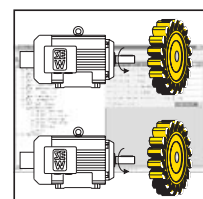
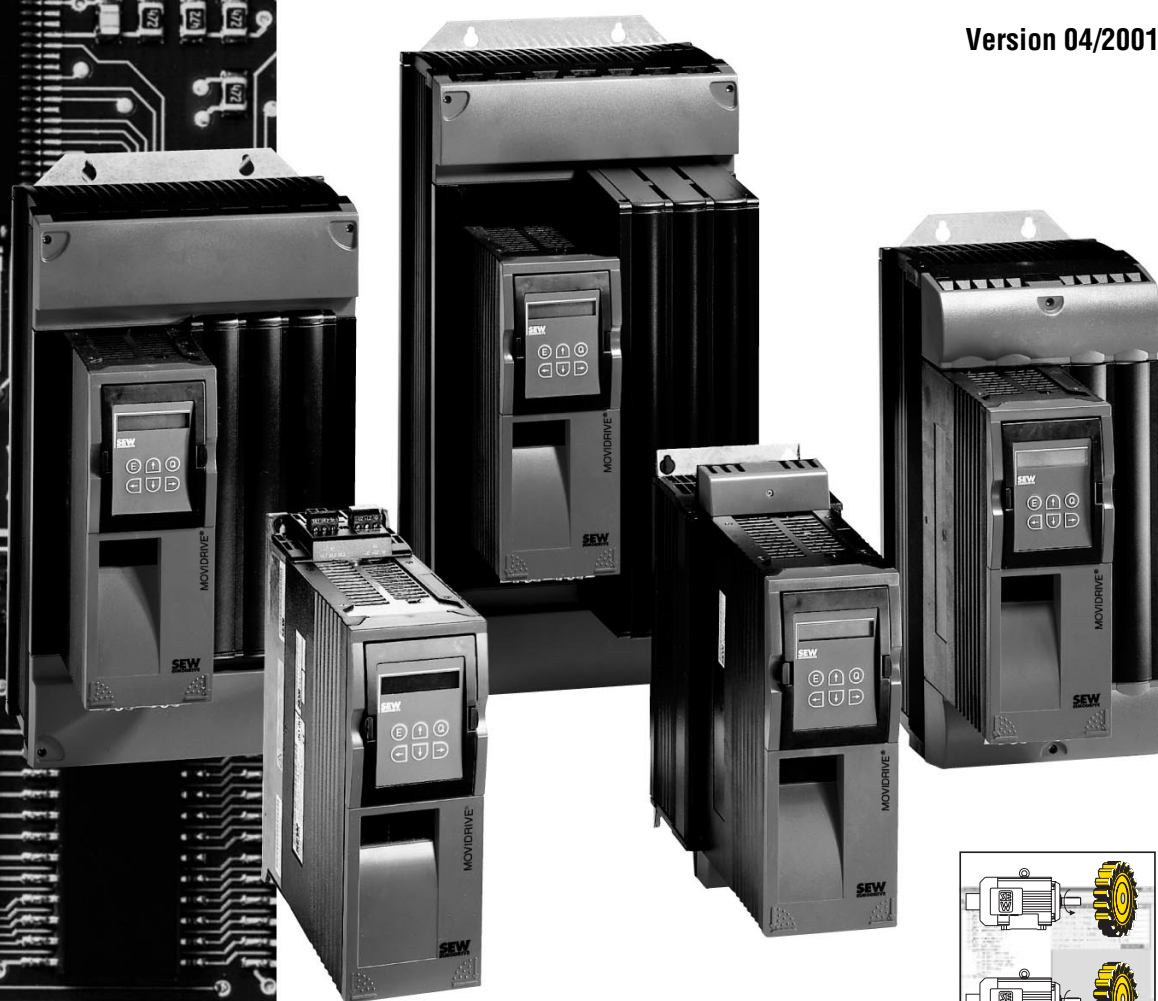


Variateurs MOVIDRIVE® MD_60A

Manuel

Synchronisation logicielle

Version 04/2001



SEW USOCOME

1050 3625 / 0401



- Le présent document ne remplace pas la notice d'exploitation **MOVIDRIVE®**!
- L'installation et la mise en service doivent être effectuées par des électriciens qualifiés en conformité avec les prescriptions en matière de prévention des accidents et selon les instructions du manuel **MOVIDRIVE®** !

- **Avant de commencer l'installation et la mise en service de la synchronisation interne pour tout variateur MOVIDRIVE®, il convient de lire attentivement ce manuel.**
Les personnes qui effectueront les opérations décrites dans ce manuel sont censées connaître la documentation (plus particulièrement le manuel pour variateurs **MOVIDRIVE®**) et savoir faire fonctionner les variateurs.
- **Consignes de sécurité:**
Respecter impérativement toutes les consignes de sécurité de ce manuel.
Les consignes de sécurité sont signalées de la façon suivante



Danger électrique, par exemple lors des travaux avec l'appareil sous tension.



Danger mécanique, par exemple dans le cas de travaux sur des appareils de levage.



Remarque importante pour un fonctionnement sûr et fiable de la machine de l'application, par exemple lors des réglages préliminaires avant la mise en route.

- Dans ce manuel, les **renvois à d'autres instructions** sont signalés par → par ex.: (→ Chap. X.X) D'autres informations sont données au chapitre X.X de ce manuel.
- Il est impératif de respecter les instructions et remarques du présent manuel afin d'obtenir un fonctionnement correct et de pouvoir bénéficier le cas échéant, d'un recours de garantie.

1	Description.....	4
1.1	Domaines d'utilisation	4
1.2	Fonctionnement.....	4
1.3	L'automate d'état de la synchronisation interne.....	5
1.4	Gestion de la synchronisation interne	5
2	Détermination	6
2.1	Exemples d'application	6
2.2	Conditions	8
2.2.1	PC et logiciels.....	8
2.2.2	Compilateur IPOS ^{plus} ®.....	8
2.2.3	Variateurs.....	8
2.2.4	Moteurs et codeurs	9
2.3	Consignes de détermination	10
2.4	Démarrage/arrêt synchronisé.....	11
3	Installation.....	12
3.1	Logiciels	12
3.2	Raccordement du codeur maître incrémental au MOVIDRIVE® esclave	13
3.3	Raccordement MOVIDRIVE® maître - MOVIDRIVE® esclave.....	14
3.4	Liaison Sbus maître-esclave	15
4	Mise en service	16
4.1	Généralités.....	16
4.2	Travaux préliminaires.....	16
4.3	Mise en service de la synchronisation interne.....	16
4.3.1	Généralités	16
4.3.2	Mise en service par la liaison X14-X14.....	17
4.3.3	Mise en service par la liaison Sbus.....	17
5	Fonctionnement	18
5.1	Gestion de la synchronisation interne	18
5.2	Pilotage de l'état principal.....	18
5.3	Commande de synchronisation.....	20
5.3.1	Rattrapage en vitesse/temps	20
5.3.2	Rattrapage en distance	20
5.3.3	Automate de synchronisation	21
5.4	Automate de désolidarisation.....	24
5.5	Gestion d'offset.....	25
5.5.1	Gestion d'offset en vitesse/temps.....	25
5.5.2	Gestion d'offset en distance	25
5.5.3	Automate d'offset	25
5.6	Synchronisme.....	27
5.7	Codeur virtuel	28
5.7.1	Codeur virtuel sans générateur de rampe.....	28
5.7.2	Codeur virtuel avec générateur de rampe	28
5.8	Remarques importantes.....	29
6	Variables système de la synchronisation interne	30
7	Exemples de programmes IPOS	33
7.1	Exemple 1	33
7.2	Exemple 2	36
7.3	Exemple 3.....	40

1 Description

1.1 Domaines d'utilisation

La synchronisation interne permet de piloter un groupe de moteurs en synchronisme de position angulaire ou avec des vitesses selon un rapport préalablement défini (réducteur électronique).

La synchronisation interne se prête en particulier aux applications suivantes :

- **Industrie des boissons**
 - chaînes de remplissage
- **Dispositif de levage multicolennes**
- **Transporteurs synchronisés**
- **Extrudeuses, outils de coupe pour matériaux sans fin**
 - scie volante
 - couteau rotatif

La synchronisation interne offre les avantages suivants :

- Possibilité de rattrapage en distance → synchronisation douce sans suroscillation
- Possibilité de gestion d'offset en distance
- Saisie du facteur de réduction du maître avec le signe
- Possibilité de synchronisation sur codeur virtuel
- Possibilité de liaison Sbus synchronisée entre maître et esclave
- Logiciel → ne nécessite pas d'installation de carte supplémentaire

1.2 Fonctionnement

La synchronisation interne est un applicatif spécifique qui ne réagit que sur des incréments envoyés par un maître. Ce dernier peut être l'entrée X14 du variateur (borne physique) ou une variable IPOS (borne virtuelle) (par ex. en combinaison avec le Sbus ou un codeur virtuel).

- **Rattrapage**

L'applicatif comprend un mécanisme de rattrapage en vitesse/temps. Celui-ci réduit à zéro tout décalage angulaire provoqué par la désolidarisation de l'esclave.

L'applicatif permet en outre un type de rattrapage particulier. L'esclave fonctionne en synchronisme de position angulaire par rapport au maître selon le nombre d'incrément défini par ce dernier (rattrapage en distance). Dans ce cas, l'esclave fonctionne selon une rampe quadratique.

- **Synchronisation**

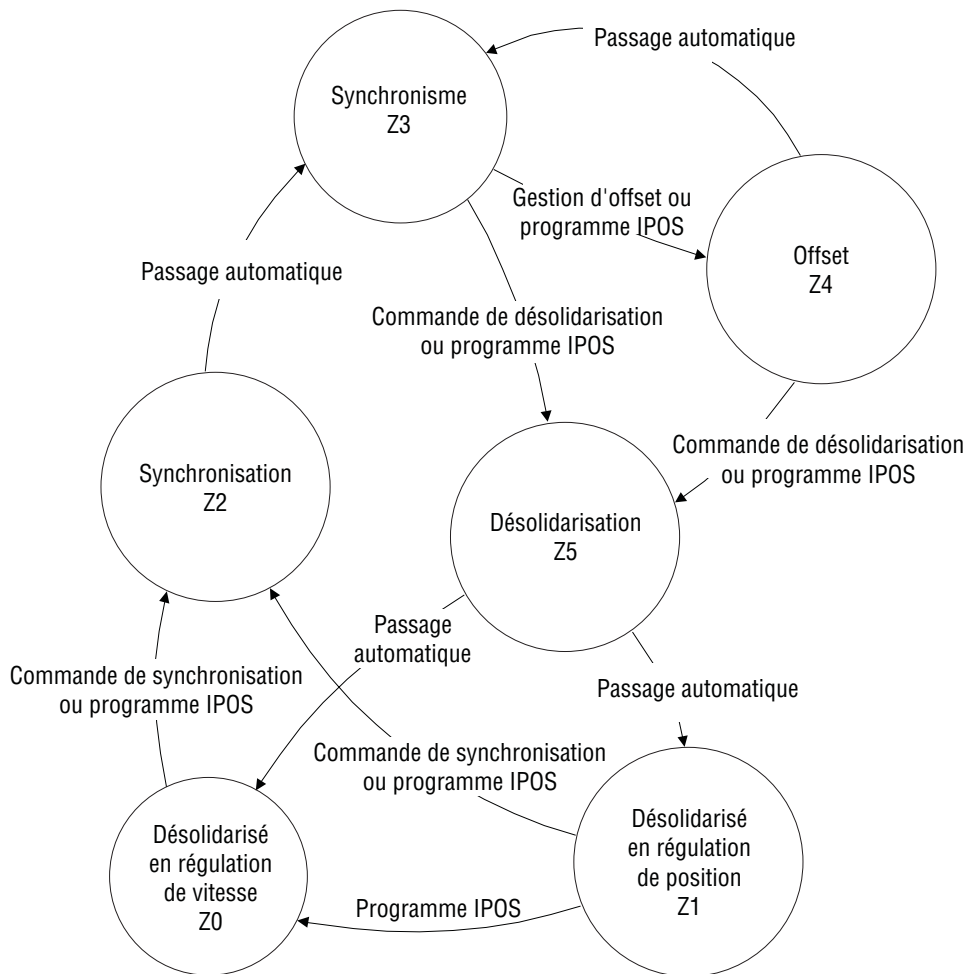
La synchronisation comprend différentes fonctions. Il est possible par exemple de réaliser un offset en distance. Dans ce cas, le nombre d'incrément défini par le maître sert d'offset entre le maître et l'esclave.

- **Désolidarisation**

Par la désolidarisation, le synchronisme entre le maître et l'esclave est rompu. La désolidarisation peut se faire, soit via une variable-système, soit par interruption événementielle via une entrée binaire.

1.3 L'automate d'état de la synchronisation interne

Les différentes fonctions de la synchronisation interne sont pilotées par un automate d'état. Celui-ci comprend cinq états principaux.



03778AFR

Fig. 1 : Aperçu des fonctions de l'automate d'état de la synchronisation interne

- Etat Z0 = Esclave désolidarisé en régulation de vitesse : l'esclave fonctionne de manière autonome en régulation de vitesse. Le décalage de l'esclave par rapport au maître est mémorisé dans un compteur d'écart.
- Etat Z1 = Esclave désolidarisé en régulation de position : l'esclave est asservi en position et ne dérive donc pas. La position par rapport au maître n'est pas mémorisée.
- Etat Z2 = Synchronisation : l'esclave est conduit vers le synchronisme avec le maître par rattrapage en distance ou en vitesse/temps.
- Etat Z3 = Synchronisme : l'esclave fonctionne en synchronisme avec le maître.
- Etat Z4 = Offset : possibilité de programmer un offset en synchronisme.
- Etat Z5 = Désolidarisation : l'esclave quitte le synchronisme avec le maître.

1.4 Gestion de la synchronisation interne

La gestion de la synchronisation interne s'effectue par les variables IPOS^{plus}® au sein du programme IPOS. L'ensemble des états est programmable et peut être visualisé au niveau des variables H360 à H446 spécialement réservées à la synchronisation interne (→ chap. 6).

2 Détermination

2.1 Exemples d'application

a) Deux moteurs en maître-esclave

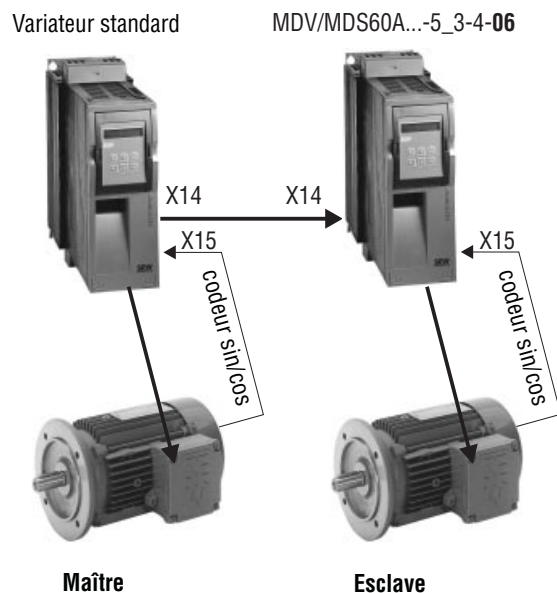


Fig. 2 : Fonctionnement maître-esclave

03779AFR

b) Deux moteurs en esclave avec codeur virtuel pour maître

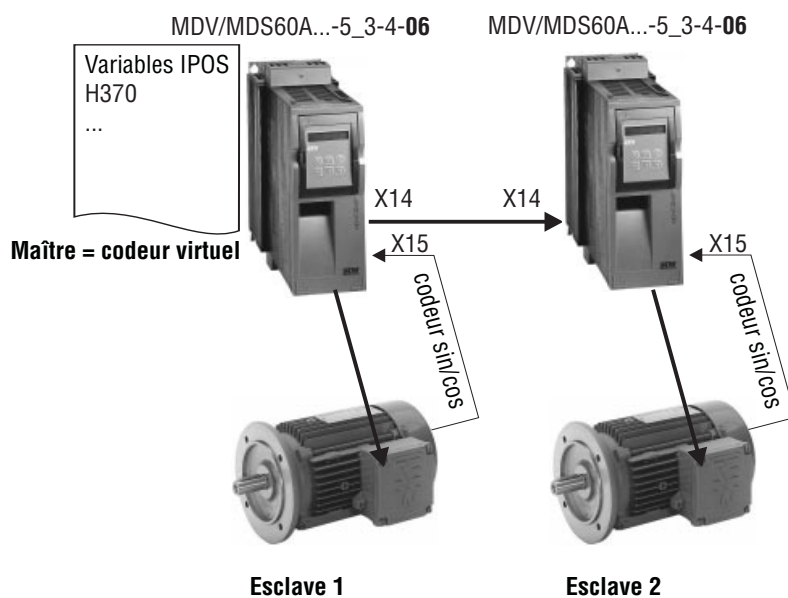
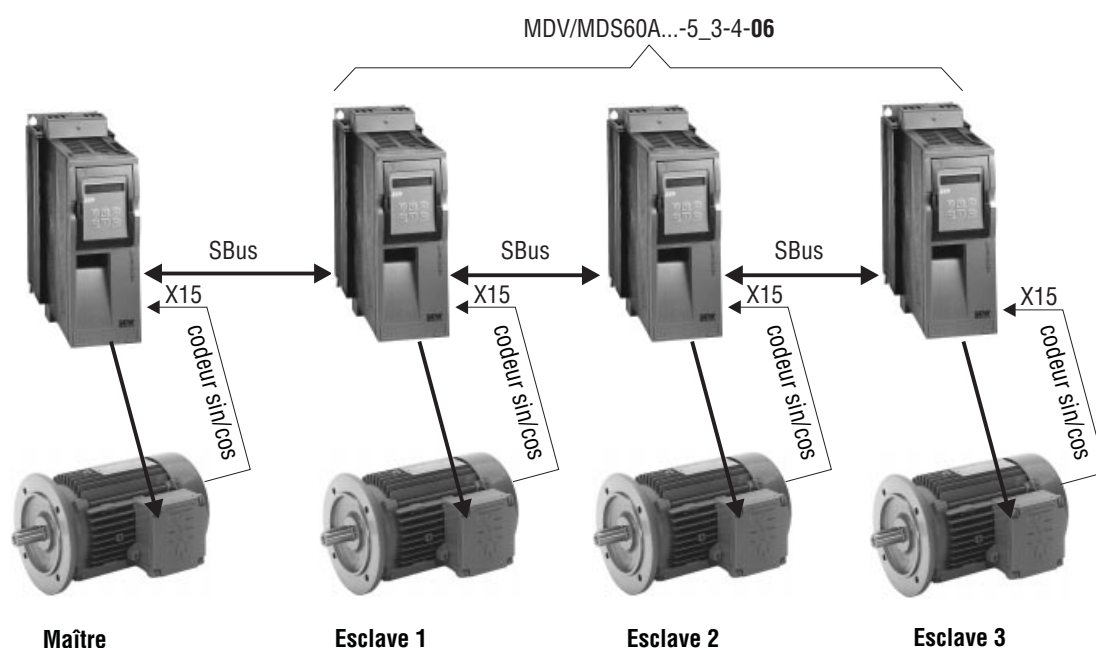


Fig. 3 : Fonctionnement maître-esclave avec codeur virtuel

03780AFR

c) Configuration de groupe : un maître et plusieurs esclaves du même rang

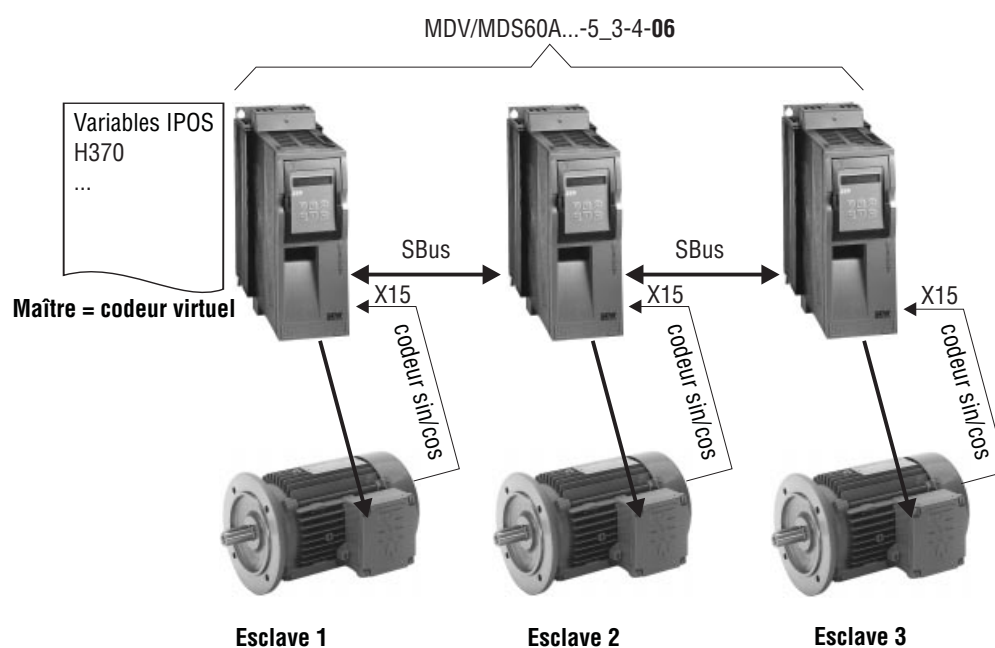
Ex.: dispositif de levage à plusieurs colonnes.



03558AFR

Fig. 4 : Configuration de groupe

d) Configuration de groupe avec codeur virtuel pour maître



03561AFR

Fig. 5 : Configuration de groupe avec codeur virtuel

2.2 Conditions

2.2.1 PC et logiciels

La synchronisation interne fait partie du logiciel MOVITOOLS de SEW. Pour utiliser MOVITOOLS, il faut disposer d'un PC avec le système d'exploitation Windows® 95, Windows® 98 ou Windows NT® 4.0.



2.2.2 Compilateur IPOS^{plus}®

Le programme utilisateur pour la synchronisation interne doit être réalisé au moyen du compilateur IPOS^{plus}®. Il n'est pas possible d'utiliser l'assembleur (programmation des masques).

Les variables IPOS^{plus}® H360 à H450 sont réservées à la synchronisation interne.

2.2.3 Variateurs

- L'exécution spéciale MOVIDRIVE® MD_60A... -5_3-4-06 comprend un applicatif pour la synchronisation interne.
- La synchronisation interne ne fonctionne qu'avec des MOVIDRIVE® MDV60A en mode CFC ou avec des MOVIDRIVE® MDS60A (SERVO). Elle ne fonctionne pas avec les MOVIDRIVE® MDV60A en mode VFC ou avec les MOVIDRIVE® MDF60A.
- Seul le jeu de paramètres 1 peut être utilisé ; le jeu de paramètres 2 n'est pas disponible pour la synchronisation interne.
- Les cartes de positionnement type DPI11A et de synchronisation type DRS11A ne sont pas supportées par l'applicatif et ne peuvent donc être utilisées.

Le tableau ci-dessous donne les références des exécutions spéciales à synchronisation interne:

MOVIDRIVE® MDV60A...	Référence
0015-5A3-4-06	826 994 7
0022-5A3-4-06	826 995 5
0030-5A3-4-06	826 996 3
0040-5A3-4-06	826 997 1
0055-5A3-4-06	826 998 X
0075-5A3-4-06	826 999 8
0110-5A3-4-06	827 000 7
0150-503-4-06	827 001 5
0220-503-4-06	827 002 3
0300-503-4-06	827 003 1
0370-503-4-06	827 004 X
0450-503-4-06	827 005 8
0550-503-4-06	827 006 6
0750-503-4-06	827 007 4

MOVIDRIVE® MDS60A...	Référence
0015-5A3-4-06	827 008 2
0022-5A3-4-06	827 009 0
0030-5A3-4-06	827 010 4
0040-5A3-4-06	827 011 2
0055-5A3-4-06	827 012 0
0075-5A3-4-06	827 013 9
0110-5A3-4-06	827 014 7
0150-503-4-06	827 015 5
0220-503-4-06	827 016 3
0300-503-4-06	827 017 1
0370-503-4-06	827 018 X

2.2.4 Moteurs et codeurs

- Pour le raccordement sur MOVIDRIVE® MDV60A :
 - Servomoteurs asynchrone CT/CV avec codeur sin/cos à haute résolution intégré de série.
 - Moteurs triphasés DT/DV/D avec codeur incrémental en option, de préférence un codeur sin/cos à haute résolution
- Pour le raccordement sur MOVIDRIVE® MDS60A :
 - Servomoteurs synchrone DS/DY avec resolver intégré de série.

Pour fonctionner de façon optimale, la synchronisation interne nécessite une mesure de vitesse à haute résolution. Les codeurs montés de série sur les moteurs CT/CV et DS/DY répondent à ces exigences. En cas d'utilisation de moteurs DT/DV/D, SEW conseille d'utiliser des codeurs sin/cos à haute résolution ES1S, ES2S ou EV1S.

2.3 Consignes de détermination

- Ne pas utiliser la fonction de synchronisation sur des entraînements à accouplement rigide.
- Le variateur esclave doit obligatoirement être équipé d'une résistance de freinage.
- Lors de la détermination, il convient de tenir compte du fait que l'esclave doit toujours être en mesure de ramener à zéro le décalage angulaire avec le maître. La vitesse maximale de l'esclave (P302) doit donc toujours être réglée à une valeur plus élevée que celle du maître.
- En cas de rattrapage en vitesse/temps, la vitesse de rattrapage de l'esclave doit être supérieure à la vitesse maximale du maître.
- Si possible, toujours utiliser des entraînements similaires pour la synchronisation interne.
- Pour les dispositifs de levage à plusieurs colonnes, toujours utiliser des moteurs et des réducteurs (rapports de réduction) identiques.
- Lorsque plusieurs entraînements similaires fonctionnent en synchronisme (par ex. dispositif de levage à plusieurs colonnes), il convient de choisir comme maître l'entraînement assumant la plus grande part de charge.
- Raccorder le codeur moteur de l'esclave à la borne X15 (ENCODER IN) et le codeur maître incrémental à la borne X14 (ENCODER IN/OUT) → manuel MOVIDRIVE®.
- Si le codeur incrémental raccordé à la borne X14 fonctionne comme maître : utiliser un codeur incrémental qui offre la plus haute résolution possible, sans toutefois dépasser 200 kHz.
- Sans carte d'option, il n'est pas possible d'utiliser des codeurs supplémentaires, tels que par ex. un codeur sur la machine, sauf en cas de montage de la carte codeur absolu type DIP11A sur l'esclave. Dans ce cas, il est possible d'utiliser un codeur absolu comme codeur externe.
- Utiliser exclusivement des esclaves avec liaison positive (= sans glissement) entre l'arbre moteur et la machine.
- Fonctionnement avec Sbus → prévoir un programme IPOS pour la transmission cyclique des données :
 - configuration de groupe : liaison Sbus possible entre le maître et tous les esclaves
 - synchronisation par Sbus avec transmission de l'identifiant de synchronisation Sbus.
 - transmission de la position du maître
 Important : la transmission par Sbus peut générer des temps morts.
- Démarrage/arrêt synchronisé (→ chap. 2.4)
- La **surveillance directe de rupture de câble codeur** (liaison X14-X14, raccordement codeur) n'est pas possible. En cas de fonctionnement par Sbus, il est possible de réaliser une **surveillance indirecte** de rupture de câble au moyen de la réaction programmable Time out Sbus (P836).

2.4 Démarrage/arrêt synchronisé

Pour certaines applications, par exemple un dispositif de levage à deux colonnes, il convient de s'assurer de la possibilité de démarrage et d'arrêt synchronisés du maître et de l'esclave pour garantir un fonctionnement correct de l'application. Toute combinaison caractérisée par un dynamisme plus élevé du maître par rapport à l'esclave est à proscrire.

Le tableau ci-dessous donne un aperçu des combinaisons maître-esclave possibles et des réglages nécessaires pour le démarrage/arrêt synchronisé.

Maître	Esclave	Paramètres maître	Paramètres esclave	Remarque
MDV	MDV	D002 = moteur alimenté	DI03 = Marche/arrêt rapide (réglage usine) DI01 et DI02 = sans fonction	Relier la sortie binaire maître D002 à l'entrée binaire esclave DI03
MDV	MDS			
MDS	MDS			

Important :

- La commande de frein doit être activée au niveau du maître et de l'esclave (P730 "Commande de frein 1" = ACTIVE).
- Le temps de déblocage du frein (P731) au niveau du maître doit être augmenté du temps de prémagnétisation (P323) de l'esclave.
- Le mode désolidarisé de l'esclave n'est possible que lorsque l'entrée binaire X13: 4 de l'esclave (DI03) reçoit un signal "1" d'une autre source que le maître.



3 Installation

3.1 Logiciels

Pour installer MOVITOOLS sur votre ordinateur, il convient de procéder comme suit :

1. Mettre le CD MOVITOOLS dans le lecteur CD de votre PC.
2. Lancer la commande “Démarrer\Exécuter...”.
3. Entrer “{nom du lecteur CD}:\setup” et valider par ENTREE.
4. Le programme lance le menu d’installation de MOVITOOLS. Suivre les indications qui apparaissent à l’écran au fur et à mesure que l’installation se poursuit.

Lancer le Manager de MOVITOOLS. Si un MOVIDRIVE® est relié au PC, sélectionner la liaison-série appropriée (liaison-série PC) et choisir sous “Connexion avec” l’option VARIATEUR UNIQUE. Cliquer sur <Actualiser> : le variateur apparaîtra dans la fenêtre “Appareils reconnus”.

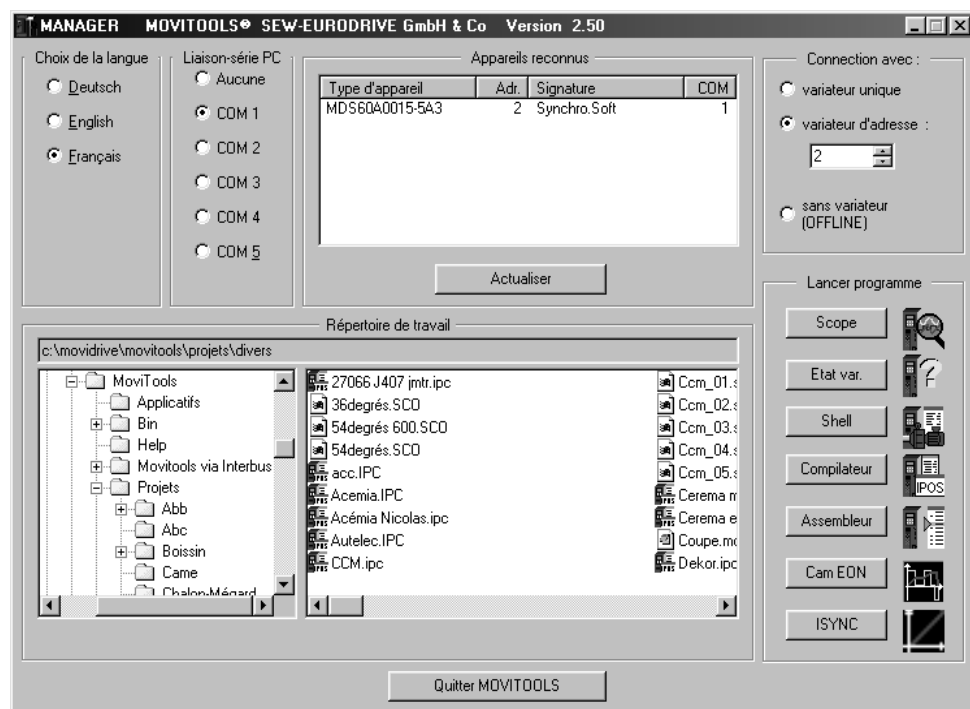
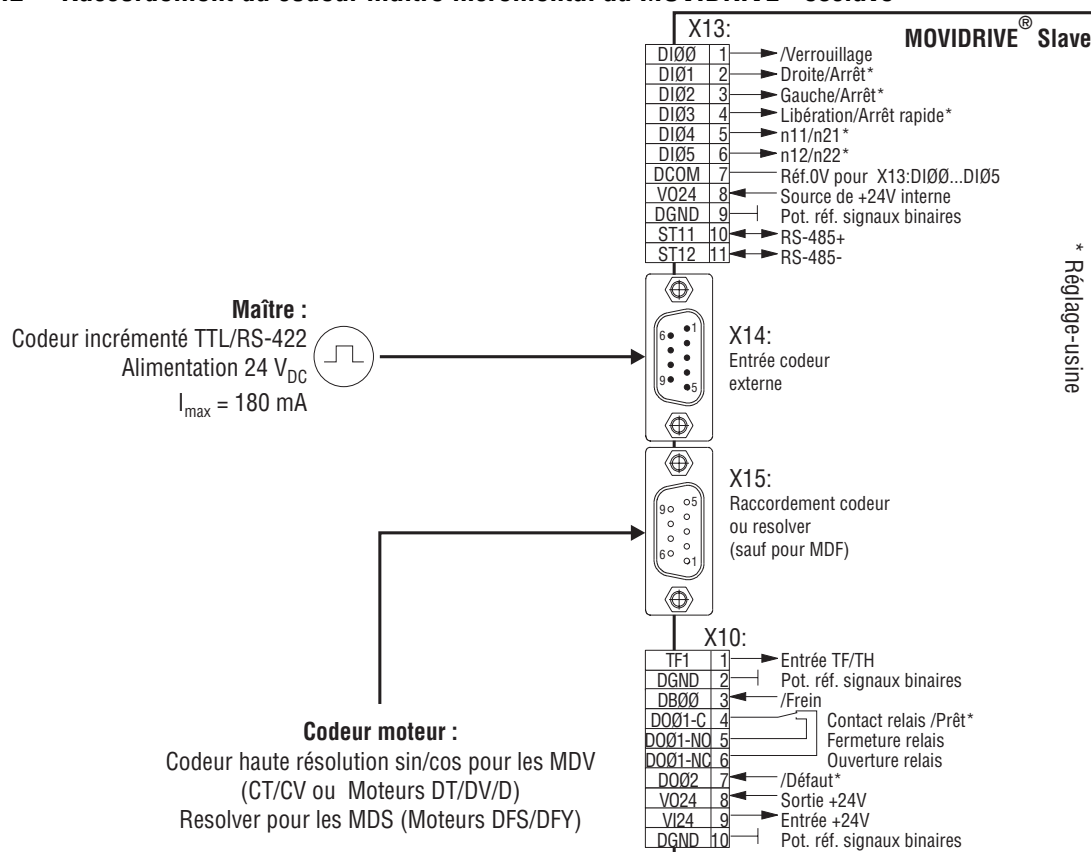


Fig. 6 : Fenêtre d'entrée MOVITOOLS

02745AFR

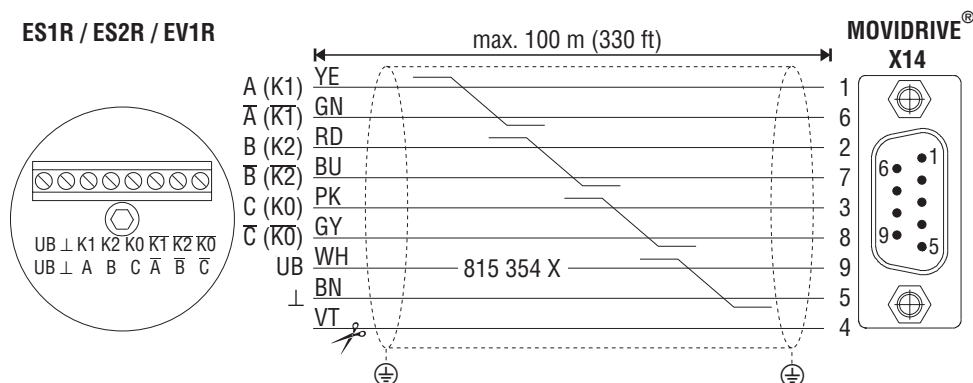
3.2 Raccordement du codeur maître incrémental au MOVIDRIVE® esclave



03405AFR

Fig. 7 : Schéma de branchement du codeur maître incrémental au MOVIDRIVE® esclave

Raccorder le codeur maître incrémental (codeur TTL avec alimentation 24 V_{CC}, par exemple ES1R, ES2R ou EV1R) comme suit :



03882AXX

✂ Couper le conducteur violet (VT) côté codeur

Fig. 8 : Schéma de branchement codeur maître incrémental à la borne X14

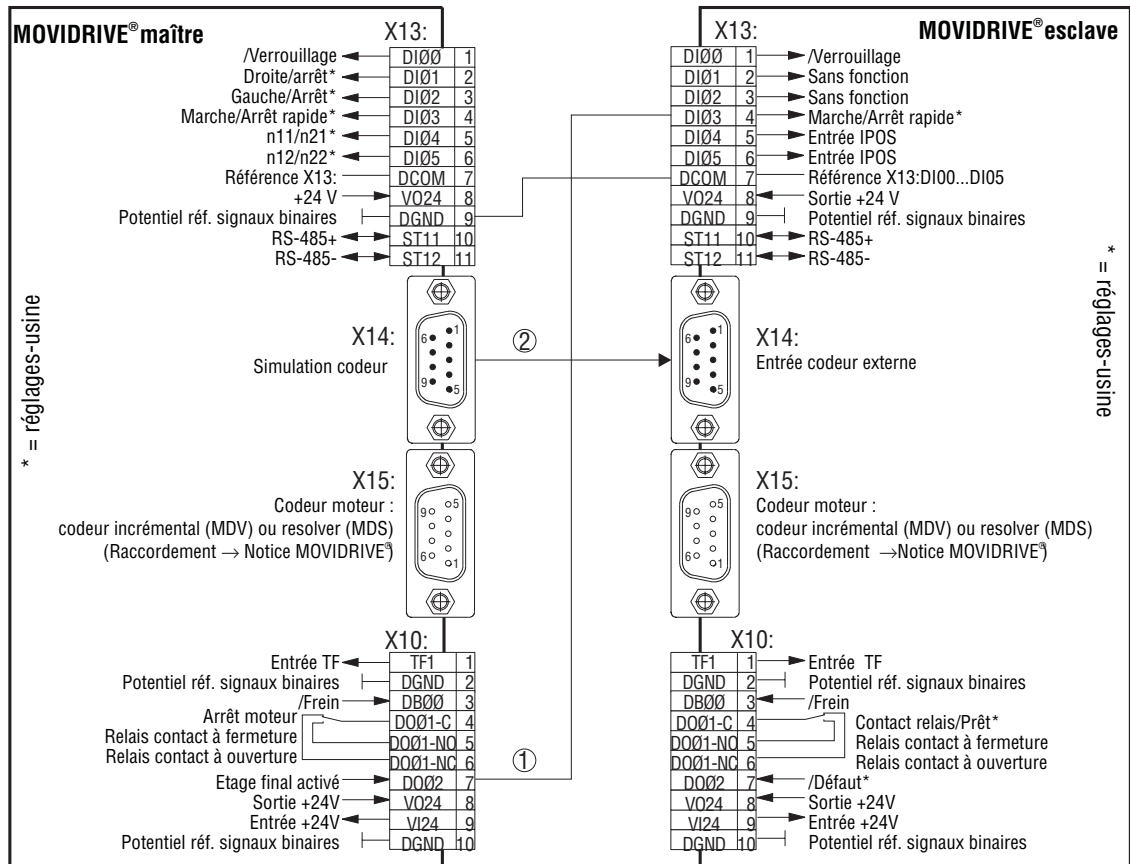
Important :

- La longueur maximale autorisée du câble codeur (codeur incrémental ou codeur moteur) est de 100 m (330 ft).
- Utiliser uniquement des câbles blindés avec paires torsadées (A et A', B et B', C et C'). Raccorder le blindage aux deux extrémités du câble.
- Installer les câbles codeur séparément des liaisons véhiculant la puissance.
- Pour simplifier le branchement et éviter toute erreur de raccordement des codeurs, SEW propose des câbles confectionnés prêts à l'emploi.



SEW
USOGOME

3.3 Raccordement MOVIDRIVE® maître - MOVIDRIVE® esclave



03848AFR

① Liaison nécessaire au démarrage/arrêt synchronisé (→ "Démarrage/arrêt synchronisé", page 11).

② Pour la liaison X14-X14, se référer à la Fig. 10 et aux consignes suivantes !

Fig. 9 : Câblage MOVIDRIVE® maître - MOVIDRIVE® esclave

Liaison X14-X14 :

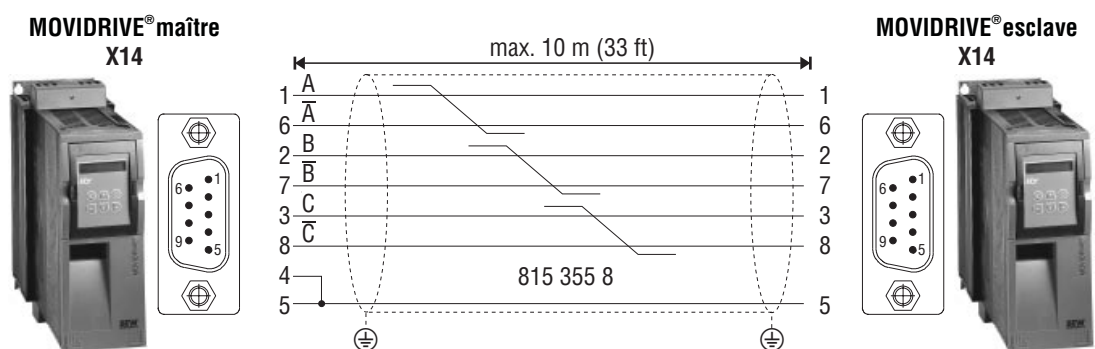


Fig. 10 : Liaison X14-X14 maître-esclave

03562AFR



Important :

- Sur le MOVIDRIVE® maître : faire un pontage entre X14: 4 et X14: 5
- Ne pas relier X14: 4 et X14: 9
- Pour simplifier le raccordement et éviter toute erreur dans la liaison X14-X14, SEW propose un câble confectionné prêt à l'emploi (réf. 815 355 8).

3.4 Liaison Sbus maître-esclave

Pour plus de détails sur le bus système, se référer au Manuel d'utilisation pour bus système (Sbus), disponible chez SEW (réf. 0918 0923).

Le bus système Sbus permet de relier au maximum 64 participants de bus CAN entre eux. Sbus supporte la technologie de transmission selon ISO 11898.

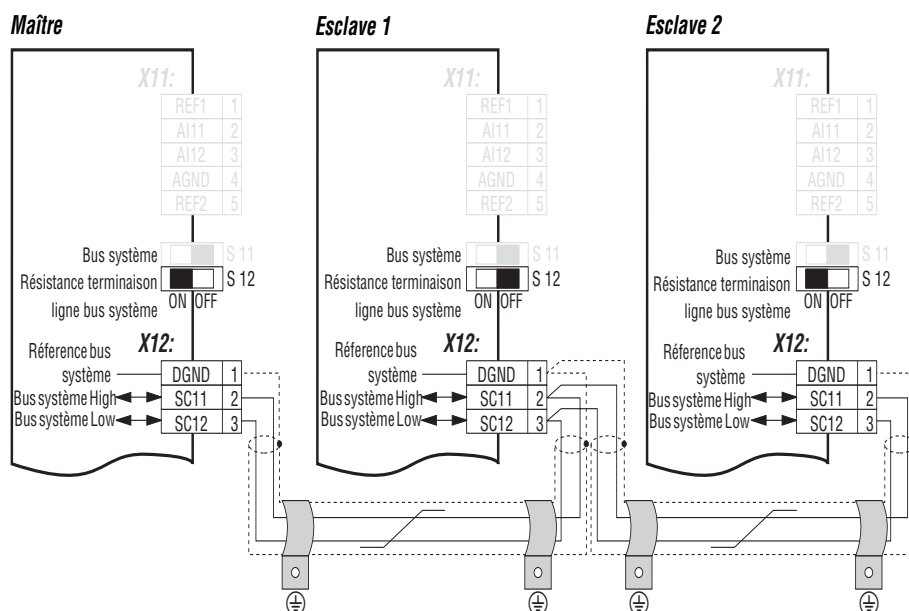


Fig. 11 : Raccordement du bus système (exemple : 1 maître et 2 esclaves)

03781AFR

Remarques :

- Utiliser un câble en cuivre bifilaire torsadé et blindé (câble de transmission de données avec blindage tresse en cuivre). Raccorder par un contact de grande surface les deux extrémités du blindage à la borne de blindage de l'électronique du MOVIDRIVE® ou au maître. Raccorder en outre au DGND

Le câble doit être conforme aux spécifications suivantes :

- section des conducteurs 0,75 mm² (AWG 18)
- résistance du câble 120 Ω à 1 Mhz
- capacité linéique ≤ 40 pF/m (12 pF/ft) à 1 Hz

L'on peut utiliser par exemple, les câbles pour bus CAN ou DeviceNet.

- La longueur totale autorisée du câble dépend du réglage de la vitesse de transmission du Sbus :
 - 125 kbauds → 320 m (1056 ft)
 - 250 kbauds → 160 m (528 ft)
 - 500 kbauds** → 80 m (264 ft)
 - 1000 kbauds → 40 m (132 ft)
- Activer la résistance de terminaison de ligne bus système aux deux extrémités de la liaison (S12 = ON). Sur les autres appareils, la résistance de terminaison de ligne doit être désactivée (S12 = OFF).
- Il ne doit pas y avoir de différence de potentiel entre les appareils reliés par Sbus. Ceci peut être évité grâce à des mesures appropriées, par exemple en reliant la masse des appareils à l'aide d'une liaison séparée.



SEW
USOCOME

4 Mise en service

4.1 Généralités

Les conditions préalables à une mise en service réussie sont le bon dimensionnement et l'installation correcte de l'entraînement. Des informations complémentaires figurent dans le manuel système MOVIDRIVE® (nous consulter).

Contrôler l'installation et le raccordement des codeurs en se référant aux indications de la notice d'exploitation MOVIDRIVE® MD_60A et de la présente notice (chap. 3, page 12).

4.2 Travaux préliminaires

Avant la mise en service de la synchronisation interne, effectuer les opérations suivantes :

- Raccorder le variateur au PC par la liaison-série (RS-232, USS21A sur PC-COM)
- Installer et lancer MOVITOOLS sur le PC (chap. 3.1, page 12).
- Appliquer un signal "0" à la borne X13:1 (DI00, /Verrouillage).
- Procéder à la mise en service du variateur avec le programme <Shell> :
 - pour les MOVIDRIVE® MDV60A avec moteurs DT/DV/D ou CT/CV en **mode d'exploitation CFC&IPOS**
 - pour les MOVIDRIVE® MDS60A avec moteurs DS/DY en **mode d'exploitation SERVO&IPOS**

4.3 Mise en service de la synchronisation interne

4.3.1 Généralités

- Lancer <Shell>.
- Activer la synchronisation interne en programmant le paramètre **P916 "Forme Rampe IPOS"** sur "I-SYNCHRONISATION"

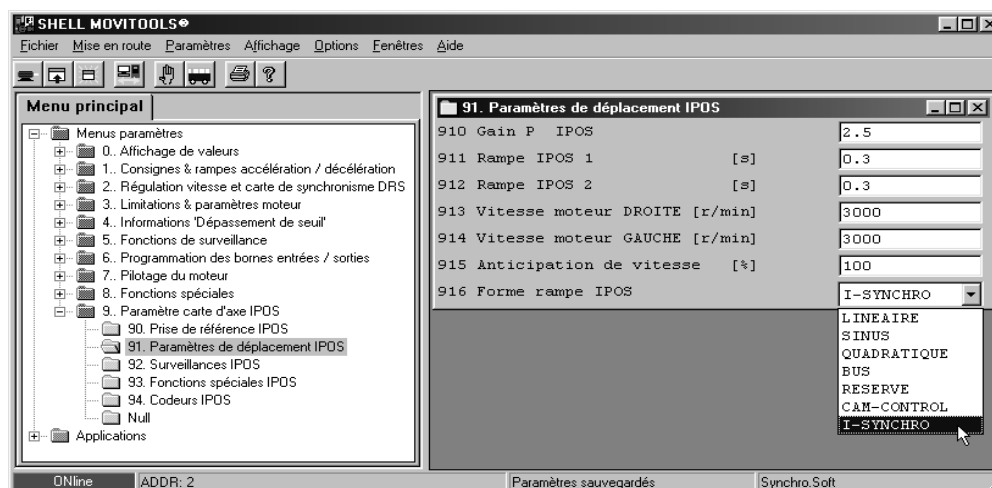


Fig. 12 : Activer la synchronisation interne via P916

03403AFR

Ce réglage peut également se faire au moyen de l'instruction "_SetSys (SS_RAMTYPE, Hxx)" dans le programme-utilisateur IPOS. La variable Hxx doit avoir la valeur 6.

4.3.2 Mise en service par la liaison X14-X14

La synchronisation interne utilise comme maître la simulation du codeur incrémental de la borne X14 d'un MOVIDRIVE® maître. La variable système H430 *MasterSource* de l'esclave doit obligatoirement être programmée sur "0", sinon la borne X14 reste inactive comme source d'incréments.

4.3.3 Mise en service par la liaison Sbus

Le maître et l'esclave sont reliés entre eux via le Sbus. C'est le cas par exemple dans une configuration de groupe. La position du maître est transmise par le Sbus. La transmission des consignes de position nécessite la synchronisation du maître et de l'esclave. Tenir impérativement compte des points suivants lors de la mise en service du Sbus :

Au niveau du maître :

- Créer deux objets d'envoi par Sbus (SCOM TRANSMIT CYCLIC), à savoir : "ID synchronisation" et "Position maître". Les deux numéros d'objet doivent être supérieurs à 1050. En outre, le numéro d'objet de l'ID synchronisation doit être inférieur (= priorité plus élevée) au numéro d'objet de la position du maître.
- Le numéro d'objet de l'ID synchronisation doit être différent de la valeur du paramètre P817.
- Le réglage du Sbus Adresse (P813) doit être différent du Sbus Adresse des esclaves.
- Le temps de cycle défini dans l'instruction SCOM pour l'ID synchronisation doit être de 5 ms.
- Le temps de cycle défini dans l'instruction SCOM pour la position du maître doit être de 1 ms.

Au niveau de l'esclave :

- Créer un seul objet de réception Sbus (SCOM RECEIVE), à savoir : Position maître.
- La valeur du paramètre P817 doit être identique au numéro d'objet d'envoi "ID synchronisation".
- La variable système H430 *MasterSource* doit être identique à la valeur du pointeur D (→ structure de l'instruction SCOM).
- Les esclaves doivent tous avoir une adresse Sbus (P813) différente.

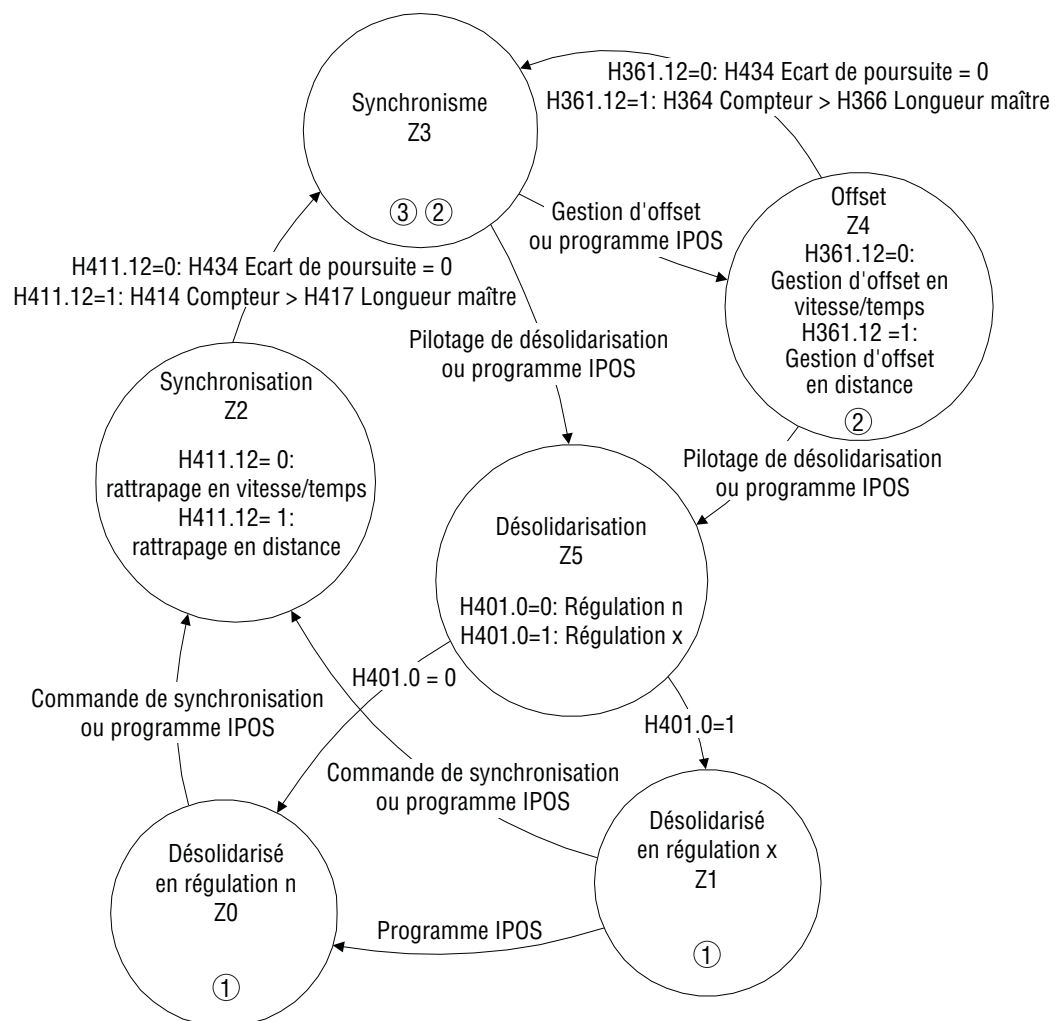
5 Fonctionnement

5.1 Gestion de la synchronisation interne

La synchronisation interne est pilotée via les variables IPOS^{plus}® au sein du programme IPOS^{plus}®, ci-après appelé "le programme". La plage des variables H360 à H446 est réservée à la synchronisation interne ; ces variables peuvent être programmées et visualisées par l'utilisateur (→ chap. 6). Toutes les variables en rapport avec la synchronisation interne sont dotées d'un nom symbolique, imprimé dans ce qui suit en ***gras et en italique***.

5.2 Pilotage de l'état principal

La figure ci-dessous montre l'automate d'état principal de la synchronisation interne (H427 → ***SynchronousState***) :



03406AFR

Sous-automates

- ① Automate de synchronisation → chap. 5.3.3, page 21
- ② Automate de désolidarisation → chap. 5.4, page 24
- ③ Automate d'offset → chap. 5.5.3, page 25

Fig.13 : Automate principal et sous-automates de la synchronisation interne

L'automate distingue 6 états ; ceux-ci sont programmés au niveau de la variable ***SynchronousState*** (H427).

Etat	Description
<i>SynchronousState</i> = 0	Désolidarisé en régulation n → l'esclave peut être piloté en régulation de vitesse au moyen de la variable H439 (<i>SpeedFreeMode</i>), le décalage étant mémorisé dans un compteur d'écart 64 bits.
<i>SynchronousState</i> = 1	Désolidarisé en régulation x → l'esclave est maintenu en position actuelle
<i>SynchronousState</i> = 2	Phase de synchronisation → synchronisation en fonction de l'état du bit 12 de la variable H411 (<i>StartupCycle-ModeControl</i>)
<i>SynchronousState</i> = 3	Synchronisme rigoureux → l'esclave suit rigoureusement le maître en position angulaire
<i>SynchronousState</i> = 4	Offset → mise en place de l'offset en fonction de l'état du bit 12 de la variable H361 (<i>OffsetCycle-ModeControl</i>).
<i>SynchronousState</i> = 5	Phase de désolidarisation → désolidarisation de l'esclave selon la Rampe t11 (P130) .

Les fonctions supplémentaires peuvent être sélectionnées au moyen des bits de la variable ***SynchronousModeControl*** (H426) :

Bit	Nom	Description
0	PosTrim	= 0: désactivé = 1: génère le déplacement de l'esclave vers la position <i>TargetPos</i> (H492) en cas de régulation de position en mode désolidarisé (Etat principal 1)
1	Lag Error	= 0: surveillance erreur de poursuite = 1: pas de surveillance d'erreur de poursuite

5.3 Commande de synchronisation

5.3.1 Rattrapage en vitesse/temps

En cas de rattrapage en vitesse/temps, le décalage entre le maître et l'esclave (compteur d'écart 64 bits) est neutralisé en accélérant ou en ralentissant l'esclave jusqu'au niveau de la vitesse de rattrapage. La durée nécessaire dépend de la vitesse et de la rampe de rattrapage ainsi que de l'écart de poursuite (H434, **LagDistance32**). La figure ci-dessous montre l'allure de la vitesse de l'esclave au cours de processus de rattrapage pour le cas d'un maître fonctionnant à vitesse constante.

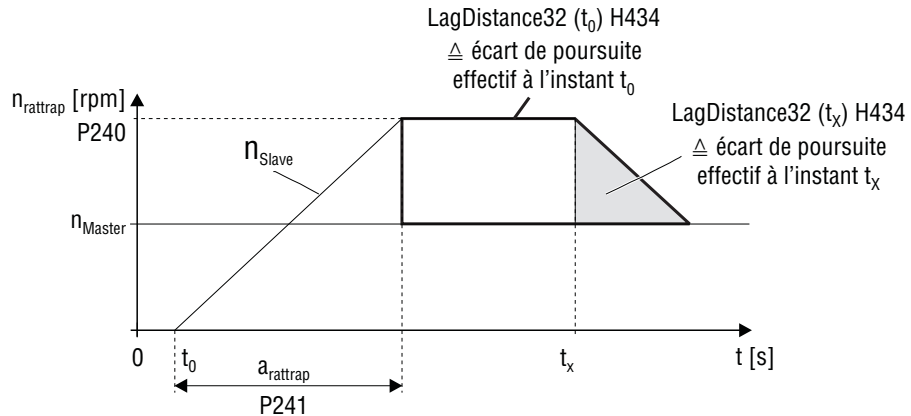


Fig.14 : Courbe de vitesse en cas de rattrapage en vitesse/temps

03409AFR

La vitesse de rattrapage n_{rattrap} et la rampe de rattrapage a_{rattrap} sont programmées au niveau du paramètre P240 "Vitesse de synchronisation" et P241 "Rampe de synchronisation". Ces deux paramètres ne sont actifs qu'en cas de rattrapage et de gestion d'offset en vitesse/temps.

5.3.2 Rattrapage en distance

Dans ce cas précis, l'esclave est synchronisé avec le maître dès que ce dernier a effectué un déplacement préalablement défini. Le déplacement doit être défini en incréments, rapportés au maître, au sein de la variable système **StartupCycleMasterLength** (H417). La contrainte imposée est le démarrage de l'esclave à la vitesse zéro.

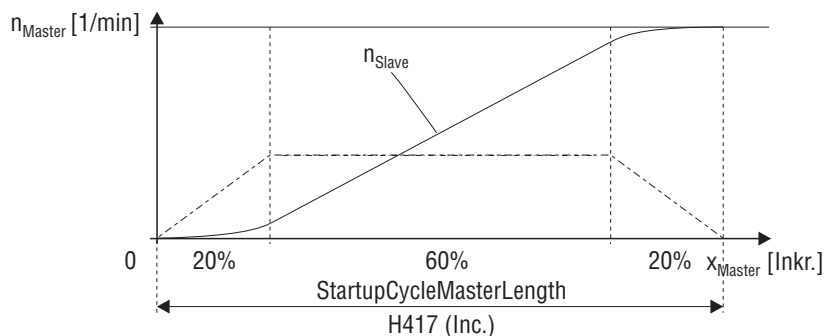


Fig.15 : Courbe de vitesse en cas de rattrapage en distance

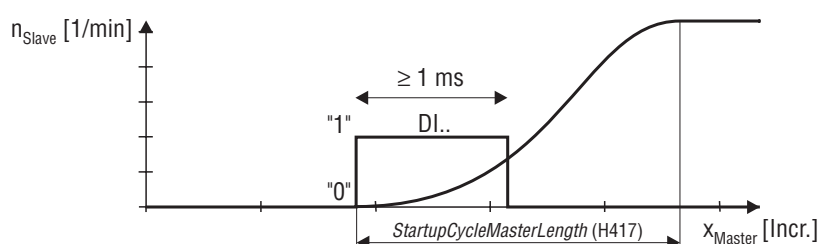
03410AFR

5.3.3 Automate de synchronisation

La commande de synchronisation est activable dans les états Z0 et Z1 (→ Fig. 13). La synchronisation de l'esclave peut se faire de façon manuelle, événementielle ou par interruption. Le mode de synchronisation est défini dans la variable système **StartupCycleMode** (H410). La variable système **StartupCycleModeControl** (H411) permet de programmer des fonctions supplémentaires.

Variable H410 (**StartupCycleMode**) → Mode-synchronisation :

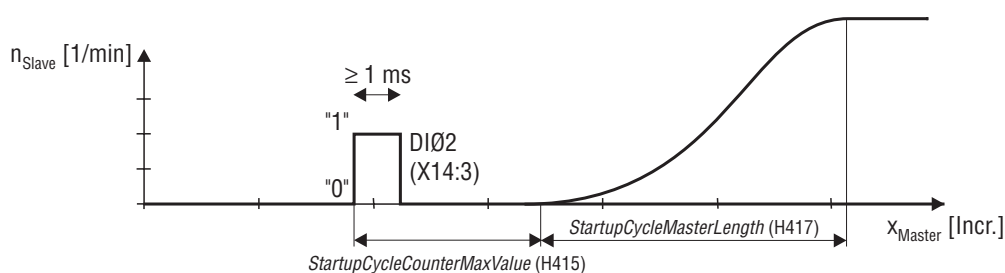
- **StartupCycleMode = 0** : synchronisation manuelle. La synchronisation est lancée dès que le programme attribue la valeur 2 à la variable système **SynchronousState** (H427).
- **StartupCycleMode = 1** : démarrage événementiel de la synchronisation via une entrée binaire. Celle-ci est définie par la variable **StartupCycleInputMask** (H413). Dès que l'entrée binaire en question reçoit un signal "1", la synchronisation est lancée. Le temps de réponse de la borne est de 1 ms.



03784AXX

Fig. 16 : Démarrage événementiel du processus de synchronisation en distance

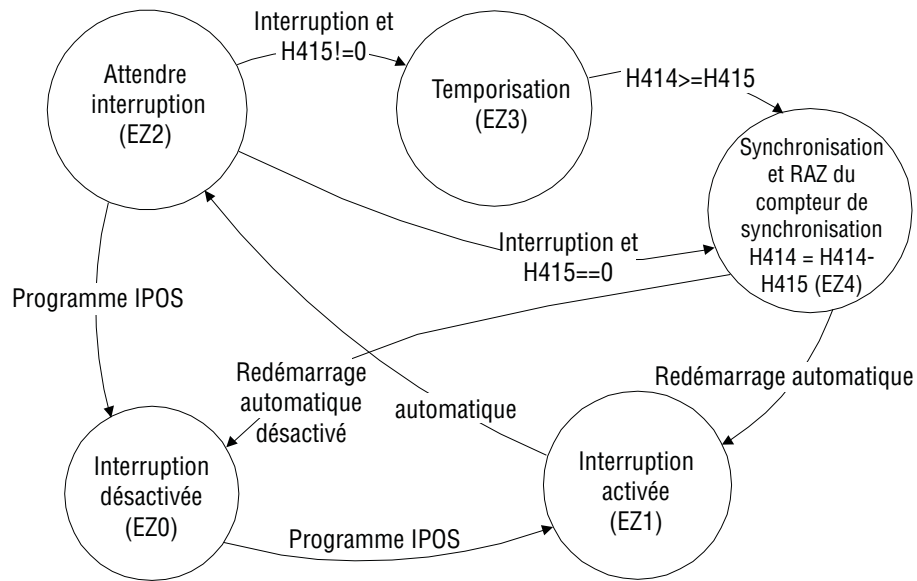
- **StartupCycleMode = 2** : la synchronisation est déclenchée par un front montant à l'entrée binaire DI02 ou sur la piste C (X14:3) (démarrage par interruption). L'entrée binaire DI02 doit être sur "Sans fonction". Il est possible, en combinaison avec la variable **StartupCycleCounterMaxValue** (H415), de programmer une temporisation du cycle du maître avant de démarrer effectivement la synchronisation. La variable **StartupCycleDelayDI02** (H416) permet de prendre en compte le temps de réaction du capteur (1 bit = 0,1 ms). Ce paramètre est également actif en cas de synchronisation au moyen de X14: 3 (piste C).



03785AXX

Fig. 17 : Démarrage par interruption du processus de synchronisation en distance

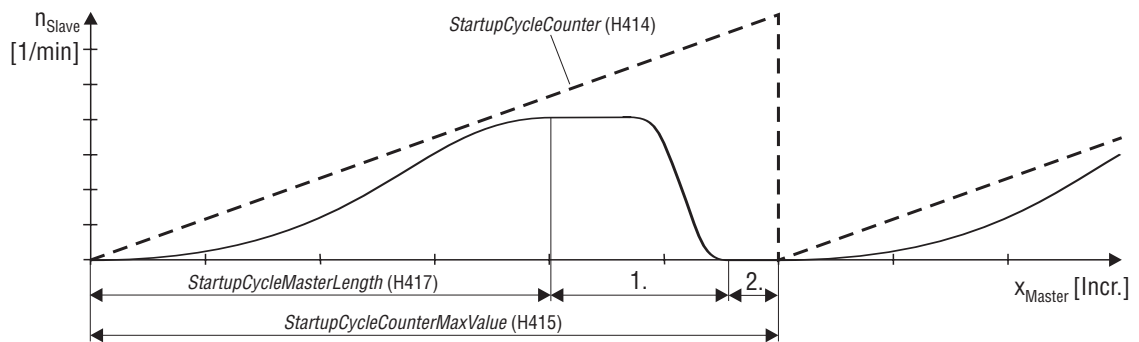
Automate de synchronisation en *StartupCycleState* (H412) :



03407AFR

Fig.18 : Automate de synchronisation avec commande d'interruption (Mode de synchronisation 2)

- **StartupCycleMode = 3** : la synchronisation est initiée par le compteur de positions **StartupCycleCounter** (H414). Si la valeur de la variable **StartupCycleCounter** est supérieure à la valeur maximale définie dans la variable **StartupCycleCounterMaxValue** (H415), la synchronisation est lancée automatiquement. La valeur programmée dans **StartupCycleCounterMaxValue** doit être supérieure à la somme des impulsions des cycles de synchronisation, du maître et de désolidarisation.

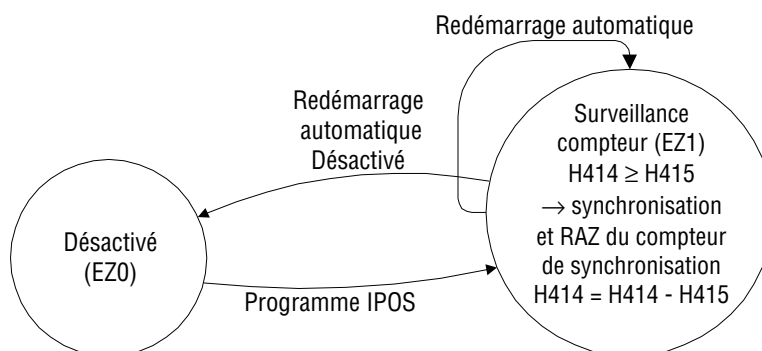


03786AXX

1. Marche synchrone et cycle de désolidarisation
2. Esclave découplé, temps de retour en position initiale

Fig.19 : Démarrage en fonction d'un compteur du processus de synchronisation en distance

Automate de synchronisation en *StartupCycleState* (H412) :



03408AFR

Fig.20 : Automate de synchronisation avec commande en distance (Mode de synchronisation 3)

Variable H411 *StartupCycleModeControl* → fonctions supplémentaires :

Bit	Nom	Description
0	AutoRestart (<i>StartupCycleMode 2 et 3</i>)	= 0 : redémarrage automatique désactivé = 1 : redémarrage automatique activé
1	StartupDisable (<i>StartupCycleMode 2 et 3</i>)	= 0 : synchronisation possible = 1 : synchronisation verrouillée
2	InterruptSelect (<i>StartupCycleMode 2</i>)	= 0 : DI02 = 1 : X14:3 (piste C)
12	StartupMode	= 0 : rattrapage en vitesse/temps = 1 : rattrapage en distance

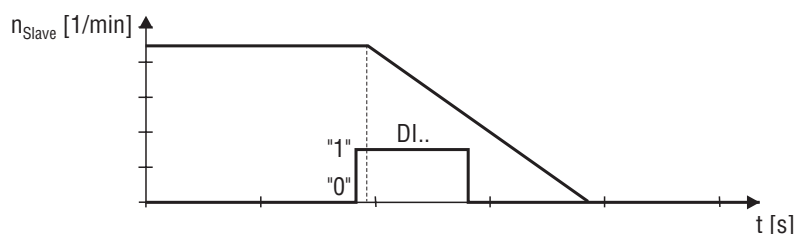
5.4 Automate de désolidarisation

La commande de désolidarisation est active dans les états principaux Z3 et Z4 (→ Fig. 13). La désolidarisation de l'esclave peut se faire de façon manuelle ou automatique. Le mode de désolidarisation est défini dans la variable système **StopCycleMode** (H400). La variable système **StopCycleModeControl** (H401) permet de programmer des fonctions supplémentaires.

Lors de la désolidarisation, en cas de régulation-x, l'entraînement est mené à la vitesse 0 selon la **Rampe t11 (P130)**. En cas de régulation-n, l'entraînement est mené à la vitesse programmée dans la variable système *SpeedFreeMode* (H439).

Variable H400 (*StopCycleMode*) → Mode-désolidarisation :

- **StopCycleMode = 0** : désolidarisation manuelle. L'esclave quitte la marche synchrone avec le maître dès que le programme attribue la valeur 5 à la variable système **SynchronousState** (H427).
- **StopCycleMode = 1** : désolidarisation événementielle via une entrée binaire. Celle-ci est définie par la variable **StartupCycleInputMask** (H413). Dès que l'entrée en question reçoit un signal "1", la désolidarisation est lancée. Le temps de réponse de la borne est de 1 ms.



03855AXX

Fig.21 : Désolidarisation événementielle

Variable H401 *StopCycleModeControl* → fonctions supplémentaires :

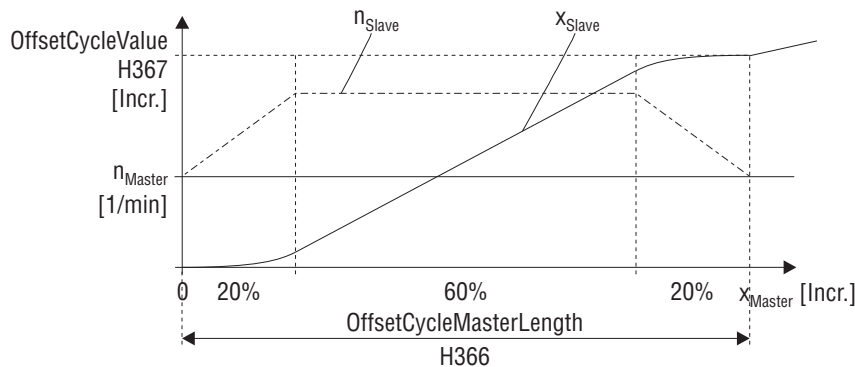
Bit	Nom	Description
0	FreeMode	= 0 : désolidarisation en état principal 0 (régulation-n) = 1 : désolidarisation en état principal 1 (régulation-x)

5.5 Gestion d'offset

5.5.1 Gestion d'offset en vitesse/temps

Dans ce cas précis, le compteur d'écart est augmenté d'une valeur offset (H367, *OffsetCycleValue*). En réduisant le décalage angulaire à zéro (rattrapage en vitesse/temps → chap. 5.3.1), l'esclave effectue un offset.

5.5.2 Gestion d'offset en distance



03412AXX

Fig.22: Courbe de vitesse en cas de gestion d'offset en distance

Dans ce cas précis, un offset se produit au niveau de l'esclave dès que le maître aura effectué un déplacement correspondant à la valeur programmée dans la variable système *OffsetCycleMasterLength* (H366). La valeur de l'offset doit être mémorisée dans la variable système *OffsetCycleValue* (H367).

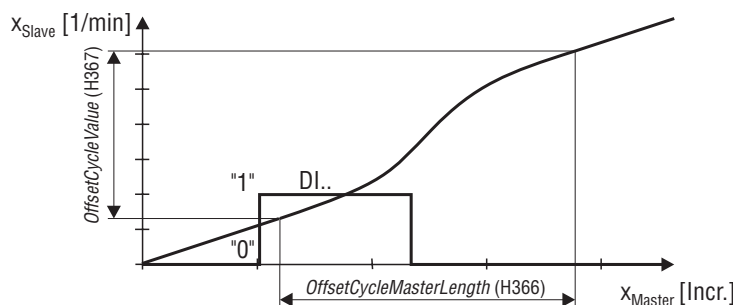
Ce processus ne fonctionne que lorsque la synchronisation interne est en état 3 (synchronisme).

5.5.3 Automate d'offset

La gestion d'offset ne réagit que sur des événements souhaités lorsque la synchronisation interne est en état principal 3 (synchronisme). Le réglage se fait au niveau de la variable système ***OffsetCycleMode*** (H360), la variable système ***OffsetCycleModeContro*** (H361) permettant la programmation de fonctions supplémentaires.

Variable H360 (*OffsetCycleMode*) → Mode offset :

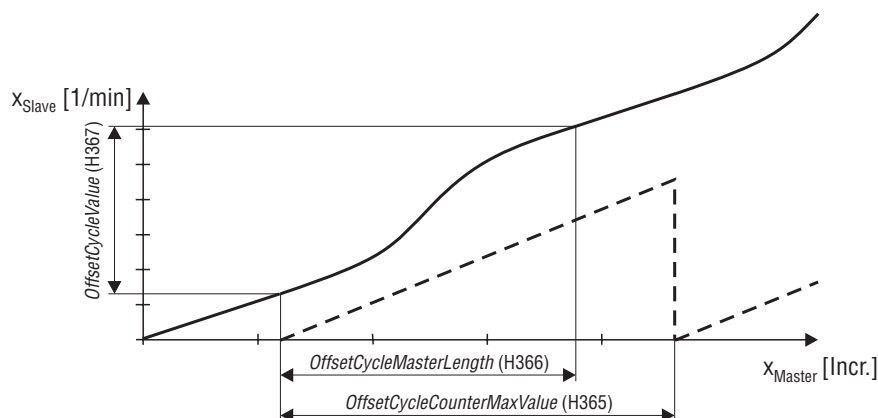
- ***OffsetCycleMode* = 0** : traitement d'offset manuel via le programme-utilisateur IPOS en affectant la valeur 4 à la variable H427 (*SynchronousState*).
- ***OffsetCycleMode* = 1** : traitement d'offset via entrée binaire (signal "1") au moyen de H363 (*OffsetCycleInputMask*) avec une résolution de 1 ms.



03791AXX

Fig. 23 : Gestion d'offset en distance pilotée par entrée binaire

- **OffsetCycleMode = 2** : réservé
- **OffsetCycleMode = 3** : gestion par compteur en combinaison avec les variables H364 (*OffsetCycleCounter*) et H365 (*OffsetCycleCounterMaxValue*), avec report des incréments excédentaires



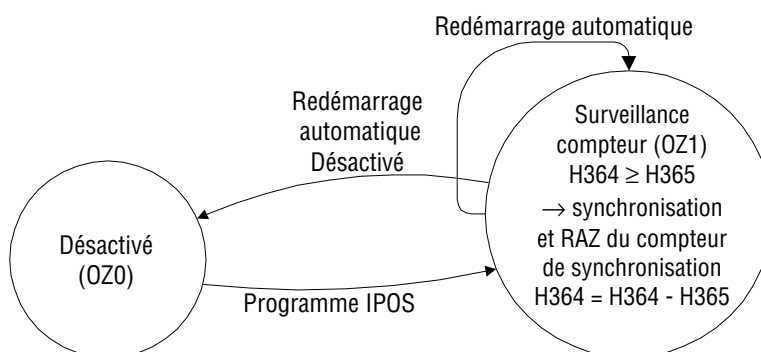
03792AXX

Fig. 24 : Gestion d'offset en distance en fonction d'un compteur

Variable H361 *OffsetCycleModeControl* → fonctions supplémentaires :

Bit	Nom	Description
0	AutoRestart (<i>OffsetCycleMode 3</i>)	= 0 : redémarrage automatique désactivé = 1 : redémarrage automatique activé
1	OffsetDisable (<i>OffsetCycleMode 3</i>)	= 0 : gestion d'offset possible = 1 : gestion d'offset verrouillée
12	OffsetMode	= 0 : gestion d'offset en vitesse/temps = 1 : gestion d'offset en distance

Automate d'offset en *OffsetCycleState* (H362) :



03567AFR

Fig. 25 : Automate d'offset

5.6 Synchronisme

La régulation se fait au moyen d'un régulateur P (P910 "Gain P IPOS"). Les impulsions du maître et de l'esclave sont pondérées et comparées, puis totalisées en valeur 64 bits. Le régulateur P en combinaison avec l'anticipation de vitesse (P228 "Filtre anticipation vitesse DRS") et la vitesse maximale programmée constituent ensemble la consigne de vitesse pour le régulateur de vitesse.

Un dispositif de réglage a été prévu afin d'éviter la perte d'incréments au moment du passage du mode rattrapage en distance en mode synchrone. Lorsque ce dispositif constate une différence d'incréments (H389 → *RegisterLoopOut*) lors de chaque échantillonnage, il ajoute un certain nombre d'incréments (H390 → *RegisterLoopDXDOut*) au compteur d'écart 64 bits. Ce dispositif ne fonctionne qu'en l'état principal Z3 (synchronisme) et peut être programmé directement par l'utilisateur.

Schéma de fonctionnement de la synchronisation interne :

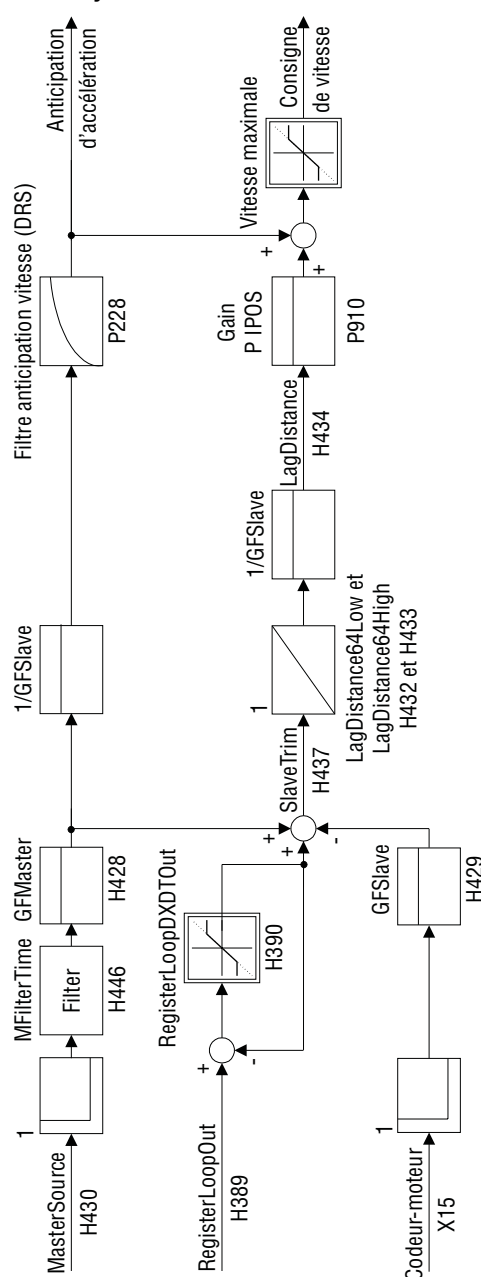


Fig. 26 : Schéma de fonctionnement de la synchronisation interne

03413ADE

SEW
USOGOME

5.7 Codeur virtuel

5.7.1 Codeur virtuel sans générateur de rampe

Le modèle le plus simple d'un codeur virtuel est la variable IPOS **MasterTrimX14** (H442). Lorsque le codeur physique est activé (H430 = 0), l'attribution de la valeur k à la variable **MasterTrimX14** a pour effet d'ajouter ou de soustraire (en fonction du signe) k impulsions physiques aux impulsions du codeur-source.



Important : en cas d'utilisation du codeur virtuel, il n'est plus possible d'utiliser X14 comme simulation codeur.

5.7.2 Codeur virtuel avec générateur de rampe

Le codeur virtuel (→ variables IPOS H370 – H377) est un compteur logiciel qui peut servir de codeur-source pour la synchronisation (attribuer la **valeur 376 à H430 MasterSource**). Ce compteur logiciel peut être transféré du "générateur" du codeur virtuel à d'autres axes MOVIDRIVE® au moyen d'une liaison par Sbus (instruction SCOM). Ce système nécessite la synchronisation par Sbus de tous les appareils par transmission de l'ID synchronisation (P817) (toutes les 5 ms).

Le codeur virtuel fonctionne en cycles de 1 ms et est exploité indépendamment de l'état actuel du synchronisme. Il génère un profil de déplacement qui dépend de la vitesse de déplacement (H373) et de la rampe programmée (H377). Le codeur virtuel est lancé en attribuant à la consigne de position (H375) une valeur différente de la position actuelle (H376). Lorsque la valeur *VEncoderXActual* (H374) atteint la valeur programmée dans *VEncoderXSetpoint* (H375), le codeur virtuel est arrêté (*VEncoderMode* = 0).

H375 = [VEncoderXSetpoint]	= 1 "Impulsions maître"	Consigne de position
H376 = [VEncoderXActual]	= 1 "Impulsions maître"	Position actuelle
H373 = [VEncoderNSetpoint]	= 1 "Impulsions maître"/ms	Vitesse de consigne
H374 = [VEncoderNActual]	= 1 "Impulsions maître"/ms	Vitesse réelle
H377 = [VEncoderNdT]	= 1 "Impulsions maître"/ms ²	Accélération

Choix du mode de fonctionnement du codeur au moyen de la variable H370 (*VEncoderMode*) :

- *VEncoderMode* = 0 : mode normal (selon le profil de déplacement défini par les variables H373, H375 et H377).
- *VEncoderMode* = 1 : réservé
- *VEncoderMode* = 2 : compteur sans fin avec vitesse de déplacement *VEncoderNSetpoint* et rampe *VEncoderNdT*.

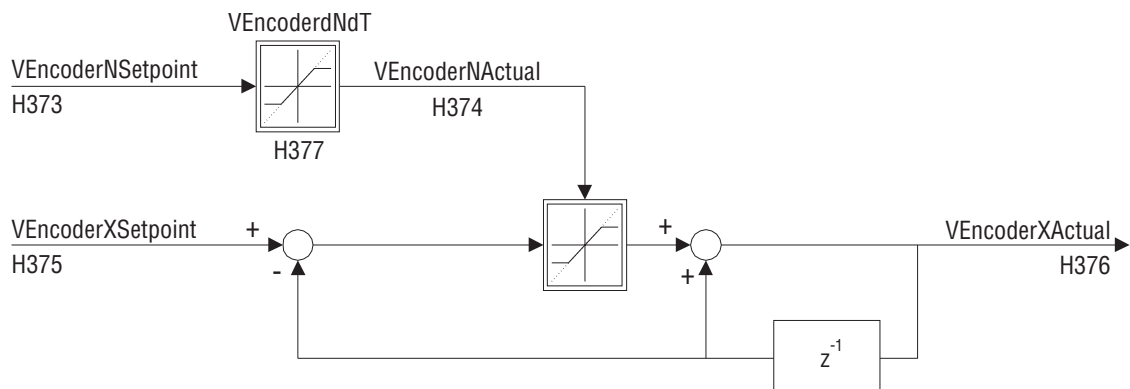


Fig. 27 : Structure du codeur virtuel avec générateur de rampe

03529ADE

Variable H371 (*VEncoderModeControl*) :

Bit	Nom	Valeur 0	Valeur 1
0	AxisStop	Arrêt de l'axe désactivé	Après l'apparition d'un défaut, la valeur de <i>VEncoderNSetpoint</i> (H373) est mis sur 0 (arrêt de l'axe virtuel).

5.8 Remarques importantes

- La possibilité pour le maître de définir un déplacement avec le signe dans la variable H417 (*StartupCycleMasterLength*) ou H366 (*OffsetCycleMasterLength*) implique qu'il faille tenir compte du sens de marche du maître. En outre, il convient de tenir compte du fait que le facteur de réduction en variable H428 (*GFMaster*) puisse également être défini avec le signe.
- L'erreur de poursuite (**P923 "Tolérance erreur de poursuite"**) est uniquement déclenchée en l'état principal Z3 (synchronisme).
- Le compteur 64 bits peut être remis à zéro en programmant les bornes sur "**DRS CALAGE**". Cela a pour effet de désolidariser l'esclave du maître lorsque l'entraînement se trouve en l'état principal Z3 (synchronisme).
- Pour obtenir des résultats précis en cas de rattrapage en distance, la variable H390 (*Register-LoopDXDToOut*) devrait contenir une valeur différente de zéro pour permettre d'éliminer les incréments excédentaires.
- **P910 "Gain P IPOS"** : la valeur de ce paramètre est optimisée lors de la mise en service avec MOVITOOLS.
- **P228 "Filtre anticipation vit. (DRS)"** : filtre de consigne pour l'anticipation de la synchronisation interne. Réglage-usine = 0 = filtre non actif ; réglage conseillé = 10 ms.
- L'interpolation des impulsions d'entrée venant du maître est assurée par la variable **MFilterTime** (H446). En augmentant la valeur de cette variable, l'on modifie la pondération des impulsions envoyées par le maître. Une correction peut être obtenue en multipliant par exemple la variable **GSlave** (H429) avec la variable **MFilterTime** (H446).
- Facteur de réduction absolu du maître = **GFMaster** (H428) x **MFilterTime** (H446). Important : le résultat ne doit pas dépasser 32767.

6 Variables système de la synchronisation interne

Variable	Nom et plage de valeur	Etat	Description
Pilotage d'offset			
H360	OffsetCycleMode 0 à 3	R/W	Offsetmode = 0 : Offset via programme IPOS = 1 : Offset via les bornes d'entrée = 2 : réservé = 3 : Offset par compteur
H361	OffsetCycleModeControl	R/W	Activation de différentes fonctions Bit 0 : AutoRestart (en mode 3) = 0 : redémarrage automatique désactivé = 1 : redémarrage automatique activé Bit 1 : OffsetDisable (en mode 3) = 0 : gestion d'offset possible = 1 : gestion d'offset verrouillée Bit 12 : OffsetMode = 0 : gestion d'offset en vitesse/temps = 1 : gestion d'offset en distance
H362	OffsetCycleState 0 à 1 maxi (en fonction du OffsetCycleMode)	R/W	Pilotage des différents modes
H363	OffsetCycleInputMask	R/W	Masque des bornes (identique à H483 "InputLevel")
H364	OffsetCycleCounter	R/W	Compteur maître pour la gestion d'offset
H365	OffsetCycleCounterMaxValue	R/W	En mode 3 : longueur maxi pour l'offset automatique
H366	OffsetCycleMasterLength	R/W	Déplacement défini pour le maître lors de la gestion d'offset
H367	OffsetCycleValue	R/W	Valeur de l'offset pour l'esclave
Codeur virtuel			
H370	VEncoderMode 0 à 2	R/W	Mode de fonctionnement du codeur virtuel = 0 : mode normal = 1 : réservé = 2 : compteur sans fin
H371	VEncoderModeControl	R/W	Bit 0 : AxisStop = 0 : désactivé = 1 : arrêt de l'axe en cas de défaut de l'appareil
H372	VencoderState	R/W	Sans fonction
H373	VencoderNSetpoint	R/W	Consigne de vitesse de déplacement en 1 incr./ms
H374	VencoderNActual	R/W	Vitesse réelle de déplacement en 1 incr./ms
H375	VencoderXSetpoint	R/W	Consigne de position en incr.
H376	VencoderXActual	R/W	Position actuelle en incr.
H377	VencoderdNdT	R/W	Accélération (rampe) en 1 incr./ms ²
Dispositif de réglage			
H389	RegisterLoopOut	R/W	La valeur à réduire en rapport avec RegisterLoopDXDToOut
H390	RegisterLoopDXDToOut	R/W	Valeur maxi du dispositif de réglage Addition maxi (compteur 64 bits) par ms

Variable	Nom et plage de valeur	Etat	Description
Commande de désolidarisation			
H400	StopCycleMode 0 à 1	R/W	Mode de désolidarisation = 0 : désolidarisation via programme IPOS = 1 : désolidarisation via les bornes d'entrée
H401	StopCycleModeControl	R/W	Activation de différentes fonctions Bit 0 : FreeMode = 0 : désolidarisation en état principal 0 (régulation-n) = 1 : désolidarisation en état principal 1 (régulation-x)
H402	StopCycleState		Sans fonction
H403	StopCycleInputMask	R/W	Masque des bornes (identique à H483 "InputLevel")
Commande de synchronisation			
H410	StartupCycleMode 0 à 3	R/W	Mode de synchronisation = 0 : synchronisation via programme IPOS = 1 : synchronisation via les bornes d'entrée = 2 : synchronisation via interruption = 3 : synchronisation par compteur
H411	StartupCycleModeControl	R/W	Activation de différentes fonctions Bit 0 : AutoRestart (en mode 2 et 3) = 0 : redémarrage automatique désactivé = 1 : redémarrage automatique Bit 1 : StartupDisable (en mode 2 et 3) = 0 : synchronisation possible = 1 : synchronisation verrouillée Bit 2 : InterruptSelect (en mode 2) = 0 : D102 = 1 : Voie X14C Bit 12 : StartupMode = 0 : rattrapage en vitesse/temps = 1 : rattrapage en distance
H412	StartupCycleState 0 à 3 maxi (en fonction du mode sélectionné)	R/W	Gestion des différents modes
H413	StartupCycleInputMask	R/W	Masque des bornes d'entrée (identique à H483 "InputLevel")
H414	StartupCycleCounter	R/W	Compteur maître pour le couplage
H415	StartupCycleCounterMaxValue	R/W	En mode 2 : temporisation pour la synchronisation En mode 3 : longueur maxi pour la synchronisation automatique
H416	StartupCycleDelayDI02 -32768 à 32767	R/W	Temps mort (en unité de 0,1 ms) Temporisation du capteur raccordé à l'entrée Touch-Probe 2
H417	StartupCycleMasterLength	R/W	Déplacement défini pour le maître lors de la synchronisation en distance

Variable	Nom et plage de valeur	Etat	Description
Variables diverses			
H425	SynchronousMode		Sans fonction
H426	SynchronousModeControl	R/W	Activation de différentes fonctions Bit 0 : PosTrim (uniquement active en état Z1 "Régulation-x") = 0 : activé = 1 : déplacement vers <i>TargetPos</i> (H492) Bit 1 : LagError (en état 3 → synchronisme) = 0 : surveillance d'erreurs de poursuite = 1 : pas de surveillance d'erreurs de poursuite
H427	SynchronousState 0 à 5	R/W	Automate principal de la synchronisation interne = 0 : désolidarisé en régulation-n = 1 : désolidarisé en régulation-x = 2 : synchronisation = 3 : synchronisme = 4 : gestion d'offset = 5 : désolidarisation
H428	GFMaster -32768 à 32767	R/W	Facteur de pondération des incréments du maître, valeur = i_S
H429	GFSlave 1 à 32767	R/W	Facteur de pondération des incréments de l'esclave, valeur = i_M
H430	MasterSource 0 à 511	R/W	Source des incréments du maître = 0 : X14 + axe virtuel (H442) > 0 : pointeur sur variable
H431	Reserved1		
H432	LagDistance64Low	R/-	32 bits inférieurs du compteur 64 bits
H433	LagDistance64High	R/-	32 bits supérieurs du compteur 64 bits
H434	LagDistance32	R/-	Ecart de poursuite 32 bits rapporté à GFSlave
H435	Reserved2		
H436	Reserved3		
H437	SlaveTrim	R/W	Consigne de position actuelle de l'esclave
H438	XMasterPos	R/-	Valeur d'affichage du compteur maître pour la synchronisation et la gestion d'offset
H439	SpeedFreeMode	R/W	Consigne de vitesse en mode désolidarisé avec régulation-n en 0,2 r/min
H440	Reserved4		
H441	Reserved5		
H442	MasterTrimX14 -32768 à 32767	R/W	Axe virtuel Nombre d'impulsions 1 incr./ms
H443	Reserved6		
H444	ReSprintClose 0 à 2	R/W	Verrouillage sens de marche = 0 : marche possible dans les deux sens = 1 : uniquement marche à gauche = 2 : uniquement marche à droite
H445	Reserved7		
H446	MFilterTime 1 à 30	R/W	Temps d'interpolation en ms = 1 : sans filtre ≤ 30 : échelle haute valeur facteur de pondération absolu des impulsions du maître = GFMaster x MFilterTime

7 Exemples de programmes IPOS

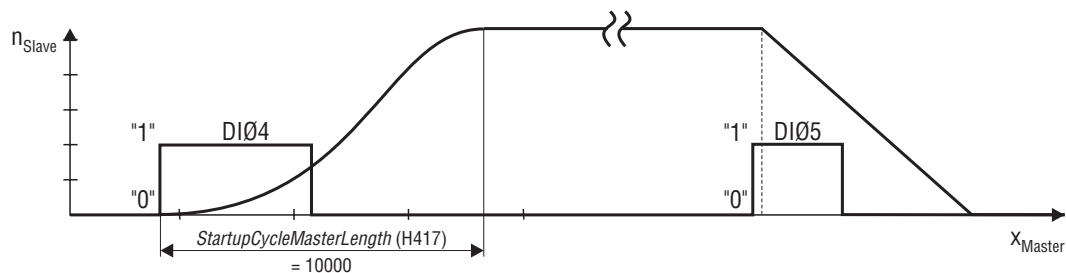
7.1 Exemple 1

Enoncé :

Le programme doit faire fonctionner un esclave en synchronisme de position angulaire par rapport à un maître. On utilisera des réducteurs identiques, le rapport de réduction étant 1:1. Le maître et l'esclave sont reliés entre eux par la liaison X14. L'esclave est piloté par les entrées binaires. La commande de synchronisation et de désolidarisation se fera au moyen des entrées binaires X13:5 (DI04) et X13:6 (DI05). Ces deux entrées doivent être programmées sur "SANS FONCTION".

- Signal "1" à la borne DI04 → lancement de la synchronisation ; elle se fera par rattrapage en distance et devra être terminée après 10.000 incréments du maître.
- Signal "1" à la borne DI05 → lancement de la désolidarisation.

Les variables IPOS nécessaires au programme sont programmées dans la fonction d'initialisation.



03865AXX

Fig. 28 : Synchronisation et désolidarisation événementielles

Programme IPOS

```

/*=====
      IPOS-Sourcefile
      for Synchronous Drive Control
      -----
      SEW-Eurodrive GmbH & Co.
      Ernst-Blickle-Str. 42
      D-76646 Bruchsal

      sew@sew-eurodrive.de
      http://www.SEW-EURODRIVE.de
      =====*/
#pragma var      300  309
#pragma globals  310  349

#include <const.h>
#include <Example01.h>          // Fichier en-tête
                                // avec définition des variables
                                // et fonction d'initialisation

/*=====
      Mainfunction (IPOS-Startfunction)
      =====*/
main()
{
    /*-----
      Startup
      -----*/
    InitSynchronization();      // Appel de la fonction d'initialisation

    /*-----
      Mainprogram loop
      -----*/
    while(1)
    {
    }
}

```

Fichier en-tête avec définition des variables

```

/*****
Example01.h
Data- and Startup headerfile for IPOS+ Compiler.
For Startup after Power On call "InitSynchronization();"
Datafile Movidrive Synchronous Drive Control Version 1.0
*****/

#define SynchronousMode          H425
#define SynchronousModeControl  H426
#define SynchronousState        H427
#define GFMaster                 H428
#define GFSlave                  H429
#define MasterSource             H430
#define Reserved1                H431
#define LagDistance64Low         H432
#define LagDistance64High       H433
#define LagDistance32            H434
#define Reserved2                H435
#define Reserved3                H436
#define SlaveTrim                H437
#define XMasterPos               H438
#define SpeedFreeMode            H439
#define Reserved4                H440
#define Reserved5                H441
#define MasterTrimX14            H442
#define Reserved6                H443

```

```

#define ReSprintClose          H444
#define Reserved7              H445
#define MFilterTime            H446

// variables for StartupCycle, StopCycle and OffsetCycle
#define StopCycleMode          H400
#define StopCycleModeControl   H401
#define StopCycleState         H402
#define StopCycleInputMask     H403

#define StartupCycleMode       H410
#define StartupCycleModeControl H411
#define StartupCycleState      H412
#define StartupCycleInputMask  H413
#define StartupCycleCounter     H414
#define StartupCycleCounterMaxValue H415
#define StartupCycleDelayDI02   H416
#define StartupCycleMasterLength H417

#define OffsetCycleMode        H360
#define OffsetCycleModeControl H361
#define OffsetCycleState       H362
#define OffsetCycleInputMask   H363
#define OffsetCycleCounter     H364
#define OffsetCycleCounterMaxValue H365
#define OffsetCycleMasterLength H366
#define OffsetCycleValue       H367

// variables to Register Control
#define RegisterLoopOut        H389
#define RegisterLoopDXDTOut    H390

// variables for Virtual Encoder
#define VEncoderMode           H370
#define VEncoderModeControl    H371
#define VEncoderState          H372
#define VEncoderNSetpoint      H373
#define VEncoderNActual        H374
#define VEncoderXSetpoint      H375
#define VEncoderXActual        H376
#define VEncoderdNdT           H377

//Startupdata from: 08.08.2000 - 16:35:22
InitSynchronization()
{
    for (H0=128; H0<=457; H0++)          // RAZ des variables supérieures à H128
    {
        *H0=0;
    }
    _Memorize(MEM_LDDATA);
    _Wait(100);

    GFMaster          = 1;          // Pondération des incréments du maître
    GFSlave           = 1;          // Pondération des incréments de l'esclave
    MFilterTime       = 1;          // Traitement des incr. du maître sans filtre
    StartupCycleMode   = 1;          // Mode synchro 1 : lancement événementiel
                                   // de la synchronisation via entrée binaire
    StartupCycleInputMask = 16;     // Choix de la borne DI04 pour la synchronisation
    StartupCycleMasterLength = 10000; // Nbre incr. maître jusqu'à fin synchronisation
    _BitSet(StartupCycleModeControl, 12); // Choix du rattrapage en distance
    RegisterLoopDXDTOut = 2;        // Limitation dispositif de correction
    StopCycleMode      = 1;          // Mode désolidarisation 1 : lancement événementiel
                                   // de la désolidarisation via entrée binaire
    StopCycleInputMask = 32;        // Choix de la borne DI05 pour la désolidari-
    sation
}

```

7.2 Exemple 2

Enoncé :

Ce programme est prévu pour la coupe à la volée de matériau sans fin au moyen d'une scie volante. Les incréments définissant le déplacement du matériau sans fin servent d'incrément maître et sont appliqués à l'entrée X14 de l'esclave qui pilote le chariot de la scie. L'esclave attend en position initiale (Start). La synchronisation est lancée par le compteur de position *StartupCycleCounter* (H414). La coupe du matériau se fait au cours de la marche synchrone du maître et de l'esclave. Lorsque la coupe est terminée, l'esclave se désolidarise du maître et reprend sa position initiale. Le rapport de réduction est 1:1.

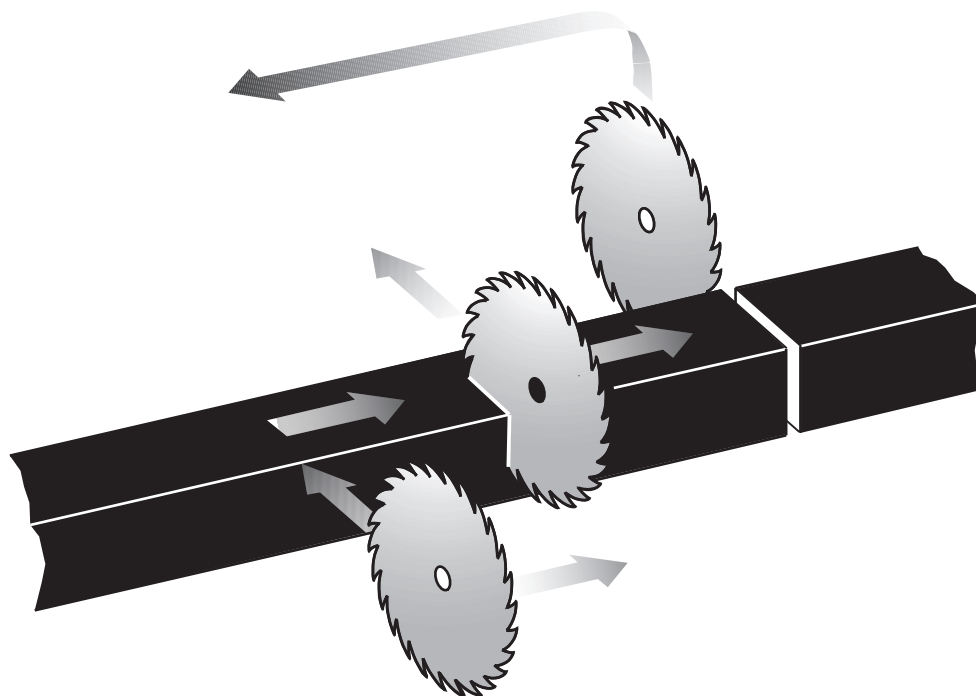
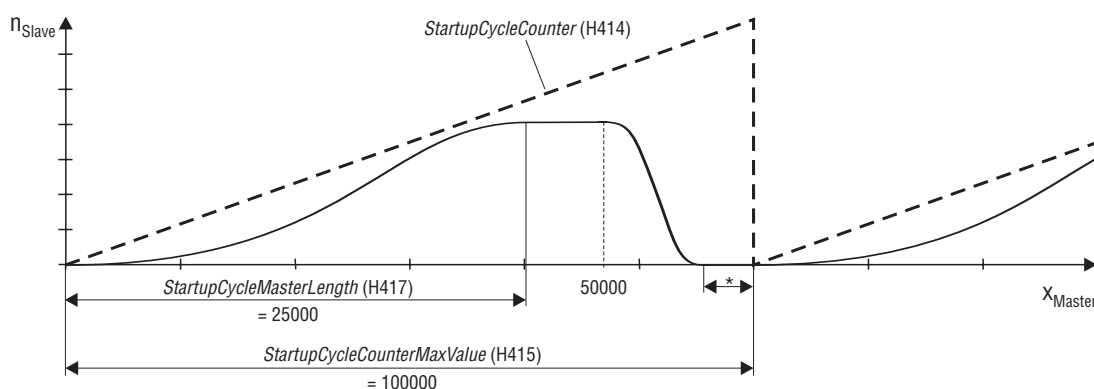


Fig. 29 : Coupe à la volée

03866AXX



* L'esclave est désolidarisé

Fig. 30 : Lancement de la synchronisation par compteur de position

03867AXX

Remarques importantes :

- La prise de référence se fera selon le type de prise de référence 3 (P903)
- La correction point 0 (P900) est programmée à 300.000
- Les FdC droite et gauche doivent être raccordés et paramétrés



SEW
USOCOME

Programme IPOS

```

/*=====
      IPOS-Sourcefile
      for Synchronous Drive Control
      -----
      SEW-Eurodrive GmbH & Co.
      Ernst-Blickle-Str. 42
      D-76646 Bruchsal

      sew@sew-eurodrive.de
      http://www.SEW-EURODRIVE.de
      =====*/
#pragma var      300  309
#pragma globals  310  349

#include <const.h>
#include <io.h>
#include <Example02.h>                                // Fichier en-tête
                                                    // avec définition des variables
                                                    // et fonction d'initialisation

#define LINEAR      0                                // Positionnement avec rampe linéaire
#define SYNCHRONLAUF 6                                // Synchronisation interne

#define Halt        _BitSet(ControlWord, 2)          // Bit Droite forcé à 1
#define Freigabe    _BitClear(ControlWord, 2)        // Bit Droite remis à 0

long Rampenform, tmp;

/*=====
      Mainfunction (IPOS-Startfunction)
      =====*/
main()
{
    /*-----
      Startup
      -----*/
    InitSynchronization();                            // Appel de la fonction d'initialisation

    Rampenform=LINEAR;                                // Rampe de positionnement
    _SetSys(SS_RAMPTYPE, Rampenform);

    while (!DI00);                                    // Attendre signal High sur DI00 "/Verrouillage"

    _Go0(GO0_C_W_ZP);                                  // Prise de référence selon type 3 / FdC Droite
                                                    // P900 "Correction point 0" : 300.000 incr.
    _GoAbs(GO_WAIT, 0);                                // Déplacement vers la position initiale

    Rampenform=SYNCHRONLAUF;                            // Activer synchronisation interne
    _SetSys(SS_RAMPTYPE, Rampenform);

    StartupCycleCounter = 0;                            // RAZ compteur
    StartupCycleState = 1;                              // Activer la commande de synchronisation

    /*-----
      Mainprogram loop
      -----*/
    while(1)
    {
        tmp=StartupCycleCounter;                        // Mémorisation temporaire compteur de synchronisation
        if ((tmp>50000)&&(SynchronousState==3))          // Commutation forme de rampe lorsque
                                                    // compteur > 50.000 incr. maître et
                                                    // entraînement en synchronisme
        {
            Halt;                                        // Verrouillage entraînement
            SynchronousState=5;                          // Désolidarisation (en régulation de position)
            Rampenform=LINEAR;                            // Rampe de positionnement
            _SetSys(SS_RAMPTYPE, Rampenform);
        }
    }
}

```

```

        Freigabe;                                // Moteur en marche
        _GoAbs(GO_WAIT, 0);                       // Déplacement vers position départ
        Halt;                                     // Verrouillage moteur
        Rampenform=SYNCHRONLAUF;                 // Activer synchronisation interne
        _SetSys(SS_RAMPTYPE, Rampenform);
        Freigabe;                                // Moteur en marche
    }
}
}

```

Fichier en-tête avec définition des variables

```

/*****
Example02.h
Data- and Startup headerfile for IP0S+ Compiler.
For Startup after Power On call "InitSynchronization();"
Datafile Movidrive Synchronous Drive Control Version 1.0
*****/

#define SynchronousMode           H425
#define SynchronousModeControl   H426
#define SynchronousState         H427
#define GFMaster                  H428
#define GFSlave                   H429
#define MasterSource              H430
#define Reserved1                 H431
#define LagDistance64Low          H432
#define LagDistance64High        H433
#define LagDistance32            H434
#define Reserved2                 H435
#define Reserved3                 H436
#define SlaveTrim                 H437
#define XMasterPos                H438
#define SpeedFreeMode             H439
#define Reserved4                 H440
#define Reserved5                 H441
#define MasterTrimX14             H442
#define Reserved6                 H443
#define ReSprintClose             H444
#define Reserved7                 H445
#define MFilterTime               H446

// variables for StartupCycle, StopCycle and OffsetCycle
#define StopCycleMode             H400
#define StopCycleModeControl      H401
#define StopCycleState            H402
#define StopCycleInputMask        H403

#define StartupCycleMode          H410
#define StartupCycleModeControl   H411
#define StartupCycleState         H412
#define StartupCycleInputMask     H413
#define StartupCycleCounter        H414
#define StartupCycleCounterMaxValue H415
#define StartupCycleDelayDI02     H416
#define StartupCycleMasterLength  H417

#define OffsetCycleMode           H360
#define OffsetCycleModeControl    H361
#define OffsetCycleState          H362
#define OffsetCycleInputMask      H363
#define OffsetCycleCounter        H364
#define OffsetCycleCounterMaxValue H365
#define OffsetCycleMasterLength   H366
#define OffsetCycleValue          H367

// variables to Register Control
#define RegisterLoopOut            H389

```

```

#define RegisterLoopDXDTOut      H390

// variables for Virtual Encoder
#define VEncoderMode             H370
#define VEncoderModeControl      H371
#define VEncoderState            H372
#define VEncoderNSetpoint        H373
#define VEncoderNActual          H374
#define VEncoderXSetpoint        H375
#define VEncoderXActual          H376
#define VEncoderdNdT             H377

//Startupdata from: 08.08.2000 - 15:54:37
InitSynchronization()
{
    for (H0=128; H0<=457; H0++)          // RAZ des variables supérieures à H128
    {
        *H0=0;
    }
    _Memorize(MEM_LDDATA);
    _Wait(100);

    GFMaster          = 1;          // Pondération des incréments du maître
    GFSlave           = 1;          // Pondération des incréments de l'esclave
    MFilterTime       = 1;          // Traitement des incr. du maître sans filtre
    StartupCycleMode   = 3;          // Mode synchro 3 : lancement synchronisation par
                                     // dépassement valeur maxi du compteur
    _BitSet(StartupCycleModeControl, 0); // Activation AutoRestart couplage
    StartupCycleCounterMaxValue = 100000; // Valeur maxi compteur de couplage
    StartupCycleMasterLength = 25000; // Nbre incr. maître jusqu'à fin synchronisation
    _BitSet(StartupCycleModeControl, 12); // Choix rattrapage en distance
    RegisterLoopDXDTOut = 2;          // Limitation dispositif de correction
    _BitSet(StopCycleModeControl, 0); // Désolidarisation en état principal 1 (régulation-x)
    _BitSet(SynchronousModeControl, 0); // Activation "Déplacement vers TargetPos (H492)"
    _BitSet(SynchronousModeControl, 1); // Pas de surveillance d'erreurs de poursuite
}

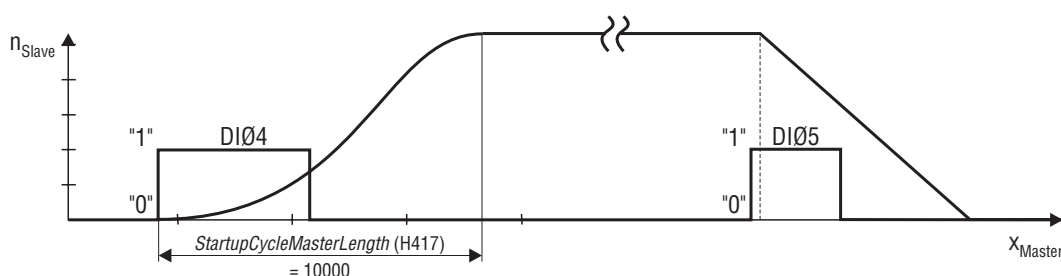
```

7.3 Exemple 3

Enoncé :

Le programme doit faire fonctionner un esclave en synchronisme de position angulaire par rapport à un maître. On utilisera des réducteurs identiques, le rapport de réduction étant 1:1. Le maître et l'esclave sont reliés entre eux par Sbus. L'esclave est piloté par les entrées binaires. La commande de synchronisation et de désolidarisation se fera au moyen des entrées binaires X13:5 (DI04) et X13:6 (DI05). Ces deux entrées doivent être programmées sur "SANS FONCTION".

- Signal "1" à la borne DI04 → lancement de la synchronisation ; elle se fera par rattrapage en distance et devra être terminée après 10.000 incréments du maître.
- Signal "1" à la borne DI05 → lancement de la désolidarisation.



03868AXX

Fig. 31 : Synchronisation et désolidarisation événementielles

Les variables IPOS nécessaires au programme sont programmées dans la fonction d'initialisation. Deux objets d'envoi (Position maître H511 et Sbus ID synchronisation) sont créés dans le programme principal du variateur maître et communiqués au Sbus lorsque la transmission cyclique des données est lancée.

Dans le programme principal de l'esclave, il convient de créer un objet pour la réception de la position du maître communiquée par le Sbus et de lancer la transmission cyclique.

Le maître et l'esclave doivent avoir des adresses Sbus (P813) différentes.

Observer les réglages suivants au niveau du maître :

- Le numéro de l'objet d'envoi "Sbus ID synchronisation" doit être différent de la valeur du paramètre P817.
- Le temps de cycle ("Cycletime") dans l'instruction SCOM pour l'ID synchronisation doit être 5 ms.
- Le temps de cycle ("Cycletime") dans l'instruction SCOM pour la position du maître doit être 1 ms.

Observer les réglages suivants au niveau de l'esclave :

- La valeur du paramètre P817 doit être identique au numéro de l'objet d'envoi "Sbus ID synchronisation".
- La variable système H430 *MasterSource* doit être identique à la valeur du D-pointeur (→ structure de l'instruction SCOM).

Programme IPOS pour le maître

```

/*=====
    Fichier source IPOS
=====*/
#include <const.h>

    SCTRCYCL Position;          // Structure standard SEW pour l'instruction _SBusCommDef
    SCTRCYCL SynchID;

/*=====
    Fonction principale (Fonction d'entrée IPOS)
=====*/
main()
{

    /*-----
        Initialisation
        -----*/

    SynchID.ObjectNo=1090;      // Description de la structure standard SEW :
    SynchID.CycleTime=5;       // Communication de l'objet n° 1090 (télégramme de
    SynchID.Offset=0;          // synchronisation à envoyer) au Sbus (temps de cycle 5 ms)
    SynchID.Format=0;
    SynchID.DPointer=0;
    SynchID.Result=0;

    Position.ObjectNo=1100;     // Description de la structure standard SEW :
    Position.CycleTime=1;      // Communication de l'objet n° 1100 (position maître
32 bits/H511 à envoyer)
    Position.Offset=0;         // au Sbus (temps de cycle 1 ms, format MOTOROLA)
    Position.Format=4;
    Position.DPointer=511;
    Position.Result=0;

    _SBusCommDef(SCD_TRCYCL, SynchID); // Création des objets d'envoi pour la transmission
    _SBusCommDef(SCD_TRCYCL, Position); // cyclique via Sbus

    _SBusCommOn();             // Initialisation des objets d'envoi et lancement de
                                // la transmission cyclique via Sbus

    /*-----
        Boucle principale
        -----*/
    while(1)
    {
    }
}

```

Programme IPOS pour l'esclave

```

/*=====
      IPOS-Sourcefile
      for Synchronous Drive Control
      -----
      SEW-Eurodrive GmbH & Co.
      Ernst-Blickle-Str. 42
      D-76646 Bruchsal

      sew@sew-eurodrive.de
      http://www.SEW-EURODRIVE.de
=====*/
#pragma var      300  309
#pragma globals  310  349

#include <const.h>
#include <Example03.h>          // Fichier en-tête
                                // avec description des variables
                                // et fonction d'initialisation

SCREC Position;                // Structure standard SEW pour l'instruction

/*=====
      Mainfunction (IPOS-Startfunction)
=====*/
main()
{
    /*-----
      Startup
      -----*/
    InitSynchronization();      // Appel de la fonction d'initialisation

    Position.ObjectNo=1100;      // Description de la structure standard SEW :
    Position.Format=4;           // Affectation de l'objet n° 1100 (position maître 32
bits à recevoir)
    Position.DPointer=200;       // à la variable H200

    _SBusCommDef(SCD_REC, Position); // Création de l'objet de réception pour la
                                // transmission cyclique par Sbus

    _SBusCommOn();              // Initialisation de l'objet de réception et lancement
                                // de la transmission cyclique via Sbus

    /*-----
      Mainprogram loop
      -----*/
    while(1)
    {

    }

}

```

Fichier en-tête avec définition des variables

```

/*****
Example03.h
Data- and Startup headerfile for IPOS+ Compiler.
For Startup after Power On call "InitSynchronization();"
Datafile Movidrive Synchronous Drive Control Version 1.0
*****/

#define SynchronousMode           H425
#define SynchronousModeControl   H426
#define SynchronousState         H427
#define GFMaster                  H428
#define GFSlave                   H429
#define MasterSource              H430
#define Reserved1                 H431
#define LagDistance64Low          H432
#define LagDistance64High        H433
#define LagDistance32             H434
#define Reserved2                 H435
#define Reserved3                 H436
#define SlaveTrim                 H437
#define XMasterPos                H438
#define SpeedFreeMode             H439
#define Reserved4                 H440
#define Reserved5                 H441
#define MasterTrimX14             H442
#define Reserved6                 H443
#define ReSprintClose             H444
#define Reserved7                 H445
#define MFilterTime               H446

// variables for StartupCycle, StopCycle and OffsetCycle
#define StopCycleMode             H400
#define StopCycleModeControl      H401
#define StopCycleState            H402
#define StopCycleInputMask        H403

#define StartupCycleMode           H410
#define StartupCycleModeControl   H411
#define StartupCycleState         H412
#define StartupCycleInputMask     H413
#define StartupCycleCounter        H414
#define StartupCycleCounterMaxValue H415
#define StartupCycleDelayDI02     H416
#define StartupCycleMasterLength  H417

#define OffsetCycleMode           H360
#define OffsetCycleModeControl    H361
#define OffsetCycleState          H362
#define OffsetCycleInputMask      H363
#define OffsetCycleCounter        H364
#define OffsetCycleCounterMaxValue H365
#define OffsetCycleMasterLength   H366
#define OffsetCycleValue          H367

// variables to Register Control
#define RegisterLoopOut            H389
#define RegisterLoopDXDTOut       H390

// variables for Virtual Encoder
#define VEncoderMode               H370
#define VEncoderModeControl        H371
#define VEncoderState              H372
#define VEncoderNSetpoint          H373
#define VEncoderNActual            H374
#define VEncoderXSetpoint          H375
#define VEncoderXActual            H376
#define VEncoderdNdT               H377

```

```

//Startupdata from: 08.08.2000 - 16:14:58
InitSynchronization()
{
    for (H0=128; H0<=457; H0++)          // RAZ des variables supérieures à H128
    {
        *H0=0;
    }
    _Memorize(MEM_LDDATA);
    _Wait(100);

    GFMaster                = 1;          // Pondération des incréments du maître
    GFSlave                 = 1;          // Pondération des incréments de l'esclave
    MasterSource             = 200;        // Source des incr. du maître :
                                           // Variable H200 "Position maître" (via Sbus)
    MFilterTime             = 1;          // Traitement des incr. du maître sans filtre
    StartupCycleMode        = 1;          // Mode synchro 1 : lancement événementiel
                                           // de la synchronisation via entrée binaire
    StartupCycleInputMask   = 16;         // Choix de la borne DI04 pour la synchronisation
    StartupCycleMasterLength = 10000;     // Nbre incr. maître jusqu'à fin synchronisation
    _BitSet(StartupCycleModeControl, 12); // Choix du rattrapage en distance
    RegisterLoopDXDTOut     = 2;          // Limitation dispositif de correction
    StopCycleMode           = 1;          // Mode désolidarisation 1: lancement événementiel
                                           // de la désolidarisation via entrée binaire
    StopCycleInputMask      = 32;         // Choix de la borne DI05 pour la désolidarisation
}

```