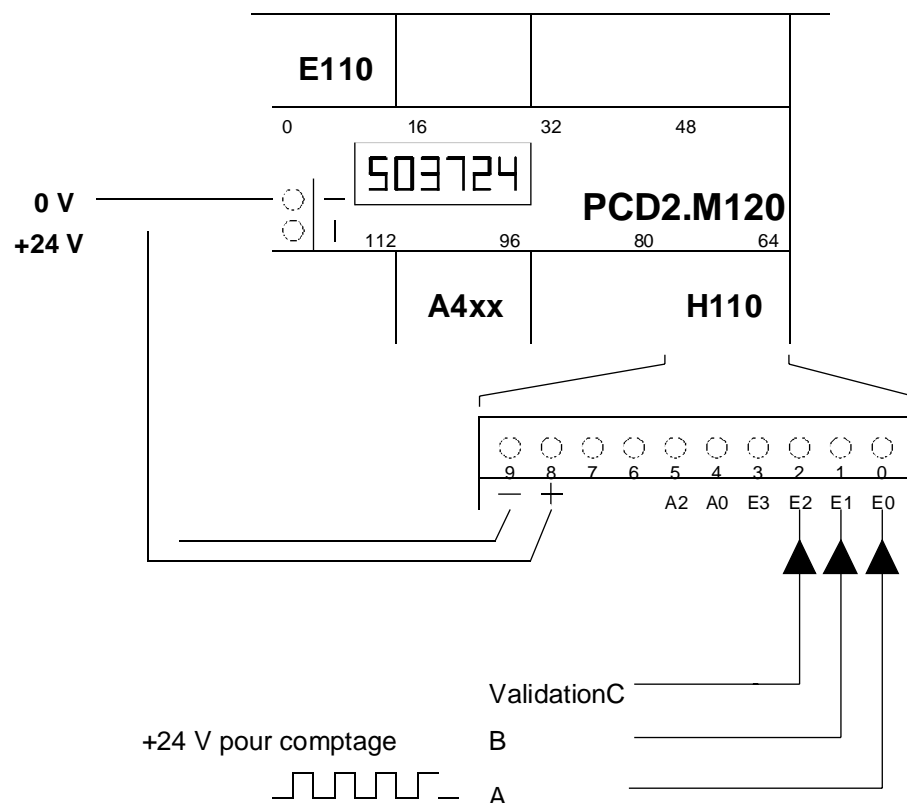
## 8.5 Méthode de programmation

Un exemple simple illustre la méthode de programmation du compteur du module PCD2.H110.

Tâche : Le compteur est chargé à 500 (valeur de départ comptage), et le registre de présélection à 900. La sortie CCO est paramétrée en « statique-normale » et l'entrée ValidationC en « statique-inversée » ; le mode de comptage est configuré en « x1 », les entrées A et B en « normales ». Après avoir appliqué des impulsions sur l'entrée A, le compteur doit s'incrémenter jusqu'à atteindre la valeur du registre de présélection (soit 900). Puis une sortie TOR du PCD est inversée, le compteur est rechargé à 500, et ainsi de suite. Le circuit peut s'apparenter à un diviseur de fréquence.

Le programme, intitulé « prinzip.src », est dans le projet « D2-H110 ». Il faut aussi copier le fichier .MBA dans ce projet et indiquer le nombre de modules H110 mis en œuvre (soit 1) et leur adresse de base (64).

Le PCD2.H110 utilisé en mode Comptage se présente comme suit :



Configuration requise :

PCD1 ou PCD2 équipé d'au moins

1 module PCD2.H110  
1 module PCD2.A4x0  
1 module PCD2.F510/530

```

; *****
; * Programme utilisateur PCD2.H110 destiné à la      *
; * fonction Comptage : prinzip.src                  *
; *****

$include d2h110_b.equ
$group h110

xob      16

ld       r 999      ; chargement du registre PCD avec la
           500      ; valeur de départ du comptage
ld       r 998      ; chargement du registre PCD avec la
           900      ; valeur du registre de comparaison
ld       r 0        ; registre de service
           0        ; vide

cfb      init      ; initialisation du H110
k 1      ; par. 1   n° de module
1        ; par. 2   mode de comptage = x1
r 999    ; par. 3   valeur départ comptage
r 998    ; par. 4   valeur reg. de comparaison
1        ; par. 5   ValidationC=statique-inversée
0        ; par. 6   CCO = statique-normale
0        ; par. 7   entrée A = normale
0        ; par. 8   entrée B = normale
0        ; par. 9   inutilisé en comptage
0        ; par. 10  inutilisé en comptage
0        ; par. 11  inutilisé en comptage
0        ; par. 12  inutilisé en comptage

cfb      exec      ; départ comptage, mise à 1 de CCO
k 1      ; n° de module
StartCt  ; commande : départ comptage
r 0      ; registre vide

exob
; -----

cob      0          ; programme utilisateur réel
           0

cfb      exec      ; lecture compteur
k 1      ; n° de module
RdCt     ; commande : lecture compteur
r 777    ; lecture de la valeur dans R 777

dsp      r 777     ; visualisation sur afficheur

sth      cco_1     ; interrogation de CCO
cpb      h 25      ; si CCO est Haut, appel du PB 25

ecob
; -----

```

```

pb          25          ; PB de relance du compteur

com         o 100       ; inversion à chaque égalité R = C

cfb         exec       ; charg. compteur avec valeur présélection
           k 1          ; n° de module
           LdCtPres    ; commande : charg. valeur présélection
           r 999       ; contenue dans le registre 999

cfb         exec       ; départ comptage, mise à 1 de CCO
           k 1          ; n° de module
           StartCt    ; commande : départ comptage
           r 0         ; registre vide

epb

$endgroup
; -----

```

### Description du programme

Le fichier « d2h110\_b.equ » doit s'insérer en début de programme utilisateur. Le programme situé entre les directives « \$group » et « \$endgroup » constitue le code programme du PCD2.H110.

Dans le bloc de démarrage à froid XOB 16, un registre du PCD, par exemple R 999, est chargé avec la valeur de départ du comptage. Si celle-ci vaut 0, le registre est chargé à 0. Le registre PCD suivant, par exemple R 998, est chargé avec la valeur de comparaison du H110. Ce registre doit être défini même si sa valeur n'est pas utilisée dans le programme. On peut aussi prévoir ici un registre de service vide, par exemple R 0.

La configuration réelle du PCD2.H110 a lieu sur appel du bloc de fonctions INIT, constitué de 12 paramètres (voir l'exemple de programme et l'annexe A de ce manuel pour en connaître la signification). Le paramètre 5 représentant l'entrée ValidationC doit être défini en « statique-inversé » et le paramètre 6 correspondant à la sortie CCO, en « statique-normal ». En cas d'égalité du compteur et du registre de présélection, CCO passe à l'état haut.

Après initialisation du compteur par le bloc de fonctions INIT, le compteur démarre avec le bloc de fonctions EXEC et sa commande de départ comptage « StartCt » (paramètre 2). La sortie CCO est aussi activée.

Dans le COB, la valeur du compteur est lue et transmise au module afficheur ; la sortie CCO est interrogée en permanence. Si elle est à l'état haut (pour signaler la concordance compteur = registre), le bloc de programme PB 25 est appelé, une sortie TOR du PCD est inversée et le compteur est rechargé avec sa valeur de départ, puis relancé.

On peut alors traiter le programme avec la commande « Build » du menu « Project » et le charger dans le PCD pour l'exécuter (« Run »). Il est possible de suivre son déroulement sur l'afficheur et sur la sortie PCD activée.

Il est également envisageable de traiter la sortie CCO en dynamique ; dans ce cas, CCO doit être à l'état haut durant environ 100  $\mu$ s à chaque égalité compteur = registre. Cette impulsion peut être ramenée à l'entrée interruptive INB1. Chaque impulsion compteur = registre déclenche l'appel du bloc d'exception XOB 20, dans lequel le compteur peut être rechargé et relancé.

Précisons toutefois que cette méthode dynamique est affaire de spécialistes, car elle peut s'avérer source de conflits entre les blocs de fonctions H110 utilisés dans le COB et dans le bloc XOB 20.

D'autres applications, plus réalistes, sont présentées dans les exemples qui suivent.

## 8.6 Exemple d'application n° 1 : Comptage en GRAFTEC

Après avoir activé l'entrée de départ « Start », le compteur doit être chargé à 0, et le registre, à la valeur du codeur BCD à deux chiffres relié aux entrées I 16 à I 23.

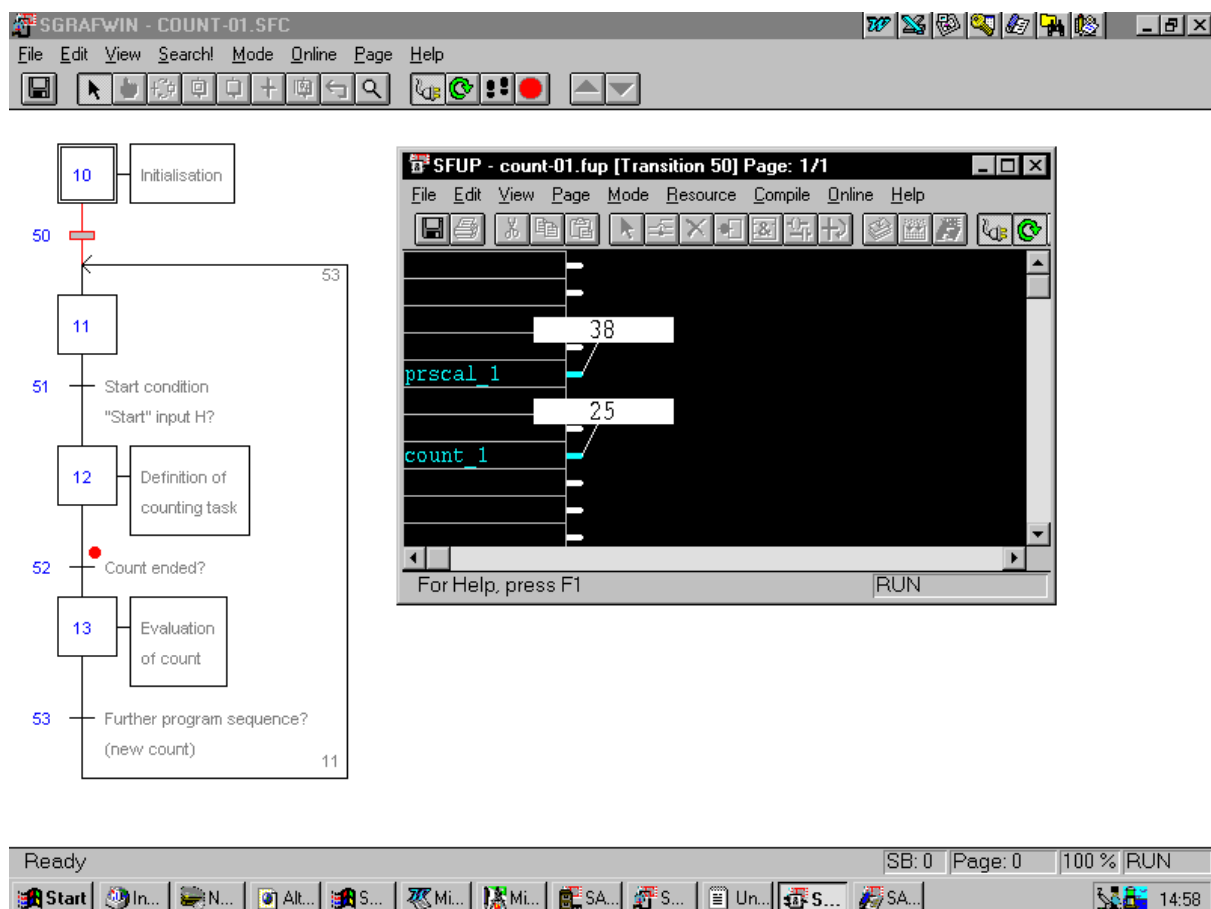
La sortie CCO doit être définie de façon à passer à l'état haut au lancement de chaque nouveau comptage. Lorsque le compteur atteint la valeur du registre de présélection, CCO doit repasser à l'état bas et y rester jusqu'au prochain comptage.

L'état du compteur doit apparaître dans la fenêtre d'affichage, en chaque point du programme. Il faut aussi pouvoir suivre le déroulement du programme en ligne, ainsi que visualiser les valeurs de registre et de compteur, sur l'écran de la console de programmation.

Le programme utilisateur est édité en GRAFTEC.

Dans le projet D2-H110, le programme s'intitule count-01.sfc.

Le programme achevé se présente comme suit :



**Code programme de count-01.sfc**

(Pour obtenir cette représentation, le fichier « count-01.sfc » doit être renommé « count-01.src »).

```

SB          0

;-----
IST          10          ; initialisation
          O 50
$include d2h110_b.equ
$group h110

ld          r 999          ; valeur de départ du comptage
          0
ld          r 998          ; valeur du registre de comparaison
          0
ld          r 0           ; registre de service vide
          0

cfb          init          ; initialisation du H110
          k 1              ; par. 1   n° de module
          1                ; par. 2   mode de comptage = x1
          r 999             ; par. 3   valeur de départ comptage
          r 999             ; par. 4   valeur du reg. de présélection
          1                 ; par. 5   ValidationC= statique-inversée
          1                 ; par. 6   CCO = statique-inversée
          0                 ; par. 7   entrée A = normale
          0                 ; par. 8   entrée B = normale
          0                 ; par. 9   inutilisé en comptage
          0                 ; par. 10  inutilisé en comptage
          0                 ; par. 11  inutilisé en comptage
          0                 ; par. 12  inutilisé en comptage

EST          ; 10

;-----
ST          11
          I 50
          I 53              ; poursuite de la tâche ?
          O 51              ; cond. initiale : entrée Départ haute ?

EST          ; 11

```



```

;-----
ST      12      ; définition de la tâche de comptage
      I 51      ; cond. initiale : entrée Départ haute ?
      O 52      ; fin de comptage ?

digir   2      ; lecture de la valeur BCD
      i 16
      r 998

cfb     exec    ; chargement du registre de présélection
      k 1      ; n° de module
      LdRegPres; commande : chargement de la valeur
      r 998    ; du registre de présélection

cfb     exec    ; chargement du compteur
      k 1      ; n° de module
      LdCtPres ; commande : chargement de la valeur
      r 998    ; du compteur de présélection

cfb     exec    ; départ comptage, mise à 1 de CCO
      k 1      ; n° de module
      StartCt  ; commande : départ comptage
      r 0      ; registre vide

EST     ; 12

;-----
ST      13      ; analyse du comptage
      I 52      ; fin de comptage ?
      O 53      ; autre séquence de programme ?

com     o 101   ; traitement

EST     ; 13

;-----
TR      50
      I 10      ; initialisation
      O 11

;; SFUP
__TR00050

ETR     ; 50

;-----
TR      51      ; cond. initiale : entrée Départ haute ?
      I 11
      O 12      ; définition de la tâche de comptage

cfb     exec    ; lecture du compteur
      k 1      ; n° de module
      RdCt     ; commande : lecture compteur
      r 777    ; lecture de la valeur dans R 777
dsp     r 777   ; visualisation sur afficheur
sth     i 0     ; entrée Départ du PCD

ETR     ; 51

```

```

;-----
TR          52          ; fin de comptage ?
  I 12          ; définition de la tâche de comptage
  O 13          ; analyse du comptage

cfb         exec       ; lecture du compteur
           k 1         ; n° de module
           RdCt       ; commande : lecture compteur
           r 777      ; lecture de la valeur dans R 777
dsp        r 777      ; visualisation sur afficheur
stl        cco_1     ; interrogation de CCO

          ETR          ; 52

;-----
TR          53          ; autre séquence de programme ?
  I 13          ; analyse du comptage
  O 11

cfb         exec       ; lecture du compteur
           k 1         ; n° de module
           RdCt       ; commande : lecture compteur
           r 777      ; lecture de la valeur dans R 777
dsp        r 777      ; visualisation sur afficheur
stl        i 0        ; entrée Départ du PCD

$endgroup

ETR          ; 53

ESB          ; 0

```

**Remarques sur l'exemple n° 1**

- La connaissance du PG4 en général et la maîtrise du GRAFTEC en particulier sont indispensables.
- Un appel du bloc séquentiel SB 0 à partir d'un COB est automatiquement généré.
- Le déroulement du GRAFTEC est visualisable en ligne.
- L'initialisation du H110 a lieu en IST 10. Cette étape initiale n'est traitée qu'au premier appel du SB, comme c'est le cas avec XOB 16. Il est logique d'effectuer l'initialisation du H110 dans l'IST appartenant au SB qui gère ce module, de façon que l'ensemble du programme destiné au module figure dans un seul endroit. On préférera le XOB 16 pour toute initialisation s'appliquant au PCD complet.
- La transition TR 50 est vide, mais elle peut être éditée en FUPLA pour visualiser en ligne les valeurs du compteur et la valeur de présélection BCD. Le code IL de FUPLA est automatiquement généré par PG4 et ne doit en aucune façon être modifié.
- Dans l'étape ST 12 la valeur BCD actuelle est chargée dans le registre de présélection, le compteur est remis à zéro, puis avec la même commande, le compteur est démarré et la sortie CCO activée.
- La transition TR 52 vérifie la fin du comptage de manière à pouvoir, le cas échéant, lancer d'autres séquences de programme. Le procédé lui-même est directement géré au niveau matériel par la sortie CCO. Avant d'interroger CCO (stl cco\_1), le contenu du compteur est lu et affiché. Il en va de même pour les transitions TR 51 et TR 53.

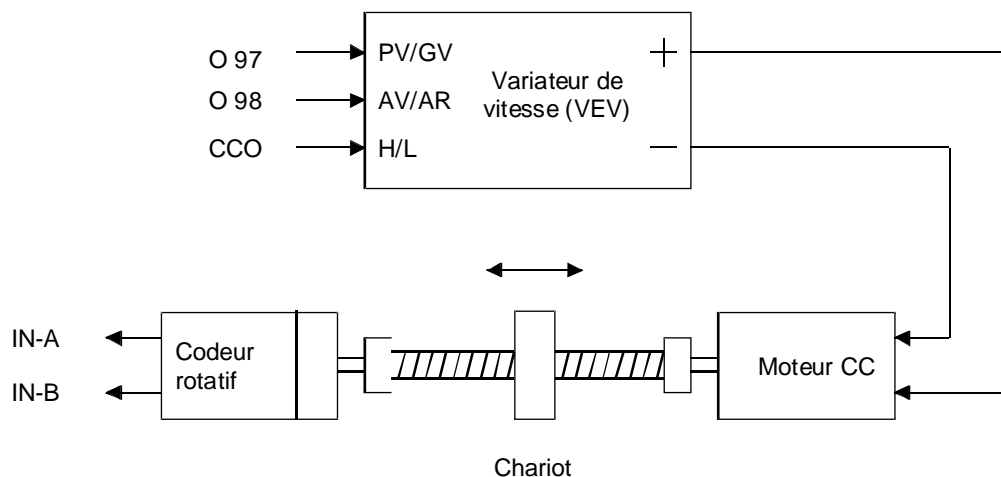
## 8.7 Exemple d'application n° 2 : Positionnement avec codeur rotatif incrémental

Le chariot d'un poste de travail (comprenant moteur à courant continu, axe, mobile, codeur rotatif incrémental et variation électronique de vitesse, VEV) doit se déplacer d'une position origine à la position désirée, marquer une pause, puis revenir à sa position de départ. Il doit effectuer une accélération maximale pour atteindre la vitesse de consigne, puis parvenir à vitesse réduite sur la cible.

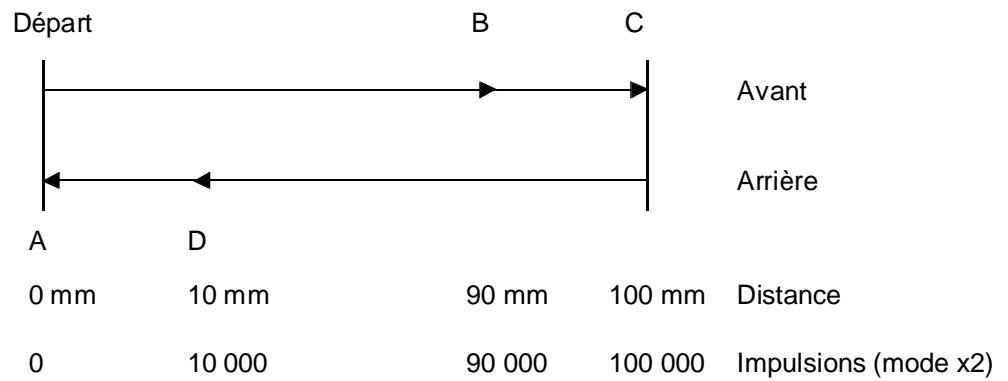
Caractéristiques générales du poste de travail (V-PCX 10) :

Vitesse moteur CC avec réducteur	≈ 1200 tours/min sous 24 VCC
Codeur rotatif incrémental	500 points/tour, biphasé, en quadrature
Pas de la vis	1.0 mm
Entrées VEV (V-PCX 11)	Signal avant/arrière (AV/AR) : Etat haut = avant ; Etat bas = arrière.  Signal petite/grande vitesse (PV/GV) : Etat bas = petite vitesse ; Etat haut = grande vitesse.  Signal marche/arrêt (H/L) : Etat bas « L » = moteur à l'arrêt, court-circuité Etat haut « H » = moteur en marche, selon : avant/arrière, petite/grande vitesse
Sorties VEV	Moteur (vérifier la polarité)
Alimentation VEV	24 VCC lissée

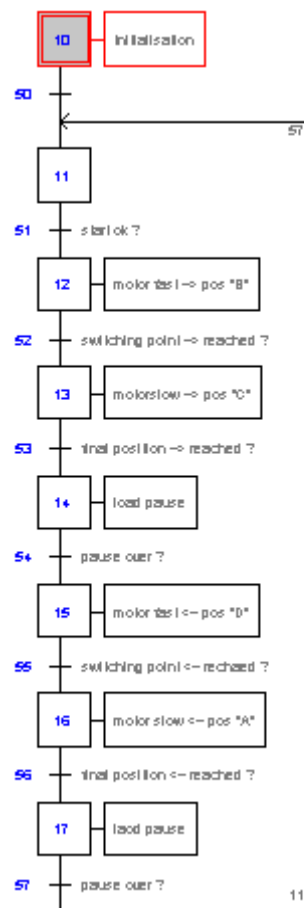
Implantation et câblage des constituants :



Déplacement :



Le GRAFTEC se présente comme suit :



**Code programme de move-01.sfc**

(Pour obtenir cette représentation, le fichier « move-01.sfc » doit être renommé « move-01.src »).

```

SB          0

;-----
IST        10          ; initialisation
          O 50
$include d2h110_b.equ
$group h110

ld         r 999          ; valeur de départ du comptage
          0
ld         r 998          ; valeur du registre de présélection
          0
ld         r 0           ; registre de service vide
          0
ld         r 995          ; reg. de service des valeurs négatives
          16000000
ld         r 996          ; reg. de service de valeur négative
          777215

cfb        init          ; initialisation du H110
          k 1            ; par. 1   n° de module
          2            ; par. 2   mode de comptage = x2
          r 999          ; par. 3   valeur de départ comptage
          r 998          ; par. 4   valeur du reg. de présélection
          1            ; par. 5   ValidationC= statique-inversée
          1            ; par. 6   CCO = statique-inversée
          0            ; par. 7   entrée A = normale
          0            ; par. 8   entrée B = normale
          0            ; par. 9   inutilisé en comptage
          0            ; par. 10  inutilisé en comptage
          0            ; par. 11  inutilisé en comptage
          0            ; par. 12  inutilisé en comptage

EST                    ; 10

;-----
ST         11
          I 50
          I 57          ; pause terminée ?
          O 51          ; départ OK ?
EST                    ; 11

;-----
ST         12          ; moteur GV --> position B
          I 51          ; départ OK ?
          O 52          ; point de commutation --> atteint ?

ld         r 998          ; chargement de la valeur de B
          8640
cfb        exec          ; chargement du registre de présélection
          k 1            ; n° de module
          LdRegPres; commande : chargement reg. présélection
          r 998          ; registre PCD avec valeur de comparaison
cfb        exec          ; départ comptage, mise à 1 de CCO
          k 1            ; n° de module
          StartCt      ; commande : départ comptage
          r 0           ; registre vide

set        o 97          ; moteur GV
set        o 98          ; moteur AV

EST                    ; 12

```

```

;-----
ST      13      ; moteur PV --> position C
      I 52      ; point de commutation --> atteint ?
      O 53      ; cible --> atteinte ?

ld      r 998   ; chargement de la valeur de C
      9600

cfb     exec    ; chargement du registre de présélection
      k 1      ; n° de module
      LdRegPres; commande : chargement reg. présélection
      r 998    ; registre PCD avec valeur de comparaison

cfb     exec    ; départ comptage, mise à 1 de CCO
      k 1      ; n° de module
      StartCt  ; commande : départ comptage
      r 0      ; registre vide

res     o 97    ; moteur PV
set     o 98    ; moteur AV

EST     ; 13

;-----
ST      14      ; chargement pause
      I 53      ; cible --> atteinte ?
      O 54      ; pause terminée ?

ld      t 0     ; pause
      50       ; 5 secondes

EST     ; 14

;-----
ST      15      ; moteur GV <-- position D
      I 54      ; pause terminée ?
      O 55      ; point de commutation <-- atteint ?

ld      r 998   ; chargement de la valeur de C
      960

cfb     exec    ; chargement du registre de présélection
      k 1      ; n° de module
      LdRegPres; commande : chargement reg. présélection
      r 998    ; registre PCD avec valeur de comparaison

cfb     exec    ; départ comptage, mise à 1 de CCO
      k 1      ; n° de module
      StartCt  ; commande : départ comptage
      r 0      ; registre vide

set     o 97    ; moteur GV
res     o 98    ; moteur AR

EST     ; 15

```

```

;-----
ST      16      ; moteur PV <-- position A
      I 55      ; point de commutation <-- atteint ?
      O 56      ; cible <-- atteinte ?

ld      r 998   ; chargement de la valeur de A
      0

cfb     exec    ; chargement du registre de présélection
      k 1      ; n° de module
      LdRegPres; commande : chargement reg. présélection
      r 998   ; registre PCD avec valeur de comparaison

cfb     exec    ; départ comptage, mise à 1 de CCO
      k 1      ; n° de module
      StartCt ; commande : départ comptage
      r 0      ; registre vide

res     o 97    ; moteur PV
res     o 98    ; moteur AR

EST     ; 16

;-----
ST      17      ; chargement pause
      I 56      ; cible <-- atteinte ?
      O 57      ; pause terminée ?

ld      t 0     ; pause
      50      ; 5 secondes

EST     ; 17

;-----
TR      50      ;
      I 10      ; initialisation
      O 11      ;
ETR     ; 50

;-----
TR      51      ; départ OK ?
      I 11      ;
      O 12      ; moteur GV --> position B

cfb     exec    ; lecture du compteur
      k 1      ; n° de module
      RdCt     ; commande : lecture compteur
      r 777   ; lecture de la valeur dans R 777

cmp     r 777   ; | test passage compteur
      r 995   ; |
jr      n next ; |          sous zéro
sub     r 777   ; |
      r 995   ; | si passage sous zéro,
      r 777   ; |
sub     r 777   ; |          affichage
      r 996   ; |
      r 777   ; | de la valeur négative

next:   dsp     r 777 ; visualisation sur afficheur

sth     i 0     ; entrée Départ du PCD

ETR     ; 51

```



```

;-----
TR          52          ; point de commutation --> atteint ?
  I 12          ; moteur GV --> position B
  O 13          ; moteur PV --> position C

cfb         exec       ; lecture du compteur
          k 1          ; n° de module
          RdCt        ; commande : lecture compteur
          r 777        ; lecture de la valeur dans R 777

cmp         r 777      ; | test passage compteur
          r 995        ; |
jr         n next     ; |         sous zéro
sub        r 777      ; |
          r 995        ; |         si passage sous zéro,
          r 777        ; |
sub        r 777      ; |         affichage
          r 996        ; |
          r 777        ; |         de la valeur négative

next:      dsp         r 777      ; visualisation sur afficheur

stl         cco_1      ; interrogation de CCO

ETR         ; 52

;-----
TR          53          ; cible --> atteinte ?
  I 13          ; moteur PV --> position C
  O 14          ; chargement pause

cfb         exec       ; lecture du compteur
          k 1          ; n° de module
          RdCt        ; commande : lecture compteur
          r 777        ; lecture de la valeur dans R 777

dsp         r 777      ; visualisation sur afficheur

stl         cco_1      ; interrogation de CCO

ETR         ; 53

;-----
TR          54          ; pause terminée ?
  I 14          ; chargement pause
  O 15          ; moteur GV <-- position D

cfb         exec       ; lecture du compteur
          k 1          ; n° de module
          RdCt        ; commande : lecture compteur
          r 777        ; lecture de la valeur dans R 777

dsp         r 777      ; visualisation sur afficheur

stl         t 0        ; interrogation du temporisateur

ETR         ; 54

```

```

;-----
TR          55          ; point de commutation <-- atteint ?
  I 15          ; moteur GV <-- position D
  O 16          ; moteur PV <-- position A

cfb         exec       ; lecture du compteur
           k 1         ; n° de module
           RdCt       ; commande : lecture compteur
           r 777      ; lecture de la valeur dans R 777

dsp         r 777      ; visualisation sur afficheur
stl         cco_1     ; interrogation de CCO

ETR         ; 55

;-----
TR          56          ; cible <-- atteinte ?
  I 16          ; moteur PV <-- position A
  O 17          ; chargement pause

cfb         exec       ; lecture du compteur
           k 1         ; n° de module
           RdCt       ; commande : lecture compteur
           r 777      ; lecture de la valeur dans R 777

cmp         r 777      ; | test passage compteur
           r 995      ; |
jr          n next     ; |          sous zéro
sub         r 777      ; |
           r 995      ; | si passage sous zéro,
           r 777      ; |
sub         r 777      ; |          affichage
           r 996      ; |
           r 777      ; | de la valeur négative

next:      dsp         r 777      ; visualisation sur afficheur
           stl         cco_1     ; interrogation de CCO

ETR         ; 56

;-----
TR          57          ; pause terminée ?
  I 17          ; chargement pause
  O 11

cfb         exec       ; lecture du compteur
           k 1         ; n° de module
           RdCt       ; commande : lecture compteur
           r 777      ; lecture de la valeur dans R 777

cmp         r 777      ; | test passage compteur
           r 995      ; |
jr          n next     ; |          sous zéro
sub         r 777      ; |
           r 995      ; | si passage sous zéro,
           r 777      ; |
sub         r 777      ; |          affichage
           r 996      ; |
           r 777      ; | de la valeur négative

next:      dsp         r 777      ; visualisation sur afficheur
           stl         t 0        ; interrogation du temporisateur

$endgroup

ETR         ; 57

ESB         ; 0

```

**Remarques sur l'exemple n° 2**

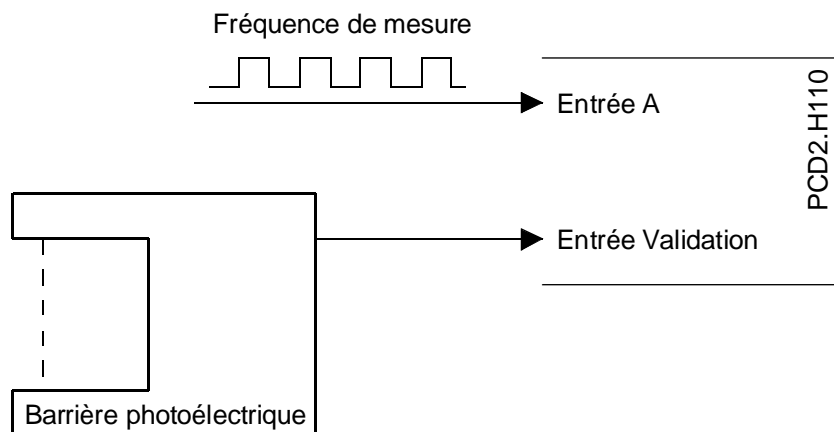
- La connaissance du PG4 en général et la maîtrise du GRAFTEC en particulier sont indispensables.
- Un appel du bloc séquentiel SB 0 à partir d'un COB est automatiquement généré.
- Le déroulement du GRAFTEC est visualisable en ligne.
- L'initialisation du H110 a lieu en IST 10. Cette étape initiale n'est traitée qu'au premier appel du SB, comme c'est le cas avec XOB 16. Il est logique d'effectuer l'initialisation du H110 dans l'IST appartenant au SB qui gère ce module, de façon que l'ensemble du programme destiné au module figure dans un seul endroit. On préférera le XOB 16 pour toute initialisation s'appliquant au PCD complet.
- Au démarrage, c'est-à-dire en IST, le compteur est seulement chargé à 0 ; après quoi, il ne reçoit rien du programme. Tous les signaux issus du codeur incrémental sont alors comptés. Le compteur conserve donc la position exacte du chariot ; même les dépassements de cible ou la rotation manuelle de l'axe sont acquis avec précision.
- L'interrogation logicielle de CCO, aux transitions TR 52, 53, 55 et 56, ne sert qu'à la commutation GRAFTEC. Le contrôle-commande du procédé lui-même, en l'occurrence notre chariot, est directement effectué par la sortie CCO.
- Le sous-programme « Test passage compteur sous zéro », présent dans la plupart des transitions, a pour seule mission de préserver la validité de l'affichage et d'éviter le positionnement de l'indicateur d'erreur, l'afficheur n'ayant que 6 chiffres. En effet, si la valeur du compteur descend au-dessous de zéro, ce sous-programme lui soustrait d'emblée 16 000 000 (R 999) pour permettre son affichage. On peut aussi contourner cette difficulté en définissant, par exemple, la position zéro à 1000.
- Le lecteur ne manquera pas de constater que cet exemple de programmation présente quelques lacunes. En effet, le passage de GV à PV entraîne un bref arrêt ou court-circuit du moteur par le variateur, la sortie CCO étant désactivée sur atteinte de la position B ou D, puis n'étant réactivée qu'après chargement d'une nouvelle valeur. Dans un programme réalisé selon les règles de l'art, cette commutation serait évitée par la mise en dérivation sur une sortie normale. Cela reste un exemple...
- En 9.1.5, nous verrons qu'une fonction mesure de fréquence peut se greffer à la fonction positionnement.

## 8.8 Exemple d'application n° 3 : Mesure et comptage

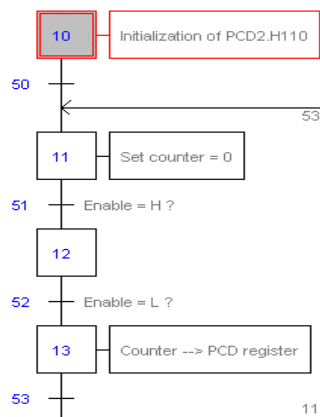
Cette application met en œuvre une barrière photoélectrique dont le faisceau lumineux est interrompu par des articles défilant sur une bande transporteuse ; le comptage des impulsions émises proportionnellement à la vitesse de la bande permet alors de mesurer, à des fins de tri, la taille des éléments convoyés. Cette méthode simple a largement fait ses preuves dans le cadre de l'automatisation d'une ligne de tri de melons et d'abricots, située dans le Midi de la France.

L'objectif est donc de compter les signaux dans une situation donnée (par exemple, l'obstruction du faisceau lumineux), ce comptage étant directement pris en charge par la barrière photoélectrique à l'aide d'une entrée du module de comptage, sans mobiliser une entrée TOR du PCD.

A cette fin, les impulsions de mesure sont appliquées à l'entrée A, et la barrière photoélectrique, à l'entrée Validation du PCD2.H110.



Le GRAFTEC se présente comme suit :



**Code programme de mess-01.sfc**

(Pour obtenir cette représentation, le fichier « mess-01.sfc » doit être renommé « mess-01.src »).

```

SB          0

;-----
IST         10          ; initialisation
          O 50

$include d2h110_b.equ
$group h110

ld         r 999        ; valeur de départ du comptage
          0
ld         r 998        ; valeur du registre de présélection
          0
ld         r 0          ; registre de service vide
          0

cfb        init        ; initialisation du H110
          k 1          ; par. 1  n° de module
          0          ; par. 2  mode de comptage = x1
          r 999        ; par. 3  valeur de départ comptage
          r 998        ; par. 4  valeur du registre (inutilisé)
          1          ; par. 5  ValidationC= statique-inversée
          0          ; par. 6  CCO (inutilisé)
          0          ; par. 7  entrée A = normale
          1          ; par. 8  entrée B = inversée
          0          ; par. 9  inutilisé en comptage
          0          ; par. 10 inutilisé en comptage
          0          ; par. 11 inutilisé en comptage
          0          ; par. 12 inutilisé en comptage

          EST          ; 10

;-----
ST         11          ; compteur à 0
          I 50
          I 53
          O 51          ; validation à l'état haut ?

cfb        exec        ; chargement compteurs (avec 0)
          k 1          ; n° de module
          LdCtPres    ; commande : chargement compteur
          r 998        ; registre PCD avec valeur de chargement

cfb        exec        ; départ comptage
          k 1          ; n° de module
          StartCt    ; commande : départ comptage
          r 0          ; registre vide

          EST          ; 11

;-----
ST         12
          I 51          ; validation à l'état haut ?
          O 52          ; validation à l'état bas ?
          EST          ; 12

```

```

;-----
ST          13          ; compteur --> registre PCD
      I 52          ; validation à l'état bas ?
      O 53

cfb          exec      ; lecture du compteur (pour résultat)
      k 1          ; n° de module
      RdCt        ; commande : lecture compteur
      r 777        ; lecture de la valeur dans R 777

putx        r 777
           r 2000
ini         k 1000

EST                          ; 13

;-----
TR          50
      I 10          ; initialisation
      O 11          ; compteur à 0
ETR         ; 50

;-----
TR          51          ; validation à l'état haut ?
      I 11          ; compteur à 0
      O 12

cfb          exec      ; lecture du compteur (pour affichage)
      k 1          ; n° de module
      RdCt        ; commande : lecture compteur
      r 777        ; lecture de la valeur dans R 777

dsp         r 777      ; visualisation sur afficheur

sth         Cstart_1 ; interrogation de ValidationC

ETR         ; 51

;-----
TR          52          ; validation à l'état bas ?
      I 12
      O 13          ; compteur --> registre PCD

cfb          exec      ; lecture du compteur (affichage)
      k 1          ; n° de module
      RdCt        ; commande : lecture compteur
      r 777        ; lecture de la valeur dans R 777

dsp         r 777      ; visualisation sur afficheur

stl         Cstart_1 ; interrogation de ValidationC

ETR         ; 52

;-----
TR          53
      I 13          ; compteur --> registre PCD
      O 11          ; compteur à 0

$endgroup

ETR         ; 53

ESB         ; 0

```

**Remarques sur l'exemple n° 3**

- La connaissance du PG4 en général et la maîtrise du GRAFTEC en particulier sont indispensables.
- Un appel du bloc séquentiel SB 0 à partir d'un COB est automatiquement généré.
- Le déroulement du GRAFTEC est visualisable en ligne.
- Quelques précisions sur la configuration de l'entrée ValidationC et de la sortie CCO s'imposent :
  - La barrière photoélectrique de cet exemple délivre un signal haut à l'entrée ValidationC en présence du faisceau lumineux ; le signal doit donc être déclaré « statique/inversé ».
  - La sortie CCO, inutilisée, est tout simplement configurée en « dynamique/normale » (soit un code 2).
- Le mode de comptage retenu ici est le « comptage/décomptage », soit x1. Avec le code 0, le +24 V doit être appliqué à l'entrée B pour autoriser le comptage. Le code 1 inverse l'entrée B, qui n'a pas besoin du +24 V pour réaliser le comptage.
- Le programme est géré par interrogation logicielle de l'entrée ValidationC avec « sth Cstart\_1 » ou « stl Cstart\_1 ». Le compteur est directement commandé par l'entrée ValidationC.
- L'étape ST 13 indique le résultat du comptage. Chaque désactivation de la barrière photoélectrique entraîne le stockage de la valeur du compteur dans des registres PCD consécutifs (partant de R 2000).
- On peut aussi bien remplacer la barrière photoélectrique par un contact antirebond.

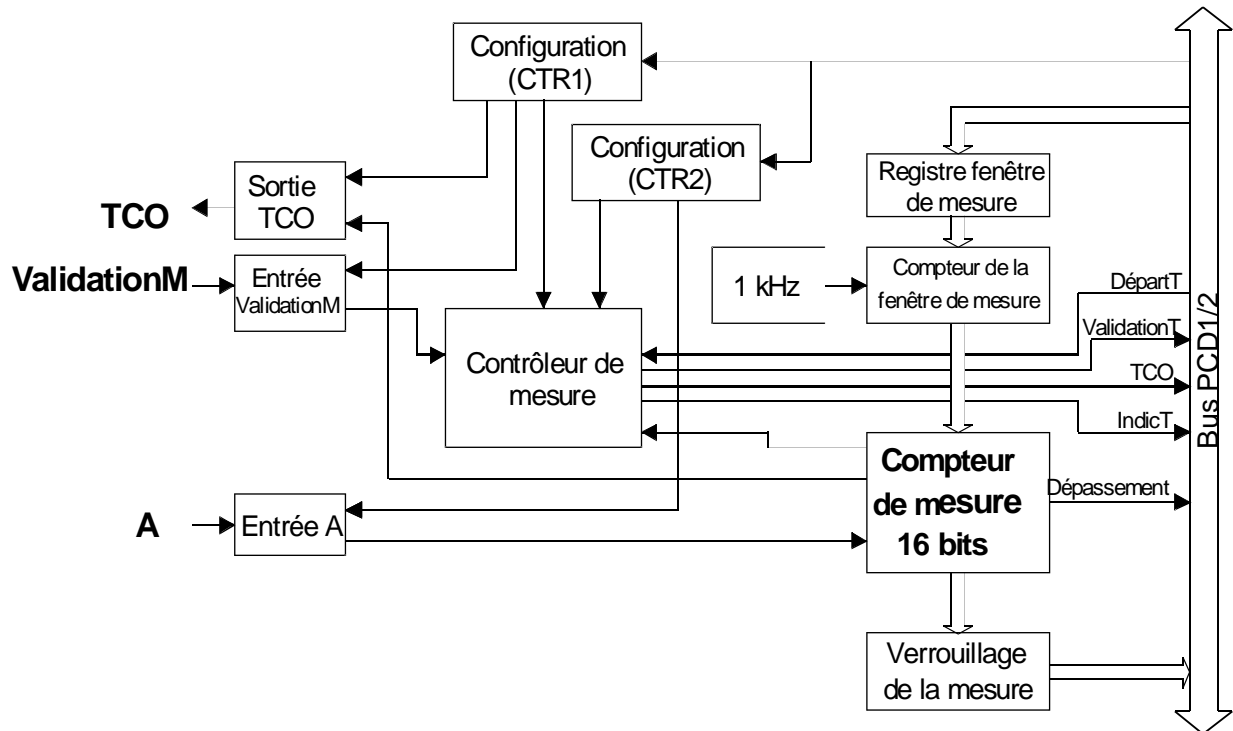
**Notes personnelles :**



## 9. Tâches de mesure

### 9.1 Mesure de fréquence

#### 9.1.1 Schéma synoptique



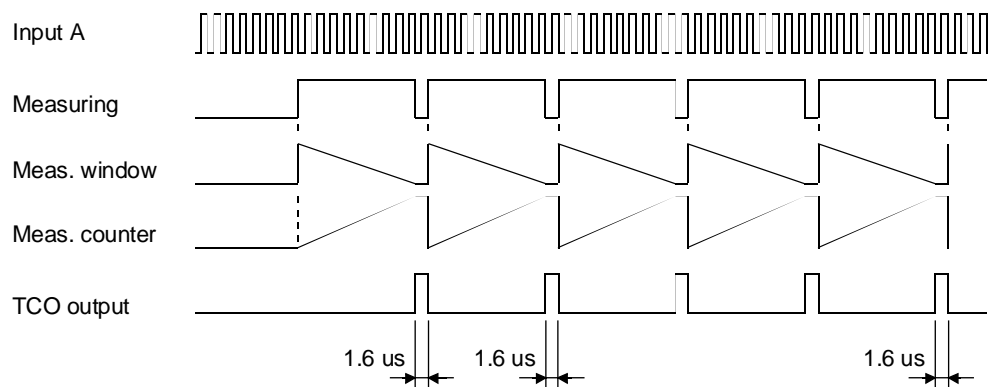
### 9.1.2 Principe de fonctionnement

La plage de mesure de fréquence s'étend de 500 Hz à 100 kHz.

La mesure peut s'effectuer pendant le comptage. Elle utilise 2 compteurs de 16 bits :

- Le premier, « compteur de fenêtre de mesure », permet, grâce à une horloge fixe de 1 kHz, d'ouvrir une fenêtre de mesure par pas de 1 ms.
- Le second, « compteur de mesure », permet de compter les signaux attaquant l'entrée A durant l'ouverture de la fenêtre de mesure. Si la fenêtre est paramétrée à 1 s (soit 1000 ms), le résultat apparaît sur le compteur de mesure directement en Hertz ou en impulsions par seconde.

La mesure de fréquence se déroule selon un processus automatique : la durée de la fenêtre de temps est mesurée, puis la sortie TCO est activée, la valeur mesurée est verrouillée, et une nouvelle mesure est effectuée, et ainsi de suite. L'activation de TCO, le verrouillage de la mesure et la remise à zéro du compteur durent 1,6  $\mu$ s.



### 9.1.3 Configuration de la mesure de fréquence

La configuration fait appel au bloc de fonctions INIT. Pour une mesure en continu, le paramètre 9 doit être à 5 (mode fréquence-automatique).

Une mesure en mode impulsionnel est possible, mais sans grande utilité ; le paramètre 9 doit alors valoir 4 (fréquence-manuel). Dans ce cas, une seule fenêtre de mesure est mesurée, puis l'ensemble de la procédure se répète pour réaliser une nouvelle mesure.

#### Plage de mesure et fenêtre de temps

La plage de mesure s'étend de 0 à 65 535 (16 bits).

Pour obtenir une résolution de 1 ‰, il faut acquérir au moins 1000 signaux par mesure. La durée d'ouverture de la fenêtre de mesure dépend de la fréquence à mesurer : pour 100 kHz, il faut au minimum 10 ms ; pour 500 Hz, il faut obligatoirement au moins 2 s.

Plus la fréquence à mesurer diminue, plus la durée de mesure augmente, ce qui n'est guère acceptable dans beaucoup d'applications. Si l'on veut concilier précision et temps de mesure acceptable, dans le cas de fréquences inférieures à environ 100 Hz, la mesure de durée (Cf. § 9.2) peut s'avérer préférable à la mesure de fréquence.

Le paramètre 11 définit le comportement de l'entrée ValidationM, et le paramètre 12, celui de la sortie TCO.

= 9	Configuration du mode de mesure		Entier	0 à 5	4 = fréquence-manuel 5 = fréquence-automatique
= 10	Chargement de la valeur de mesure		Entier	0 à 65 535	Durée de la fenêtre de mesure (mesure de fréquence)
= 11	Configuration de l'entrée ValidationM		Entier	0 à 3	0 = statique-normal 1 = statique-inversé 2 = dynamique-normal 3 = dynamique-inversé
= 12	Configuration de la sortie TCO		Entier	0 à 3	0 = statique-normal 1 = statique-inversé 2 = dynamique-normal 3 = dynamique-inversé

Ce tableau est reproduit in extenso en page 2 de l'annexe A.

### 9.1.4 Méthode de programmation

Tâche : Un train d'impulsions est appliqué à l'entrée A. Il faut alors effectuer une mesure de fréquence continue (mode automatique) avec une fenêtre de temps de 1 s (1000 ms). Le résultat apparaît en unités de comptage (Hz) sur l'afficheur.

```

; *****
; * Programme utilisateur PCD2.H110 destiné à la *
; * Mesure de fréquence : frequ-01.src          *
; *****
;
#include d2h110_b.equ
$group h110

xob      16

ld       r 0      ; registre de service vide
         0

cfb      init     ; initialisation du H110
         k 1      ; par. 1  n° de module
         0        ; par. 2  inutilisé en mesure
         r 0      ; par. 3  inutilisé en mesure
         r 0      ; par. 4  inutilisé en mesure
         0        ; par. 5  inutilisé en mesure
         0        ; par. 6  inutilisé en mesure
         0        ; par. 7  entrée A = normale
         0        ; par. 8  inutilisé
         5        ; par. 9  mode = fréquence-automatique
         1000     ; par. 10 durée fenêtre de temps en ms
         0        ; par. 11 ValidationM=statistique-normale
         2        ; par. 12 TCO = dynamique-normale

cfb      exec     ; départ mesure
         k 1      ; n° de module
         StartMs ; commande : départ mesure
         r 0      ; registre vide

exob

; -----

cob      0
         0

wait:    sth      EndMes_1 ; mesure terminée ?
         jr       l wait   ; si non, attente

cfb      exec     ; lecture du résultat de la mesure
         k 1      ; n° de module
         RdMsImp ; commande : lecture mesure en impulsions
         r 777   ; lecture de la valeur dans R 777

dsp      r 777   ; visualisation sur afficheur

;(wait:)
ecob
; -----

xob      20      ; interruption INB1
com      o 101   ; inversion après chaque mesure
exob

$endgroup

```

## Description du programme

Au niveau matériel, un transmetteur de signal fournissant les signaux à mesurer (un générateur d'impulsions, par exemple) doit être relié à l'entrée A, l'entrée B restant ouverte.

L'entrée ValidationComptage, inutile à la mesure, reste ouverte.  
L'entrée ValidationMesure devrait certes recevoir le +24 V, mais dans la mesure où elle a été inversée à la configuration, elle reste ouverte <sup>\*)</sup>.

La sortie TCO est reliée à l'entrée interruptive INB1 du PCD. Au terme de chaque mesure, le XOB 20 est appelé : il inverse ici la sortie O 101 du PCD, ce qui permet de mieux visualiser le déroulement du programme. Dans cet exemple, O 101 est inversée toutes les secondes.

La base de temps de cet exemple est fixée à 1000, pour une fenêtre de temps de 1000 ms, soit 1 s (Cf. « Plage de mesure et fenêtre de temps » en 9.1.3).

Le COB attend la fin de chaque mesure avant de lire le résultat du compteur de mesure, visualisable par le débogueur ou sur un afficheur au format entier. L'unité de mesure est dictée par la base de temps préalablement définie (ici, le Hertz).

Si l'on choisit la commande « RdMsUnit » (lecture dans l'unité donnée) au lieu de « RdMsImp » (lecture en nombre d'impulsions), le résultat est toujours converti en Hertz ; il peut être visualisé dans le registre PCD, en virgule flottante.

La commande « StopMs » interrompt une mesure en cours, sans résultat valide. La mesure peut reprendre avec la commande « StartMs ».

\*) Une désactivation de l'entrée ValidationM invalide le résultat de la mesure, qui ne peut être relancée que par la commande StartMs.

### 9.1.5 Mesure de fréquence et comptage

Nous l'avons vu, la mesure de fréquence peut s'effectuer pendant le comptage. Cette possibilité s'illustre parfaitement en combinant l'exemple d'application n° 2 « Positionnement avec codeur rotatif incrémental » (Cf. paragraphe 8.7) et l'explication que nous donnons ici de la mesure de fréquence. Il faut seulement supprimer de l'exemple de positionnement les commandes d'affichage de position (lecture compteur), dans la mesure où l'on ne dispose que d'un afficheur réservé à la visualisation de la fréquence.

Ces deux fonctions étant exécutées sur un même H110, l'initialisation porte sur l'ensemble du module. Dans cet exemple, elle a lieu dans l'étape initiale IST du programme de positionnement et est immédiatement suivie du lancement de la mesure.

```

IST          10

$include d2h110_b.equ
$group h110

ld          r 0          ; registre de service vide
            0

ld          r 999        ; valeur de départ du compteur
            0

ld          r 998        ; valeur du registre de présélection
            0

cfb         init        ; initialisation du H110
k 1         ; par. 1    n° de module
2         ; par. 2    mode de comptage = x2
r 999      ; par. 3    valeur de départ comptage
r 998      ; par. 4    valeur du reg. de présélection
1         ; par. 5    ValidationC= statique-inversée
1         ; par. 6    CCO = statique-inversée
0         ; par. 7    entrée A = normale
0         ; par. 8    entrée B = normale
5         ; par. 9    mode de mesure=fréquence-auto.
100        ; par. 10   durée fenêtre de temps en ms
0         ; par. 11   ValidationM = statique-normale
2         ; par. 12   TCO = dynamique-normale

cfb         h exec      ; départ mesure
k 1         ; n° de module
StartMs    ; commande : départ mesure
r 0         ; registre vide

EST          ; 10

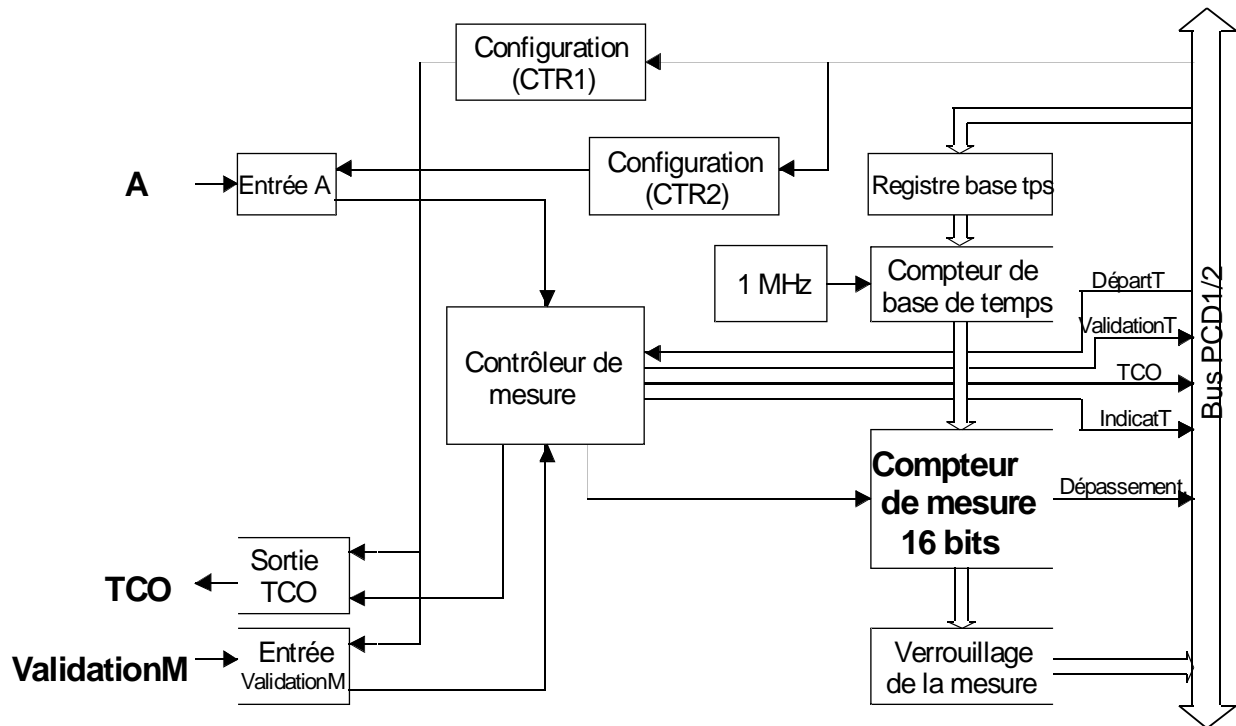
```

Ce projet est donc constitué de deux programmes : « move-02.sfc » avec l'initialisation du H110 et « frequ-02 » sans le bloc XOB 16.

Le déplacement peut être visualisé en ligne en GRAFTEC, avec la fréquence correspondante sur l'afficheur.

## 9.2 Mesure de durée de période

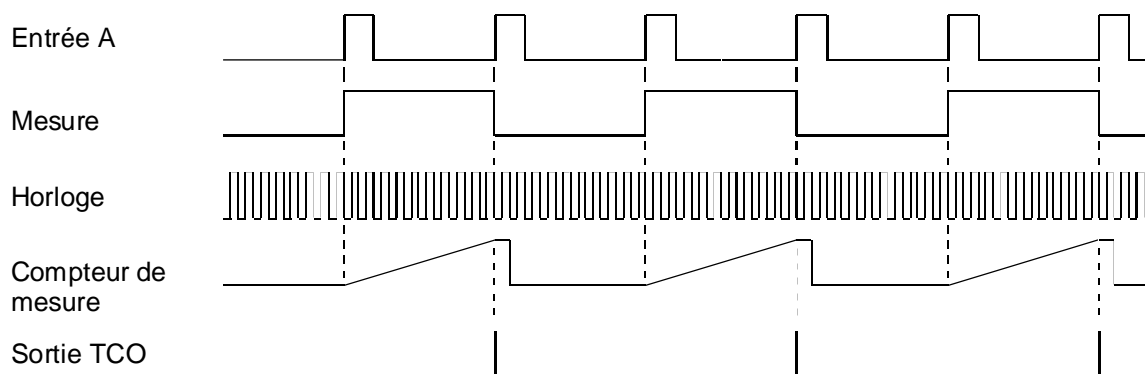
### 9.2.1 Schéma synoptique



### 9.2.2 Principe de fonctionnement

La mesure de durée de période utilise 2 compteurs de 16 bits :

- Le premier, « compteur de base de temps », possède une horloge fixe de 1 MHz qui fournit une base de temps fondamentale de 1  $\mu$ s. C'est à ce niveau que se génère la base de temps définie par l'utilisateur.
- Le second, « compteur de mesure », compte les tops d'horloge entre deux fronts montants de l'entrée A. En conséquence, lorsque des impulsions consécutives apparaissent en A, la mesure s'effectue toujours par paires d'impulsions ; après quoi, le processus marque une pause pour se préparer à la mesure suivante.





### 9.2.3 Configuration de la mesure de durée de période

La configuration fait appel au bloc de fonctions INIT. Pour une mesure en continu, le paramètre 9 doit être à 3 (mode période-automatique).

Une mesure en manuel est possible, mais sans grande utilité ; le paramètre 9 doit alors valoir 2 (période-manuel). Dans ce cas, une seule période est mesurée, puis l'ensemble de la procédure se répète pour réaliser une nouvelle mesure.

Le paramètre 7 permet d'inverser le signal sur l'entrée A.

#### Plage de mesure et base de temps

La plage de mesure s'étend de 0 à 65 535 (16 bits).

La formule ci-dessous permet de calculer la valeur à saisir pour définir la base de temps :

$$n = \frac{T * 10^6}{\text{tops}} - 1$$

avec : T = durée de la période en secondes  
 tops = nombre de tops d'horloge  
 n = valeur à saisir

Exemple : Prenons une période de durée 10 s et un nombre de tops d'horloge s'élevant à 10 000. On obtient :

$$n = \frac{10 * 10^6}{10\ 000} - 1 = 999$$

Le paramètre 11 définit le comportement de l'entrée ValidationM, et le paramètre 12, celui de la sortie TCO.

= 7	Configuration de l'entrée A		Entier	0 à 3	0 : C=normal M=normal 1 : C=inversé M=normal 2 : C=normal M=inversé 3 : C=inversé M=inversé
= 9	Configuration du mode de mesure		Entier	0 à 5	2 = période-manuel 3 = période-automatique
= 10	Chargement de la valeur de mesure		Entier	0 à 65 535	Base de temps (mesure de durée de période)
= 11	Configuration de l'entrée ValidationM		Entier	0 à 3	0 = statique-normal 1 = statique-inversé 2 = dynamique-normal 3 = dynamique-inversé
= 12	Configuration de la sortie TCO		Entier	0 à 3	0 = statique-normal 1 = statique-inversé 2 = dynamique-normal 3 = dynamique-inversé

Ce tableau est reproduit in extenso en page 2 de l'annexe A.

### 9.2.4 Méthode de programmation

Tâche : Une barrière photoélectrique (ou un contact antirebond) doit être reliée à l'entrée A. Il faut alors mesurer et visualiser sur l'afficheur l'intervalle de temps séparant deux fronts montants. L'ensemble doit être configuré de façon que pour une période de durée 1 seconde, le nombre de tops comptés s'élève à 10 000. La base de temps est fixée à 99 (voir la formule de calcul en 9.2.3).

```

; *****
; * Programme utilisateur PCD2.H110 destiné à la *
; * Mesure de durée de période : peri-01.src      *
; *****
;
$include d2h110_b.equ
$group h110

xob      16

ld       r 0          ; registre de service vide
         0

cfb      init        ; initialisation du H110
         k 1          ; par. 1  n° de module
         0            ; par. 2  inutilisé en mesure
         r 0          ; par. 3  inutilisé en mesure
         r 0          ; par. 4  inutilisé en mesure
         0            ; par. 5  inutilisé en mesure
         0            ; par. 6  inutilisé en mesure
         3            ; par. 7  entrée A = inversée
         0            ; par. 8  inutilisé
         3            ; par. 9  mode = période-autotique
         99           ; par. 10 base de temps en µs
         0            ; par. 11 ValidationM=statistique-normale
         2            ; par. 12 TCO = dynamique-normale

cfb      exec        ; départ mesure
         k 1          ; n° de module
         StartMs     ; commande : départ mesure
         r 0          ; registre vide

exob

; -----

cob      0
         0

wait:   sth          EndMes_1 ; mesure terminée ?
         jr          l wait   ; si non, attente

cfb      exec        ; lecture du résultat de la mesure
         k 1          ; n° de module
         RdMsImp     ; commande : lecture mesure en impulsions
         r 777       ; lecture de la valeur dans R 777

dsp      r 777       ; visualisation sur afficheur

ecob

; -----

xob      20          ; interruption INB1
com      o 101       ; inversion après chaque mesure
exob

$endgroup

```

## Description du programme

Au niveau matériel, un transmetteur de signal fournissant les signaux à mesurer (une barrière photoélectrique, par exemple) doit être relié à l'entrée A, l'entrée B restant ouverte.

L'entrée ValidationComptage, inutile à la mesure, reste ouverte.  
L'entrée ValidationMesure devrait certes recevoir le +24 V, mais dans la mesure où elle a été inversée à la configuration, elle reste ouverte <sup>\*)</sup>.

La sortie TCO est reliée à l'entrée interruptive INB1 du PCD. Au terme de chaque mesure, le XOB 20 est appelé : il inverse ici la sortie O 101 du PCD, ce qui permet de mieux visualiser le déroulement du programme.

La base de temps de cet exemple est fixée à 99 pour une fenêtre de temps de 10000 ms, soit 1 s (Cf. « Plage de mesure et base de temps » en 9.2.3).

L'entrée A est inversée (paramètre 7 du bloc INIT à 3), puisque la barrière photoélectrique fournit un signal inversé.

Le COB attend la fin de chaque mesure avant de lire le résultat du compteur de mesure, visualisable par le débogueur ou sur un afficheur au format entier. L'unité de mesure est dictée par la base de temps préalablement définie (ici, 1/10000 de secondes).

Si l'on choisit la commande « RdMsUnit » (lecture dans l'unité donnée) au lieu de « RdMsImp » (lecture en nombre d'impulsions), le résultat est toujours converti en secondes ; il peut être visualisé dans le registre PCD, en virgule flottante.

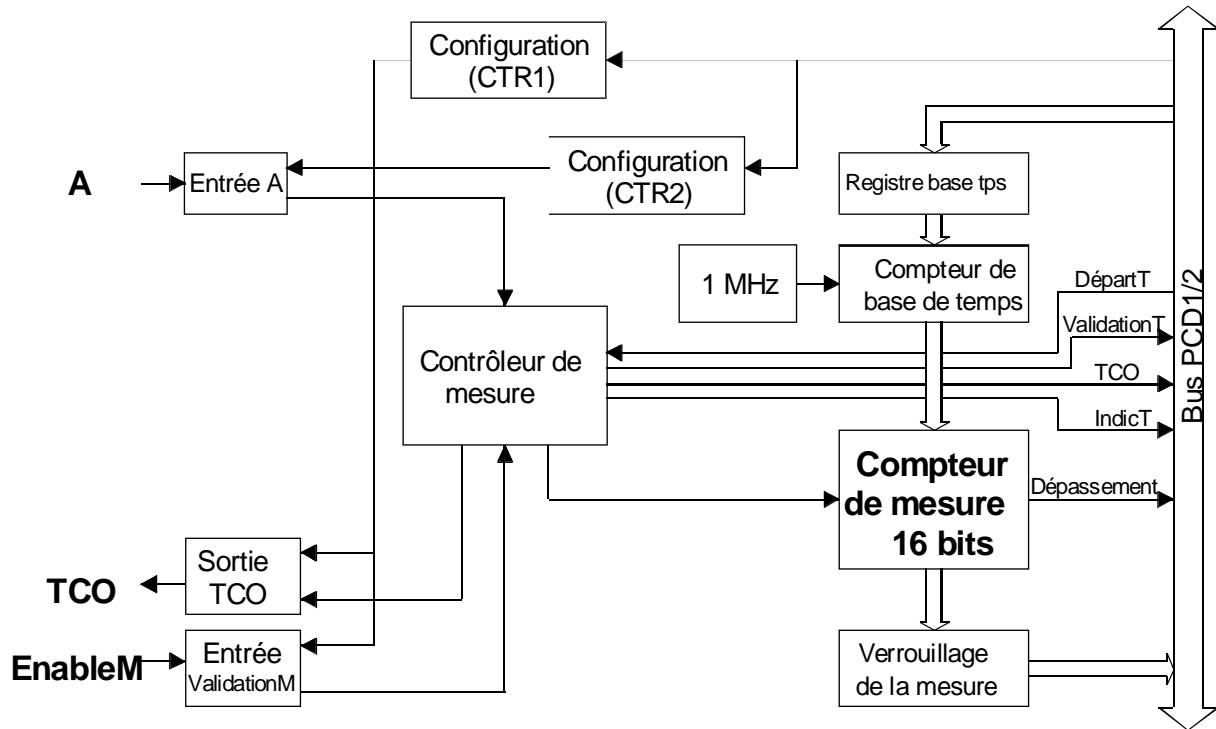
La commande « StopMs » interrompt une mesure en cours, sans résultat valide. La mesure peut reprendre avec la commande « StartMs ».

\*) Une désactivation de l'entrée ValidationM invalide le résultat de la mesure, qui ne peut être relancée que par la commande StartMs.

**Notes personnelles :**

## 9.3 Mesure de durée d'impulsion

### 9.3.1 Schéma synoptique

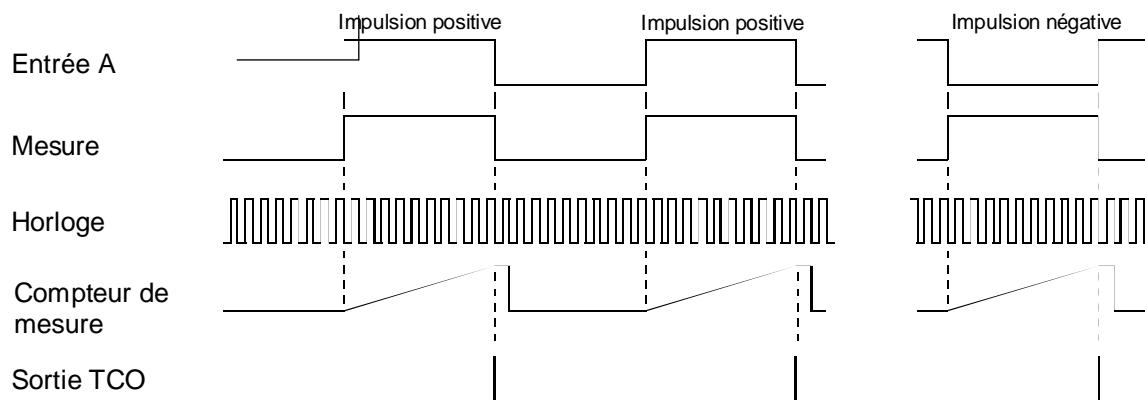


### 9.3.2 Principe de fonctionnement

La mesure de durée d'impulsion utilise 2 compteurs de 16 bits :

- Le premier, « compteur de base de temps », possède une horloge fixe de 1 MHz qui produit une base de temps fondamentale de 1  $\mu$ s. C'est à ce niveau que se génère la base de temps définie par l'utilisateur.
- Le second, « compteur de mesure », compte les tops d'horloge lorsque l'entrée A est à l'état haut (mesure sur impulsion positive ou « normale ») ou à l'état bas (mesure sur impulsion négative ou « inversée »).

Remarque : La mesure de durée d'impulsion négative ou « inversée » est obtenue en positionnant à 3 le paramètre 7 du bloc INIT.



### 9.3.3 Configuration de la mesure de durée d'impulsion

La configuration fait appel au bloc de fonctions INIT. Pour une mesure en continu, le paramètre 9 doit être à 1 (mode impulsion-automatique).

Une mesure en manuel est possible, mais sans grande utilité ; le paramètre 9 doit alors valoir 0 (impulsion-manuel). Dans ce cas, une seule impulsion est mesurée, puis l'ensemble de la procédure se répète pour réaliser une nouvelle mesure.

#### Plage de mesure et base de temps

La plage de mesure s'étend de 0 à 65 535 (16 bits).

La formule ci-dessous permet de calculer la valeur à saisir pour définir la base de temps :

$$n = \frac{T * 10^6}{\text{tops}} - 1$$

avec : T = durée d'impulsion en secondes  
 tops = nombre de tops d'horloge  
 n = valeur à saisir

Exemple : Prenons une durée d'impulsion de 10 s et un nombre de tops d'horloge s'élevant à 10 000. On obtient :

$$n = \frac{10 * 10^6}{10\,000} - 1 = 999$$

Le paramètre 11 définit le comportement de l'entrée ValidationM, et le paramètre 12, celui de la sortie TCO.

= 7	Configuration de l'entrée A		Entier	0 à 3	0 : C=normal M=normal 1 : C=inversé M=normal 2 : C=normal M=inversé 3 : C=inversé M=inversé
= 9	Configuration du mode de mesure		Entier	0 à 5	0 = impulsion-manuel 1 = impulsion-automat.
= 10	Chargement de la valeur de mesure		Entier	0 à 65 535	Base de temps (mesure de durée d'impulsion)
= 11	Configuration de l'entrée ValidationM		Entier	0 à 3	0 = statique-normal 1 = statique-inversé 2 = dynamique-normal 3 = dynamique-inversé
= 12	Configuration de la sortie TCO		Entier	0 à 3	0 = statique-normal 1 = statique-inversé 2 = dynamique-normal 3 = dynamique-inversé

Ce tableau est reproduit in extenso en page 2 de l'annexe A.

### 9.3.4 Méthode de programmation

Tâche : Une barrière photoélectrique (ou un contact antirebond) doit être reliée à l'entrée A. Il faut alors mesurer et visualiser sur l'afficheur la durée d'une impulsion (entrée A à l'état haut ou à l'état bas). L'ensemble doit être configuré de façon que pour une durée de 1 seconde, le nombre de tops comptés s'élève à 1 000. La base de temps est fixée à 999 (voir la formule de calcul en 9.3.3).

```

; *****
; * Programme utilisateur PCD2.H110 destiné à la *
; * Mesure de durée d'impulsion : imp-01.src      *
; *****
;
#include d2h110_b.equ
$group h110

xob      16

ld       r 0      ; registre de service vide
        0

cfb      init     ; initialisation du H110
        k 1      ; par. 1  n° de module
        0      ; par. 2  inutilisé en mesure
        r 0      ; par. 3  inutilisé en mesure
        r 0      ; par. 4  inutilisé en mesure
        0      ; par. 5  inutilisé en mesure
        0      ; par. 6  inutilisé en mesure
        3      ; par. 7  entrée A = inversée
        0      ; par. 8  inutilisé
        1      ; par. 9  mode = impulsion-autotique
        999    ; par. 10 base de temps en µs
        0      ; par. 11 ValidationM=statistique-normale
        2      ; par. 12 TCO = dynamique-normale

cfb      exec     ; départ mesure
        k 1      ; n° de module
        StartMs ; commande : départ mesure
        r 0      ; registre vide

exob

; -----

cob      0
        0

wait:   sth      EndMes_1 ; mesure terminée ?
        jr      l wait   ; si non, attente

cfb      exec     ; lecture du résultat de la mesure
        k 1      ; n° de module
        RdMsImp ; commande : lecture mesure en impulsions
        r 777   ; lecture de la valeur dans R 777

dsp      r 777   ; visualisation sur afficheur

ecob

; -----

xob      20      ; interruption INB1
com      o 101   ; inversion après chaque mesure
exob

$endgroup

```



## Description du programme

Au niveau matériel, un transmetteur de signal fournissant les signaux à mesurer (une barrière photoélectrique, par exemple) doit être relié à l'entrée A, l'entrée B restant ouverte.

L'entrée ValidationComptage, inutile à la mesure, reste ouverte.  
L'entrée ValidationMesure devrait certes recevoir le +24 V, mais dans la mesure où elle a été inversée à la configuration, elle reste ouverte <sup>\*)</sup>.

La sortie TCO est reliée à l'entrée interruptive INB1 du PCD. Au terme de chaque mesure, le XOB 20 est appelé : il inverse ici la sortie O 101 du PCD, ce qui permet de mieux visualiser le déroulement du programme.

La base de temps de cet exemple est fixée à 999 pour un résultat en millisecondes (Cf. « Plage de mesure et base de temps » en 9.3.3).

L'entrée A est inversée (paramètre 7 du bloc INIT à 3), puisque la barrière photoélectrique fournit un signal inversé.

Le COB attend la fin de chaque mesure avant de lire le résultat du compteur de mesure, visualisable par le débogueur ou sur un afficheur au format entier. L'unité de mesure est dictée par la base de temps préalablement définie (ici, des millisecondes).

Si l'on choisit la commande « RdMsUnit » (lecture dans l'unité donnée) au lieu de « RdMsImp » (lecture en nombre d'impulsions), le résultat est toujours converti en secondes ; il peut être visualisé dans le registre PCD, en virgule flottante.

La commande « StopMs » interrompt une mesure en cours, sans résultat valide. La mesure peut reprendre avec la commande « StartMs ».

\*) Une désactivation de l'entrée ValidationM invalide le résultat de la mesure, qui ne peut être relancée que par la commande StartMs.

**Notes personnelles :**

## Annexe A Récapitulatif des blocs de fonctions et des commandes PCD2.H110 en langage IL

### Bloc de fonctions « INIT »

#### INIT

Initialisation du module PCD2.H110

N° de module	→	= 1	<b>Init</b> Bloc de fonctions
Configuration du compteur	→	= 2	
Valeur compt. de présélection	→	= 3	
Valeur reg. de présélection	→	= 4	
Configuration de ValidationG	→	= 5	
Configuration de CCO	→	= 6	
Configuration de l'entrée A	→	= 7	
Configuration de l'entrée B	→	= 8	
Configuration de la mesure	→	= 9	
Valeur de la mesure	→	= 10	
Configuration de ValidationM	→	= 11	
Configuration de TCO	→	= 12	
Niveaux de blocs : 1			
Modification de l'index : non			
Temps d'exécution : 5 ms *)			

\*) mesuré sur PCD2.M120

#### Description et fonctionnement :

Ce bloc définit le paramétrage du PCD2.H110 et lit son adresse de base dans le fichier D2H110\_B.MBA.

Le paramètre 1 s'exprime sous forme de constante K.

Les paramètres 3 et 4 sont des adresses (absolues ou symboliques) de registres du PCD.

Tous les autres paramètres sont des nombres entiers.

**Description des E/S intervenant dans la commande :**

Par.	Désignation	Type	Format	Plage	Remarques
= 1	N° de module	K	K n	K 1 à K 16	
= 2	Configuration du mode de comptage		Entier	0 à 4	0 / 1 = mode x1 2 = mode x2 3 = mode x3 4 = mode x4
= 3	Chargement de la valeur du compteur de présélection	R	Entier	0 à 16 777 215	Valeur initiale du compteur
= 4	Chargement de la valeur du registre de présélection	R	Entier	0 à 16 777 215	Valeur de mise à 1 de CCO
= 5	Configuration de l'entrée ValidationC		Entier	0 à 3	0 = statique-normal 1 = statique-inversé 2 = dynam.-normal 3 = dynam.-inversé
= 6	Configuration de la sortie CCO		Entier	0 à 3	0 = statique-normal 1 = statique-inversé 2 = dynam.-normal 3 = dynam.-inversé
= 7	Configuration de l'entrée A		Entier	0 à 1	0 = normal (comptage) 1 = inversé (comptage) 2 = normal (mesure) 3 = inversé (mesure)
= 8	Configuration de l'entrée B		Entier	0 à 1	0 = normal 1 = inversé
= 9	Configuration du mode de mesure		Entier	0 à 5	0 = impulsion-manuel 1 = impulsion-autom. 2 = période-manuel 3 = période-autom. 4 = fréquence-manuel 5 = fréquence-autom.
= 10	Chargement de la valeur de mesure		Entier	0 à 65 535	Fenêtre de mesure (pour la mesure de fréquence) et base de temps (pour la mesure de durée d'impulsion ou de période)
= 11	Configuration de l'entrée ValidationM		Entier	0 à 3	0 = statique-normal 1 = statique-inversé 2 = dynam.-normal 3 = dynam.-inversé
= 12	Configuration de la sortie TCO		Entier	0 à 3	0 = statique-normal 1 = statique-inversé 2 = dynam.-normal 3 = dynam.-inversé

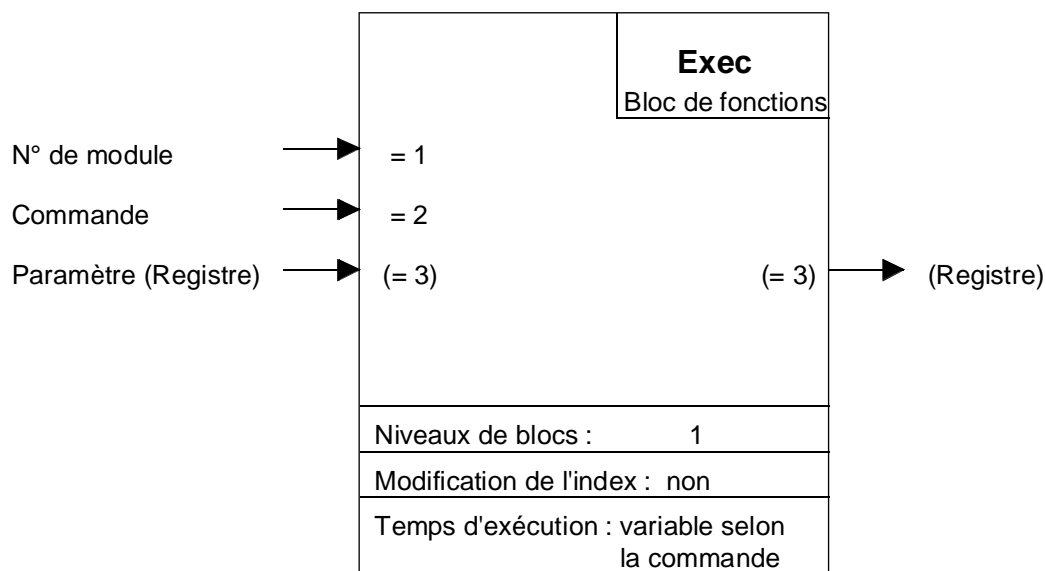
## Bloc de fonctions « EXEC »

---

### EXEC

Exécution d'une commande PCD2.H110

---



#### Description et fonctionnement :

Ce bloc de fonctions envoie des commandes au PCD2.H110.

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 D2H110\_B.MBA.

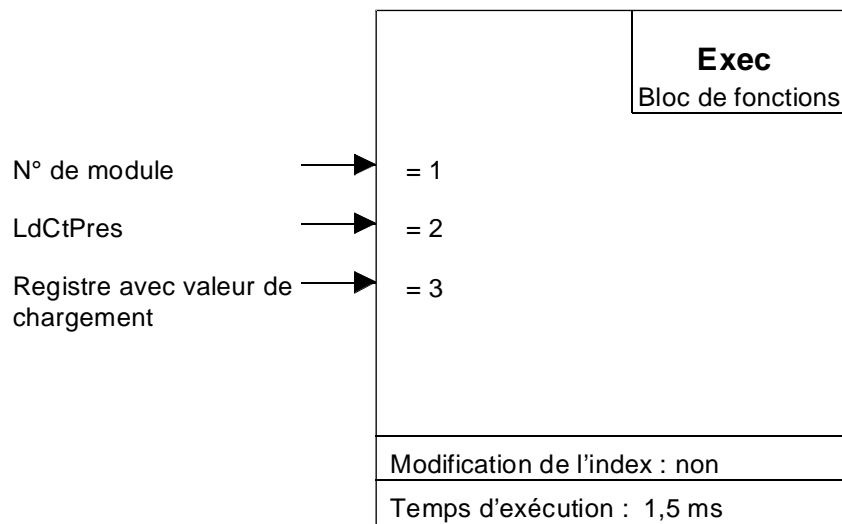
Ce bloc de fonctions accepte jusqu'à 16 modules PCD2.H110 par automate PCD.

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

Le paramètre d'une commande (par ex., la valeur du compteur de pré-sélection dans la commande « LdCtPres ») 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 départ comptage « StartCt »), on utilise n'importe quel registre ou le registre « rNotUse ».

## Commandes du bloc de fonctions « EXEC »

### LdCtPres      Chargement de la valeur du compteur de présélection



#### Description et fonctionnement :

Cette commande charge la valeur du compteur de présélection.

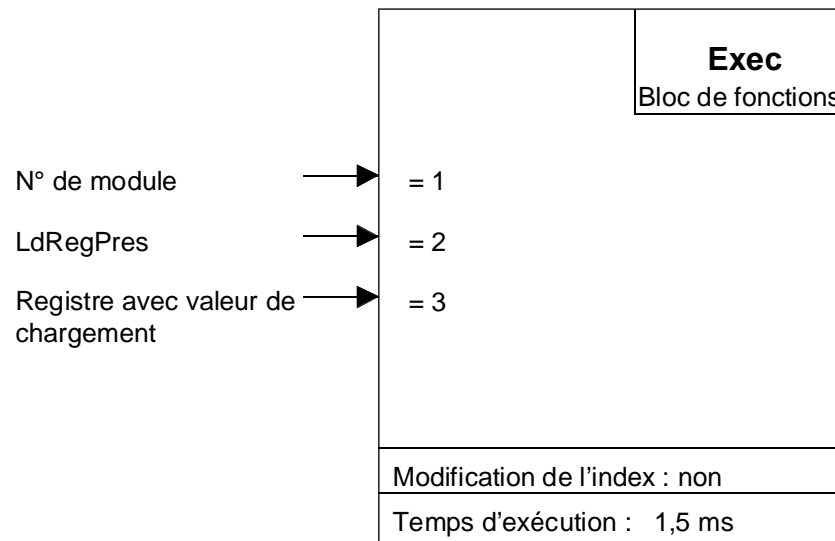
Celle-ci constituera la valeur de départ du comptage (ascendant ou descendant), après la commande correspondante « StartCt ».

#### Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Plage	Remarques
= 1	N° de module	K		1 à 16	
= 2	Commande : LdCtPres				Compteur 24 bits
= 3	Registre contenant la valeur de chargement	R	Entier	0 à 16 777 215	

## LdRegPres

Chargement de la valeur du registre de présélection



### Description et fonctionnement :

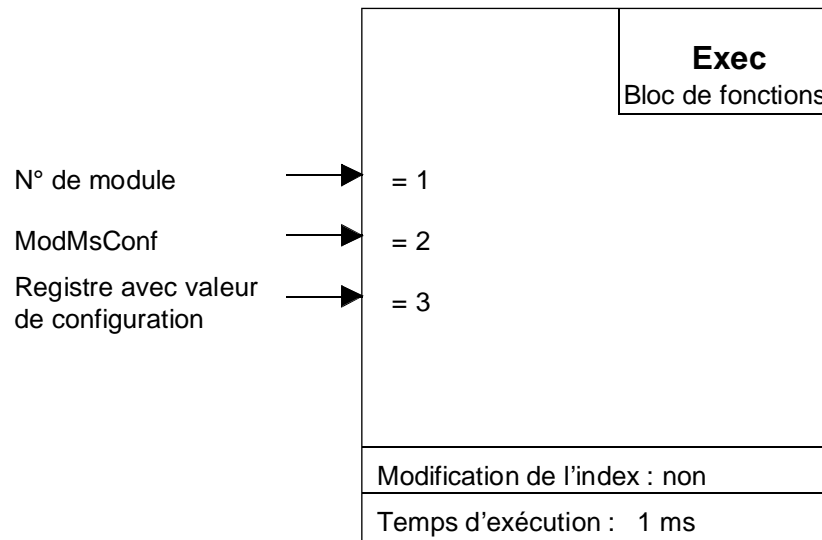
Cette commande charge la valeur du registre de présélection.

Celle-ci sera comparée à la valeur du compteur. En cas d'égalité, la sortie CCO sera activée, suivant sa configuration.

### Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Plage	Remarques
= 1	N° de module	K		1 à 16	
= 2	Commande : LdRegPres				Compteur 24 bits
= 3	Registre PCD contenant la valeur de chargement	R	Entier	0 à 16 777 215	

## ModMsConf Configuration du mode de mesure



### Description et fonctionnement :

Cette commande charge le mode de mesure choisi dans la configuration du registre de mesure.

En mode manuel, un nouveau départ compteur doit avoir lieu après chaque mesure ; en automatique, le premier départ doit être effectué par le programme utilisateur, puis il faut laisser finir la mesure par une condition (telle que CCO).

### Description des E/S intervenant dans la commande :

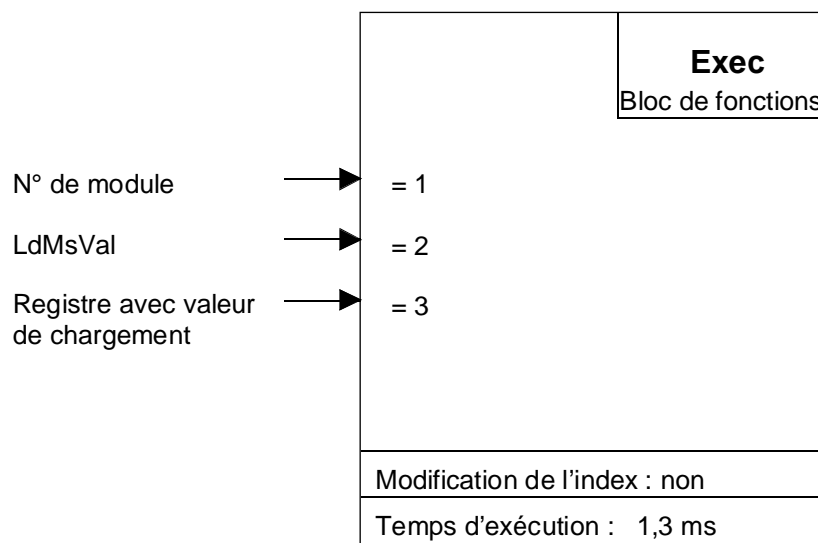
Par.	Désignation/Fonction	Type	Format	Plage	Remarques
=1	N° de module	K		1 à 16	
= 2	Commande : ModMsConf				
= 3	Registre PCD contenant la valeur de configuration	R	Entier	0 à 5	Valeur par défaut = 0

Cette commande n'a d'utilité que si la configuration doit être modifiée en cours de programme. Normalement, la configuration est effectuée en début de programme par le paramètre 9 du bloc de fonctions INIT.



## LdMsVal

Chargement de la valeur de mesure



### Description et fonctionnement :

Cette commande charge la valeur de la fenêtre de mesure (pour la mesure de fréquence) et de la base de temps (mesure de durée de période ou d'impulsion).

### Mesure de fréquence :

Pour obtenir une résolution de 0,1 %, il faut compter 1000 impulsions sur l'entrée A. La fenêtre de mesure est fonction de la fréquence à mesurer.

Exemples :

Pour mesurer une fréquence de :	La fenêtre de mesure doit valoir :
100 kHz	10 ms
500 Hz	2000 ms

Formule :

$$\begin{aligned}
 f &= \text{fréquence} \\
 t &= \text{durée de la fenêtre de mesure} & t &= \frac{1000 * R}{f} \\
 R &= \text{résolution}
 \end{aligned}$$

Mesure de durée de période ou d'impulsion : Cf. page suivante

Mesure de durée de période ou d'impulsion :

Le temps de mesure est toujours égal à la durée du signal ou de l'impulsion.

Exemples :

Pour mesurer une période de fréquence :	Le temps de mesure doit valoir :
500 Hz	2 ms
270 $\mu$ Hz	1 heure

La formule suivante permet de calculer la valeur de la base de temps. Une résolution de 0,1 % nécessite au moins 1000 impulsions.

$$\begin{aligned}
 t &= \text{durée de la période ou de l'impulsion (en s)} \\
 n &= \text{valeur à charger} & n &= \frac{10^6 * t}{\text{tops}} - 1 \\
 \text{tops} &= \text{nombre de tops d'horloge}
 \end{aligned}$$

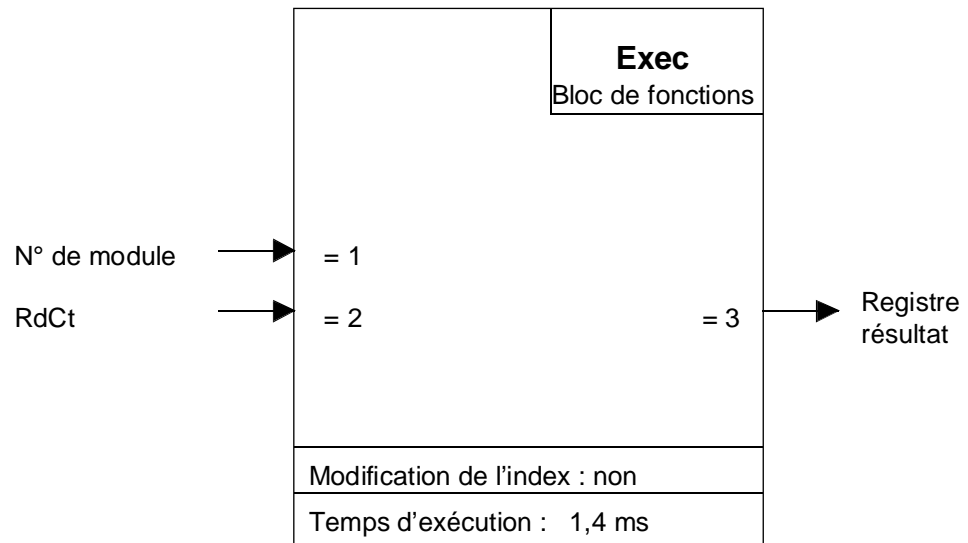
Vous devez soustraire 1 puisque lorsque la base de temps du compteur atteint 0, il lui faut 1  $\mu$ s pour se réinitialiser.

**Description des E/S intervenant dans la commande :**

Par.	Désignation/Fonction	Type	Format	Valeur	Remarques
= 1	N° de module	K		1 à 16	
= 2	Commande : LdMsVal				
= 3	Registre PCD contenant la valeur de chargement	R	Entier	0 à 65 535	16 bits

**RdCt**

Lecture du compteur

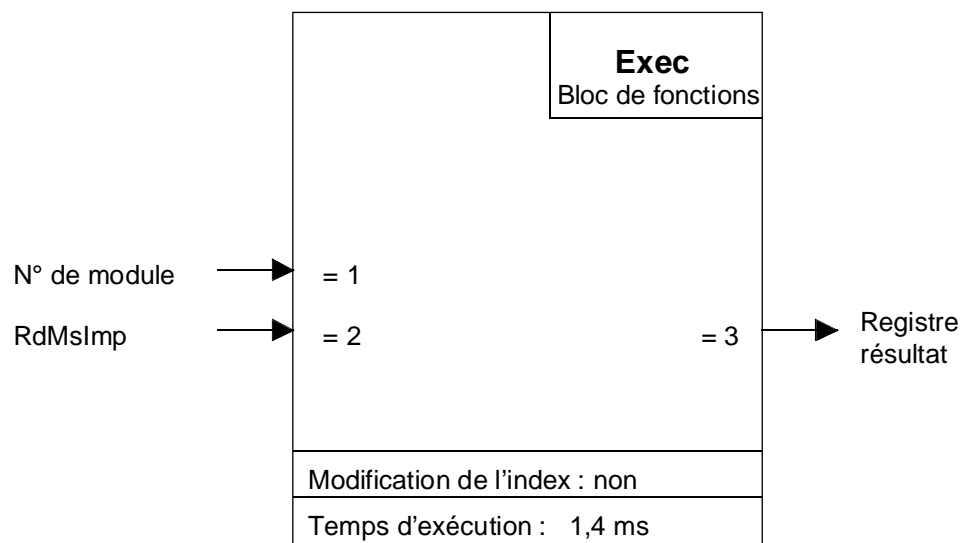
**Description et fonctionnement :**

Cette commande lit la valeur en cours du compteur.

**Description des E/S intervenant dans la commande :**

Par.	Désignation/Fonction	Type	Format	Valeur	Remarques
= 1	N° de module	K		1 à 16	
= 2	Commande : RdCt				
= 3	Registre PCD contenant le résultat de la lecture	R	Entier	0 à 16 777 215	24 bits

## RdMslmp Lecture de la mesure en nombre d'impulsions



### Description et fonctionnement :

Cette commande lit le résultat de la mesure en nombre d'impulsions.

Ce résultat revêt plusieurs formes selon le type de mesure :

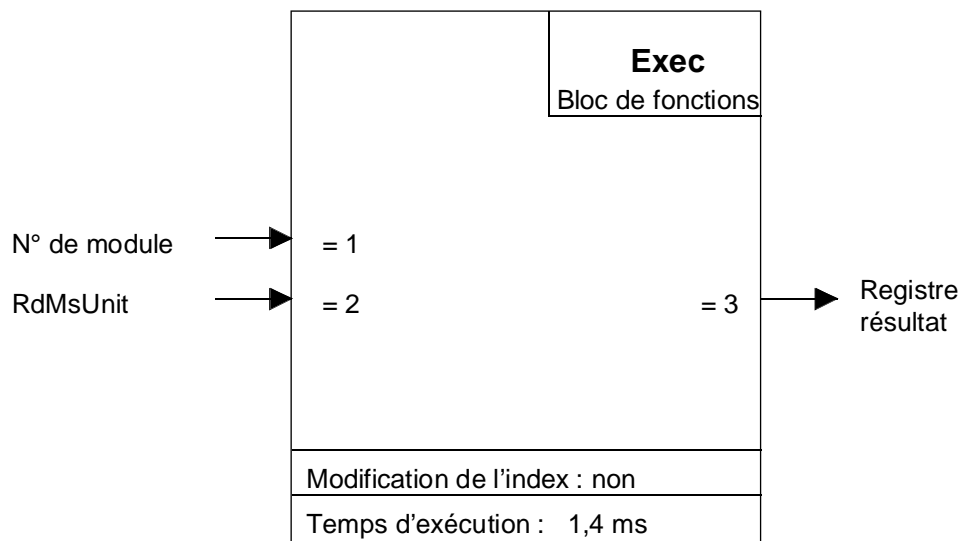
Mesure de fréquence : il s'agit du nombre d'impulsions durant l'ouverture de la fenêtre de mesure.

Mesure de durée de période ou d'impulsion : il s'agit du nombre d'impulsions entre deux fronts montants (durée de période) ou entre un front montant et un front descendant (durée d'impulsion).

### Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Valeur	Remarques
= 1	N° de module	K		1 à 16	
= 2	Commande : RdMslmp				
= 3	Registre PCD contenant le résultat de la lecture	R	Entier	0 à 65 535	16 bits

## RdMsUnit Lecture de la mesure dans l'unité donnée



### Description et fonctionnement :

Cette commande lit le résultat de la mesure dans l'unité dictée par le type de mesure :

Mesure de fréquence : résultat en Hertz ;

Mesure de durée de période ou d'impulsion : résultat en seconde (s).

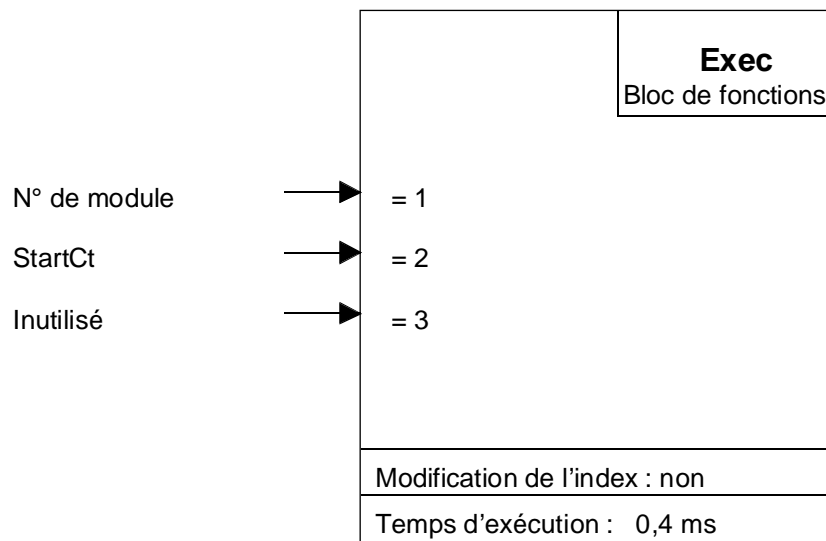
Dans les deux cas, le résultat est au format flottant.

### Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Valeur	Remarques
= 1	N° de module	K		1 à 16	
= 2	Commande : RdMsUnit				
= 3	Registre PCD contenant le résultat de la lecture	R	Virgule flottante		16 bits

## StartCt

Départ comptage



### Description et fonctionnement :

Cette commande lance le comptage si le signal ValidationComptage est à l'état haut, ou le met en attente si ValidationC est à l'état bas.

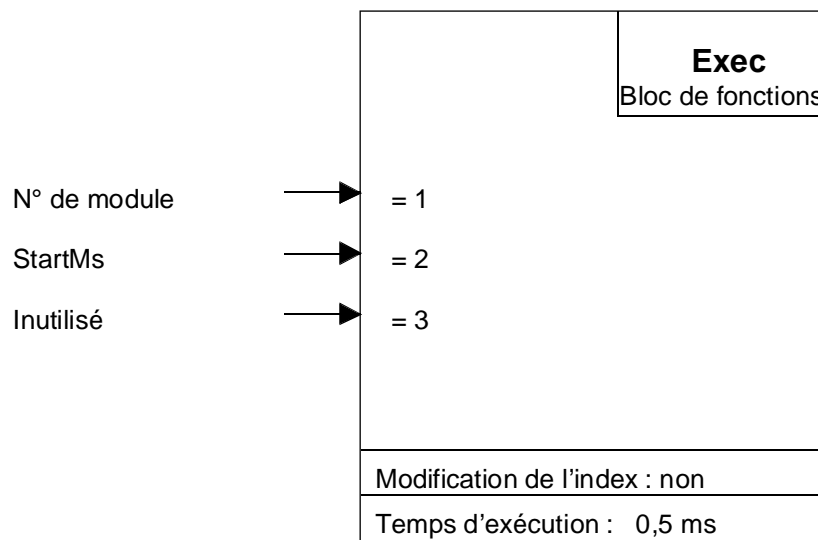
En mode manuel, cette commande doit avoir lieu après chaque mesure ; en automatique, elle n'est déclenchée qu'une fois, par le programme utilisateur.

### Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Valeur	Remarques
= 1	N° de module	K		1 à 16	
= 2	Commande : StartCt				
= 3	Registre PCD vide	R	Entier	0	

## StartMs Départ mesure

---



### Description et fonctionnement :

Cette commande lance la mesure si le signal ValidationMesure est à l'état haut, ou la met en attente si ValidationM est à l'état bas.

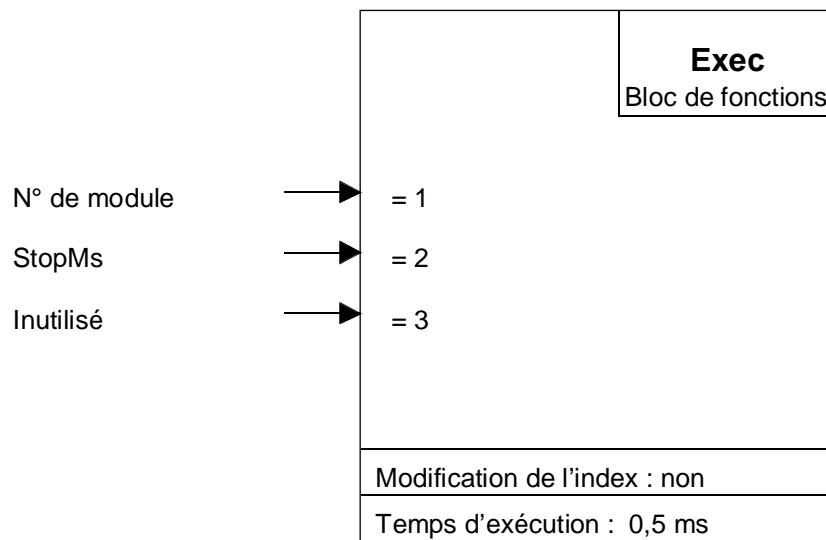
En mode manuel, cette commande doit avoir lieu après chaque mesure ; en automatique, elle n'est déclenchée qu'une fois, par le programme utilisateur.

### Description des E/S intervenant dans la commande :

Par.	Désignation/Fonction	Type	Format	Valeur	Remarques
= 1	N° de module	K		1 à 16	
= 2	Commande : StartMs				
= 3	Registre PCD vide	R	Entier	0	

**StopMs**

Arrêt mesure

**Description et fonctionnement :**

Cette commande arrête la mesure.

Les résultats des deux compteurs 16 bits sont donnés par la dernière mesure qui peut être terminée ou complètement fausse.

**Description des E/S intervenant dans la commande :**

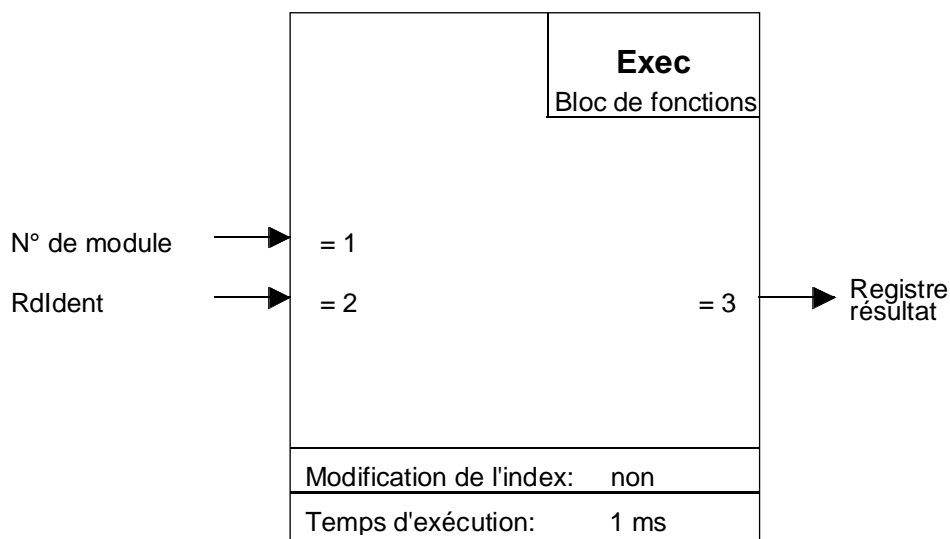
Par.	Désignation/Fonction	Type	Format	Valeur	Remarques
= 1	N° de module	K		1 à 16	
= 2	Commande : StopMs				
= 3	Registre PCD vide	R	Entier	0	

Pour relancer la mesure, il faut de nouveau exécuter la commande « StartMs ».



**RdIdent**

Lecture de l'identification du module

**Description et fonctionnement :**

Cette commande vérifie le bon fonctionnement du PCD2.H110 et indique la version du circuit FPGA.

Un fonctionnement correct du module donne la valeur 17xx (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	Remarques
= 1	N° de module	K		1 à 16	
= 2	Commande : RdIdent				
= 3	Registre PCD contenant le résultat de la lecture	R	Entier	12 bits	0 = dysfonctionnement

Identification du H110 :

Valeur	Version du FPGA
2759	HC0
1761	HC1
1762	HC2
1763	HC3
...	...
3215	HDF

**Notes personnelles :**

## **Annexe B Récapitulatif des boîtes de fonctions et des commandes PCD2.H110 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.H110

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

**Vos commentaires :**