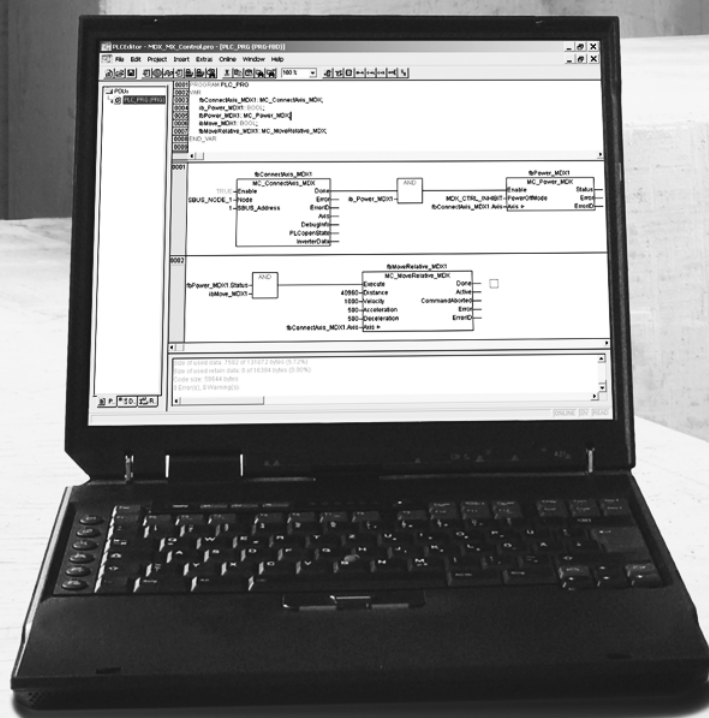


**SEW**
EURODRIVE

Bibliothèques MPLCTec.._MX pour MOVI-PLC® GearMotion, CamMotion, CamSwitch

Version 01/2009

16751620 / FR

Manuel





1	Remarques générales.....	6
1.1	Structure des consignes de sécurité.....	6
1.2	Recours en cas de défectuosité.....	6
1.3	Exclusion de la responsabilité.....	6
1.4	Autres documentations.....	7
1.5	Consignes de sécurité spécifiques pour les systèmes de bus.....	7
1.6	Fonctions de sécurité.....	7
1.7	Applications de levage.....	7
2	MPLCTecGearMotion_MX.....	8
2.1	Introduction.....	8
2.2	Domaines d'application.....	9
2.3	Configuration.....	10
2.3.1	Conditions.....	10
2.3.2	Principe de base et remarques.....	10
2.3.3	Différents modes de synchronisation.....	11
2.3.4	Gestion de l'offset.....	13
2.3.5	Différents résultats de démarrages de synchronisation et d'offset.....	14
2.3.6	Logique de détermination.....	16
2.4	La bibliothèque MPLCTecGearMotion_MX.....	18
2.4.1	Aperçu via la bibliothèque MPLCTecGearMotion_MX.....	18
2.4.2	Types de données de la bibliothèque MPLCTecGearMotion_MX.....	20
2.4.3	Configuration PDO.....	25
2.5	Les blocs fonction de la bibliothèque MPLCTecGearMotion_MX.....	28
2.5.1	MC_LinktecGear_MX.....	28
2.5.2	MC_SetGearConfig_MX.....	31
2.5.3	MC_PrepareGearInTimeBased_MX.....	33
2.5.4	MC_PrepareGearInDistanceBased_MX.....	35
2.5.5	MC_PrepareOffsetDistanceBased_MX.....	37
2.5.6	MC_GearInControl_MX.....	39
2.5.7	MC_GearClearLag_MX.....	41
2.5.8	MC_GearInDirect_MX.....	42
2.5.9	MC_GearOffsetDirectTime_MX.....	43
2.5.10	MC_GearOffsetDirectDistance_MX.....	44
2.6	Exemples.....	45
2.6.1	1. Le maître est un codeur virtuel, synchronisation directe par rattrapage en vitesse / temps.....	45
2.6.2	2. MOVIAXIS® maître, synchronisation par interruption DI02 ...	47
2.6.3	3. Gestion d'offset en distance, activée selon une trajectoire maître définie, répétition automatique.....	50
2.7	Annexes.....	54
2.7.1	Détection des défauts (ErrorID).....	54
2.7.2	Interruption des tâches des blocs fonction (CommandAborted).....	54



3	MPLCTecCamMotion_MX.....	56
3.1	Introduction	56
3.2	Domaines d'application.....	57
3.3	Configuration.....	58
3.3.1	Conditions.....	58
3.3.2	Principe de base et remarques.....	58
3.3.3	Synchronisation directe	59
3.3.4	Calage de l'axe et synchronisation.....	59
3.3.5	Synchronisation avec interruption.....	59
3.3.6	Gestion de l'offset.....	60
3.3.7	Logique de détermination	63
3.3.8	Stockage des données	64
3.4	La bibliothèque MPLCTecCamMotion_MX.....	66
3.4.1	Bref aperçu de la bibliothèque MPLCTecCamMotion_MX	66
3.4.2	Types de données de la bibliothèque MPLCTecCamMotion_MX.....	69
3.4.3	Configuration avancée.....	77
3.4.4	ExtConfig avec MC_LinkTecCam_MX	77
3.4.5	ExtConfig avec MC_SetCamConfig_MX	78
3.4.6	Configuration PDO.....	78
3.5	Les blocs fonction de la bibliothèque MPLCTecCamMotion_MX	79
3.5.1	MC_LinktecCam_MX.....	79
3.5.2	MC_SetCamConfig_MX	82
3.5.3	MC_PrepareCamIn_MX	84
3.5.4	MC_CamInControl_MX.....	86
3.5.5	MC_CamInDirect_MX.....	88
3.5.6	MC_CamInAdjust_MX	89
3.5.7	MC_CamOffsetDirect_MX	91
3.5.8	MC_CamTableSelect_MX	92
3.5.9	MC_CamTableSelectTrans_MX.....	93
3.5.10	MC_CamTableScale_MX	95
3.5.11	MC_CamCurveFlowSelect_MX	96
3.5.12	MC_SetCamTransitionDynamics_MX	99
3.5.13	MC_CamMasterPosSHL_MX.....	100
3.5.14	MC_SetMasterCycle_MX	101
3.5.15	MC_CamCalcRtoR_MX.....	102
3.5.16	MC_CamCalcRtoR50_MX.....	108
3.5.17	MC_CamCalcSpline_MX.....	112
3.5.18	MC_ReadCamTable_MX.....	115
3.5.19	MC_WriteCamTable_MX.....	116
3.5.20	MC_CamGCDMasterNumDenom_MX.....	117
3.5.21	MC_CamGCDMasterUnit_MX.....	118
3.5.22	MC_DerivateGenSwitch_MX.....	119
3.6	Exemple	120
3.6.1	Came électronique avec codeur virtuel maître	120
3.7	Annexes	126
3.7.1	Détection des défauts (ErrorID).....	126
3.7.2	Interruption des blocs fonction (CommandAborted)	126



4	MPLCTecCAMSwitch_MX.....	128
4.1	Introduction	128
4.2	Domaines d'application.....	129
4.2.1	Fenêtre de cames et voies	129
4.2.2	Vitesse de traitement	132
4.2.3	Résultats.....	133
4.3	Configuration.....	134
4.3.1	Principe de base et remarques.....	134
4.3.2	Logique de détermination	134
4.3.3	Stockage des données dans le MOVIAXIS®	135
4.3.4	Structure DDB complète	136
4.4	La bibliothèque MPLCTecCamSwitch_MX	140
4.4.1	Bref aperçu de la bibliothèque MPLCTecCamSwitch_MX	140
4.4.2	Les types de données de la bibliothèque MPLCTecCamSwitch_MX141	
4.4.3	Configuration PDO.....	145
4.5	Les blocs fonction de la bibliothèque MPLCTecCamSwitch_MX	146
4.5.1	MC_LinkTecCamSwitch_MX.....	146
4.5.2	MC_SetCamSwitchConfig_MX.....	149
4.5.3	MC_SetCamSwitchAreas_MX.....	150
4.5.4	MC_StarStopCamSwitch_MX.....	152
4.5.5	MC_ForceCamSwitch_MX	153
4.6	Exemple	154
4.6.1	Huit voies et 32 cames	154
4.7	Annexes	163
4.7.1	Définition des défauts (ErrorID)	163
5	Index.....	164



1 Remarques générales

1.1 Structure des consignes de sécurité

Les consignes de sécurité du présent manuel sont structurées de la manière suivante.

Pictogramme	TEXTE DE SIGNALISATION !
	<p>Nature et source du danger</p> <p>Risques en cas de non-respect des consignes</p> <ul style="list-style-type: none"> Mesure(s) préventive(s)

Pictogramme	Texte de signalisation	Signification	Conséquences en cas de non-respect
Exemple :	DANGER !	Danger imminent	Blessures graves ou mortelles
 Danger général	AVERTISSEMENT !	Situation potentiellement dangereuse	Blessures graves ou mortelles
 Danger spécifique, p. ex. d'électrocution	ATTENTION !	Situation potentiellement dangereuse	Blessures légères
	STOP !	Risque de dommages matériels	Endommagement du système d'entraînement ou du milieu environnant
	REMARQUE	Remarque utile ou conseil facilitant la manipulation du système d'entraînement	

1.2 Recours en cas de défectuosité

Il est impératif de respecter les instructions et remarques de la documentation MOVI-PLC® afin d'obtenir un fonctionnement correct et de bénéficier, le cas échéant, d'un recours en cas de défectuosité. Il est donc recommandé de lire le manuel avant de faire fonctionner les appareils !

Vérifier que le manuel est accessible aux responsables de l'installation et de son exploitation ainsi qu'aux personnes travaillant sur l'installation sous leur propre responsabilité dans des conditions de parfaite lisibilité.

1.3 Exclusion de la responsabilité

Le respect des instructions de la documentation MOVI-PLC® est la condition pour être assuré du bon fonctionnement et pour obtenir les caractéristiques de produit et les performances indiquées. SEW décline toute responsabilité en cas de dommages corporels ou matériels survenus suite au non-respect des consignes de la notice d'exploitation. Les recours de garantie sont exclus dans ces cas.



1.4 Autres documentations

- Ne faire installer et mettre en service que par du personnel électricien qualifié conformément aux prescriptions de protection en vigueur et selon les instructions des documents suivants :
 - Notice d'exploitation "Servovariateurs multi-axes MOVIAXIS® MX"
 - Manuel "MOVI-PLC® *advanced* DH.41B"
 - Manuel "Bibliothèques MPLCMotion_MDX et MPLCMotion_MX pour MOVI-PLC®"
 - Manuel "Bibliothèques pour MOVI-PLC® - Codes défaut"
 - Manuel Programmation "MOVI-PLC® dans l'éditeur PLC"
- Il est recommandé de lire attentivement ces documents avant de commencer l'installation et la mise en service de la commande MOVI-PLC®.
- Il est impératif de respecter les instructions et remarques de la présente documentation afin d'obtenir un fonctionnement correct et de bénéficier, le cas échéant, d'un recours en garantie.

1.5 Consignes de sécurité spécifiques pour les systèmes de bus

Ce système de communication permet d'adapter précisément la carte de pilotage MOVI-PLC® et le servovariateur multi-axes MOVIAXIS® à l'application. Comme pour tout système programmable, il subsiste le risque d'une erreur de programmation qui peut mener à un comportement incontrôlé.

1.6 Fonctions de sécurité

La commande MOVI-PLC® ne doit pas assurer de fonctions de sécurité.

Pour les applications en mode sécurisé, il est impératif de tenir compte des indications de la documentation suivante :

- Coupure sécurisée pour MOVIAXIS®

Pour les applications en mode sécurisé, seuls les sous-ensembles SEW fournis explicitement pour cette exécution sont autorisés !

1.7 Applications de levage

- Les applications de levage avec MOVI-PLC® ne sont possibles que dans les conditions suivantes.
 - Une mise en service spécifique pour dispositif de levage doit être réalisée.
- Les MOVI-PLC® ne doivent pas être utilisés comme dispositifs de sécurité pour les applications de levage.

Pour éviter des dommages corporels ou matériels, prévoir des systèmes de surveillance ou des dispositifs de protection mécaniques.



2 MPLCTecGearMotion_MX

2.1 Introduction

Les servovariateurs multi-axes MOVIAXIS® offrent des fonctions technologiques très complètes, telles par exemple :

- Réducteur électronique,
- Cames électroniques,
- Capteurs d'événements,
- Gestion d'événements,
- Boîte à cames.

L'assistant de mise en route "Motion Technologie-Editor" de MotionStudio permet de configurer toutes les fonctions technologiques. Il permet d'effectuer un réglage de base. Mais il est fréquent que certaines valeurs ou réglages nécessitent une modification en cours de process, ou alors diverses fonctions devront être combinées et se dérouler de manière coordonnée dans le temps.

Les blocs fonction de la bibliothèque "MPLCTecGearMotion_MX" règlent les paramètres Gear généraux et configurent également d'autres fonctions technologiques nécessaires pour la fonction de pilotage souhaitée.

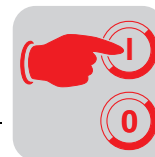
Une initiation approfondie aux paramètres respectifs des fonctions technologiques n'est ainsi pas nécessaire. La configuration avec l'éditeur technologique n'est pas nécessaire, tous les réglages étant effectués par les blocs fonction de la bibliothèque "MPLCTecGearMotion_MX".

Les avantages de la bibliothèque "MPLCTecGearMotion_MX" sont :

- Paramétrage selon CEI61131, permettant l'implémentation aisée des applications spécifiques
- Bibliothèque pour le pilotage de la fonction technologique Gear. Celle-ci permet la configuration et le pilotage simplifiés des fonctionnalités Gear sans nécessiter de connaissances spécifiques des différentes fonctions technologiques.
- Pilotage centralisé de plusieurs axes synchronisés.
- Les fonctions complémentaires, comme p. ex. la définition de la valeur maître par un codeur maître virtuel, sont commandées de manière centralisée via la commande MOVI-PLC®.
- Possibilité de sélectionner divers événements et types d'événements de synchronisation.
- Possibilité d'activer un déplacement d'offset supplémentaire à la synchronisation et / ou en mode synchronisé.
- Possibilité de sélectionner divers événements de lancement de l'offset et types de lancement d'offset.

D'autres informations figurent dans les documentations suivantes :

- Les remarques générales concernant le fonctionnement de la bibliothèque figurent dans le document "Bibliothèques MPLCMotion_MDX et MPLCMotion_MX pour MOVI-PLC®".
- Manuel "Fonction technologique réducteur électronique pour MOVIAXIS® MX".



2.2 Domaines d'application

La bibliothèque `MPLCTecGearMotion_MX` convient à toutes les applications nécessitant des déplacements d'axes synchronisés.

Exemples d'application

- Coupe à la volée, par exemple pour la découpe d'un matériau continu
- Transport de matériel synchronisé, p. ex. synchronisation de plusieurs bandes transporteuses sur un codeur maître
- Dispositifs de translation et de levage multi-axes
- Dispositifs de remplissage dans l'industrie des boissons
- Techniques d'emballage, p. ex. ensacheuses

Propriétés

- Possibilité de fonctionnement synchronisé de 64 axes moteur maximum (avec `MOVI-PLC® advanced`).
- Nombreuses possibilités de configuration, permettant une programmation claire à l'aide des blocs fonctions correspondants.
- Possibilité de régler différentes sources de codeur maître
- Choix entre divers modes de synchronisation
- Choix entre divers types d'offset
- Mise en route simplifiée de l'offset de synchronisation ou en mode synchronisé.
- Possibilité de réaliser diverses combinaisons maître/esclave et de configurer plusieurs maîtres de synchronisation



2.3 Configuration

2.3.1 Conditions

- Une commande MOVI-PLC® en exécution technologique T1 ou supérieure est nécessaire afin de pouvoir utiliser la bibliothèque `MPLCTecGearMotion`.
- La bibliothèque `MPLCMotion_MX` doit être intégrée. Les conditions préalables indiquées dans le manuel "Bibliothèques MPLCMotion_MDX et MPLCMotion_MX pour MOVI-PLC®" sont également applicables.

2.3.2 Principe de base et remarques

- Réglage des fonctions technologiques de l'axe adéquates à l'aide de blocs fonction CEI.
- La synchronisation et la resynchronisation s'effectuent via la fonction technologique Gear, puis le système bascule automatiquement sur la fonction linéaire Cam, ce qui rend possible l'offset en mode synchronisé.
- Synchronisation rapide gérée par interruption grâce au recours à la fonction capteur d'événements et à la gestion d'événements (la configuration correspondante s'effectue automatiquement).
- Offset synchronisé en temps réalisé par dépassement du générateur de profil Cam avec possibilité d'effectuer l'offset en valeur relative ou absolue.
- Offset en distance par la superposition d'une seconde came électronique configurée de manière adéquate. Lancement de l'offset configurable pour gestion par interruption, selon une trajectoire maître ou directement depuis le programme PLC.
- Procédures de synchronisation et d'offset gérées par interruption ou par compteur, pouvant être répétées automatiquement via fonction AutoRestart.
- Le bloc fonction "MC_LinkTecGear_MX" est impérativement nécessaire comme interface de données centrale pour la fonction technologique Gear.
- L'éditeur technologique de MotionStudio n'est pas nécessaire, mais pour le diagnostic il est possible de lancer une liaison online du moniteur avec l'exemple de projet "MOVI-PLC Defaultprojekt.Tec-Function". Cependant l'exemple de projet ne devra pas être chargé, sans quoi les réglages effectués par la bibliothèque seraient modifiés.



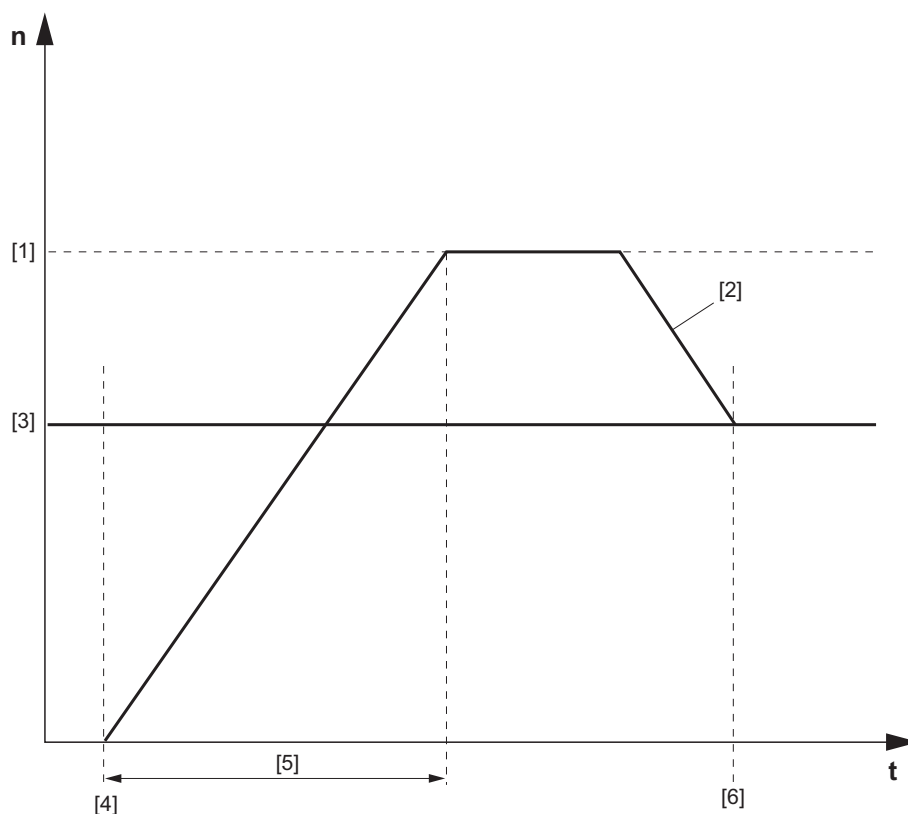
2.3.3 Différents modes de synchronisation

Il est en principe possible de définir si la synchronisation doit s'effectuer par rattrapage en distance ou en vitesse/temps.

Synchronisation en temps

Pour une synchronisation en temps, sélectionner le bloc fonction "MC_PrepareGearInTimeBased_MX".

Dans le cas du rattrapage en vitesse/temps, l'esclave se synchronise sur le maître selon la vitesse *GearSpeed* et *GearAcceleration/Deceleration* réglée.



61685AXX

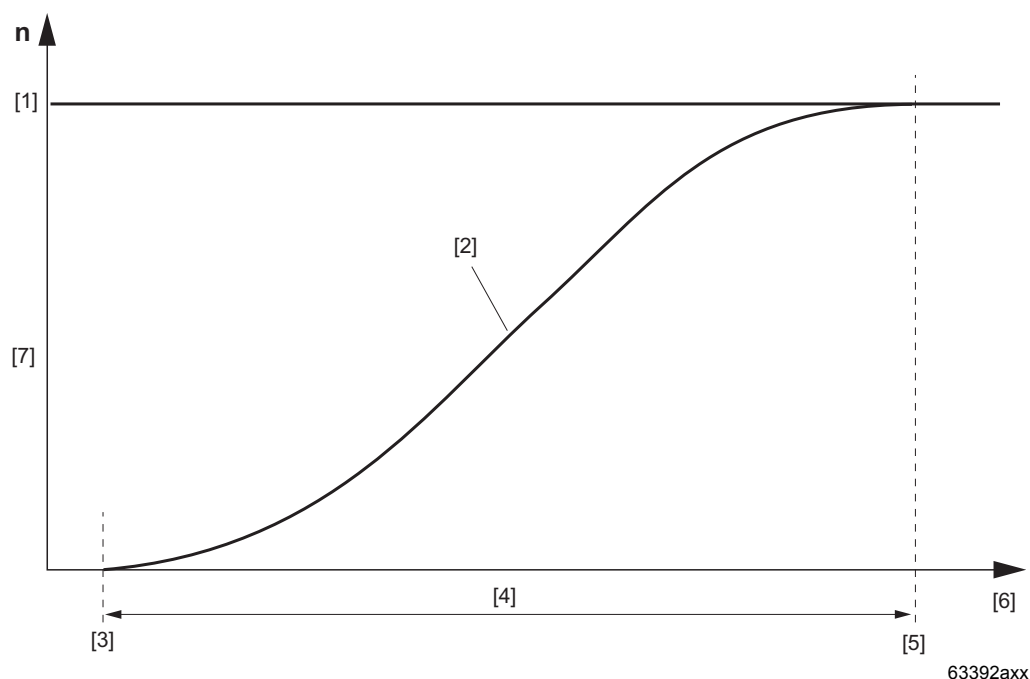
- [1] Vitesse réglée via MC_SetGearDynamics_MX
- [2] Vitesse de l'esclave
- [3] Vitesse du maître
- [4] Démarrage du processus de synchronisation par rattrapage en vitesse/temps
- [5] Rampe réglée via MC_SetGearDynamics_MX (rapportée à 3000 tr/min)
- [6] L'esclave est synchronisé sur le maître


**Synchronisation
par rattrapage en
distance**

Pour une synchronisation par rattrapage en distance, sélectionner le bloc fonction "MC_PrepareGearInDistanceBased_MX".

En cas de synchronisation en distance, l'esclave se synchronise sur le maître sur une distance maître réglable. Pour régler la distance parcourue par l'esclave, sélectionner le bloc fonction *GearInSlaveDistance*.

La synchronisation par rattrapage en distance dépend du sens de rotation. En d'autres termes, si la trajectoire du maître est indiquée comme valeur positive pour la synchronisation avec *GearInMasterDistance*, le maître doit tourner en sens positif.



- [1] Vitesse du maître
- [2] Vitesse de l'esclave
- [3] Démarrage de la synchronisation par rattrapage en distance
- [4] Trajectoire du maître pour la synchronisation
- [5] L'esclave est synchronisé sur le maître
- [6] Position du maître
- [7] Trajectoire de l'esclave pour la synchronisation


REMARQUES

La calcul du profil de synchronisation s'effectue selon un polynôme 5è degré.

Par conséquent, la distance de synchronisation de l'esclave doit être égale à environ la moitié de la trajectoire mise à l'échelle du maître, sans quoi l'esclave risquerait de dépasser la position cible ou de modifier brièvement son sens de rotation durant la phase de synchronisation.

$$GearInSlaveDistance = 0.5 \cdot GearInMasterDistance \cdot Numerator / Denominator$$



2.3.4 Gestion de l'offset

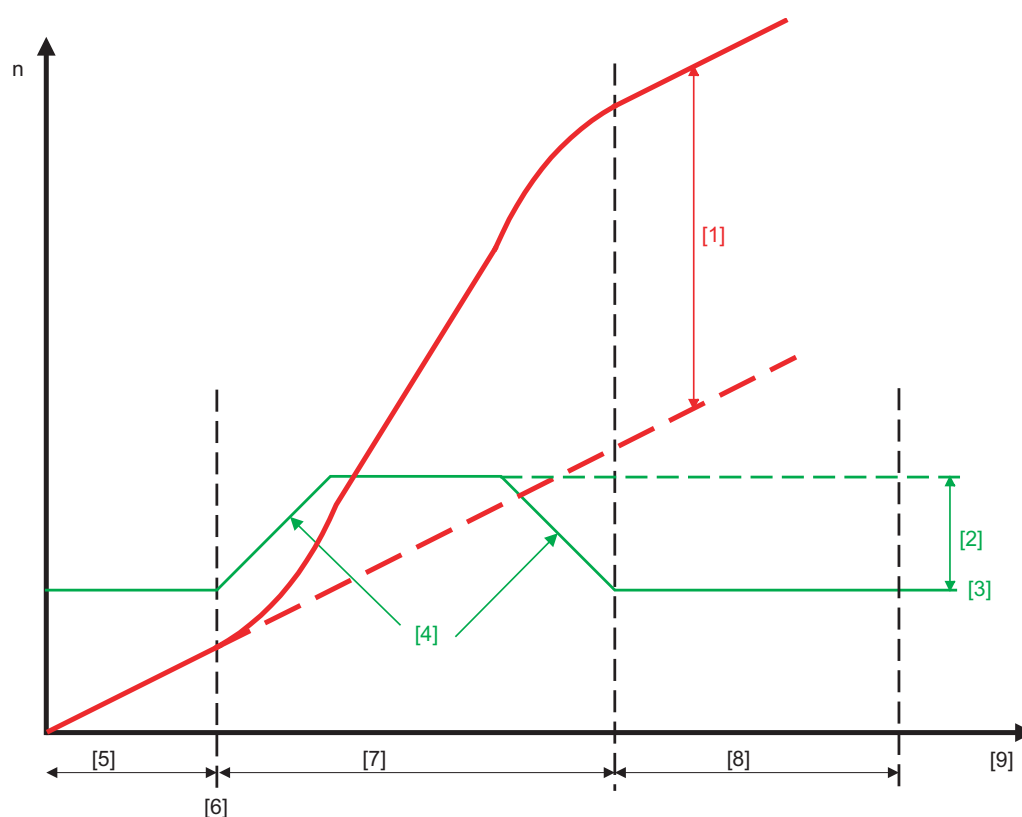
Lorsque la synchronisation est achevée, il est possible d'activer un déplacement d'offset au niveau de l'esclave en cours de fonctionnement synchronisé. L'entrée *OffsetMode* permet de définir si le déplacement doit s'effectuer en valeur relative ou absolue :

OffsetMode = MX_GEAR_OFFSET_RELATIVE

Dans ce cas, en plus du déplacement synchronisé avec le maître, l'esclave effectue un positionnement. Il est ainsi possible de réaliser un décalage de phase définitif.

OffsetMode = MX_GEAR_OFFSET_ABSOLUTE

Ce réglage permet d'effectuer le déplacement *OffsetDistance* en valeur absolue. En cas de passage en mode Cam (FCB16), la valeur de référence est mise à zéro ; les valeurs de position suivantes se réfèrent à cette valeur. Ceci permet p. ex d'effectuer un offset et de supprimer le décalage de phase par un deuxième déplacement d'offset avec *Distance* = 0.



63360axx

- | | | | |
|-----|------------------------------------|-----|----------------------|
| [1] | OffsetPosition | [6] | Offset de démarrage |
| [2] | OffsetSpeed | [7] | Déplacement d'offset |
| [3] | n esclave | [8] | Synchrone |
| [4] | Offset accélération / décélération | [9] | Position maître |
| [5] | Synchrone | | |



2.3.5 Différents résultats de démarrages de synchronisation et d'offset

Outre le mode de synchronisation utilisé, les blocs fonction "PrepareGearIn..." permettent également de définir l'événement qui déclenchera la synchronisation.

Que la synchronisation soit définie par rattrapage en distance ou en temps, l'événement de synchronisation est à définir via l'entrée *GearInEvent* du bloc fonction "Prepare" concerné.

Événements de synchronisation possibles :

- **MX_GEAR_DIRECT**

Le bloc fonction "MC_GearInDirect_MX" permet de lancer directement la synchronisation.

- Événements de synchronisation gérés par interruption :

MX_GEAR_INTERRUPT_DI01_POS_FLAG

MX_GEAR_INTERRUPT_DI01_NEG_FLAG

MX_GEAR_INTERRUPT_DI01

...

MX_GEAR_INTERRUPT_DI08_POS_FLAG

MX_GEAR_INTERRUPT_DI08_NEG_FLAG

MX_GEAR_INTERRUPT_DI08

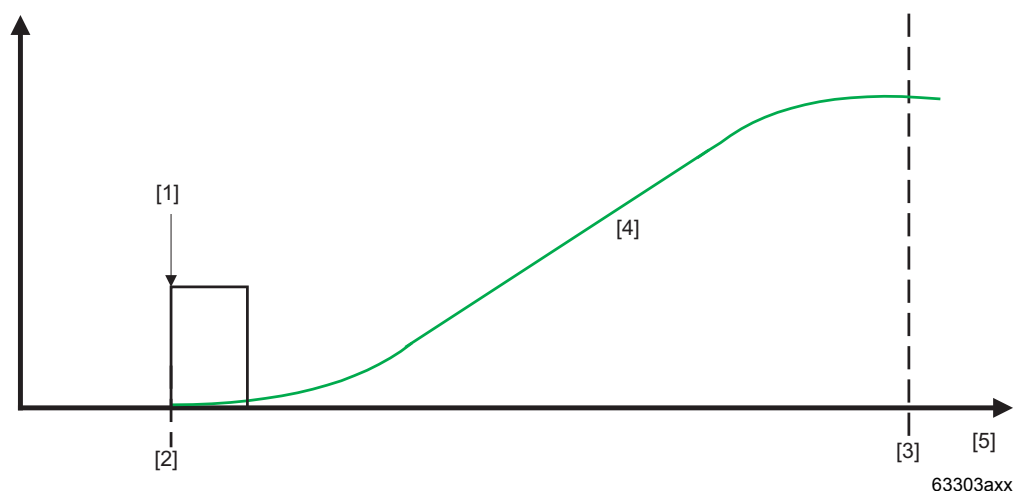
MX_GEAR_INTERRUPT_C_TRACK_ENC1

MX_GEAR_INTERRUPT_C_TRACK_ENC2

MX_GEAR_INTERRUPT_C_TRACK_ENC3

La synchronisation est lancée par un événement de synchronisation, déclenchée par un changement de signal sur l'entrée DI01 - DI08 du MOVIAXIS®, par un front montant / descendant (Pos / Neg Flag) ou par un changement de signal sur l'entrée DI01 - DI08 / voie C.

Exemple : *GearInEvent* = MX_GEAR_INTERRUPT_DI02_POS_FLAG

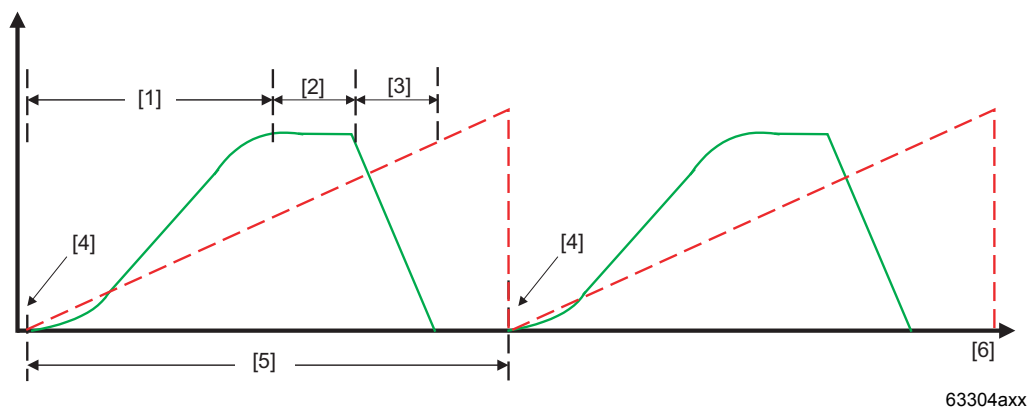


- [1] Interrupt DI02 (PosFlag)
- [2] Démarrage synchronisation
- [3] L'esclave est synchronisé
- [4] n esclave
- [5] Position maître



- MX_GEAR_DISTANCE_COUNTER

Avec ce réglage, la synchronisation est lancée selon la distance maître définie. Le compteur actuel pour la distance du maître n'est alors pas remis à zéro.



- [1] Mise en synchronisation
- [2] Synchrone
- [3] Désolidarisation
- [4] Démarrage synchronisation
- [5] DistanceForStartGearIn
- [6] Position maître

- MX_GEAR_DISTANCE_COUNTER_RESET_COUNT

La synchronisation s'effectue de la même manière qu'en mode MX_GEAR_DISTANCE_COUNTER, à la différence que le compteur pour la distance du maître est remis à zéro en cas d'appel de la fonction "MC_PrepareGearIn....".

OffsetEvent

La description des événements de synchronisation s'applique de manière similaire pour les diverses possibilités de lancement d'un déplacement d'offset.



2.3.6 Logique de détermination

Les blocs fonction de la bibliothèque MPLCTecGearMotion_MX sont répartis en quatre répertoires.

MX_Gear_Config

- MC_SetGearConfig_MX

MX_Gear_Main

- MC_LinkTecGear_MX
- MC_GearClearLag_MX

MX_Gear_Parameter

- MC_GearInControl_MX
- MC_PrepareGearInTimeBased_MX
- MC_PrepareGearInDistanceBased_MX
- MC_PrepareOffsetDistanceBased_MX

MX_Gear_SingleAxis

- MC_GearInDirect_MX
- MC_GearOffsetDirectDistance_MX
- MC_GearOffsetDirectTime_MX

Procédure

- Insérer la bibliothèque "MPLCTecGearMotion_MX" avec le gestionnaire des bibliothèques en complément à la bibliothèque "MPLCMotion_MX" dans le projet.
- Le bloc fonction "MC_LinkTecGear_MX" est une interface d'une fonction technologique de l'axe et doit donc être appelé de manière cyclique dans le programme. Lorsque le bloc fonction "MC_ConnectAxis_MX" a établi la liaison avec l'axe (Done = True), la fonction *LinkTecGear* active les fonctions technologiques du MOVIAXIS® nécessaires pour la fonction Gear. Les réglages nécessaires sont par ailleurs effectués dans l'éditeur PDO. Un objet de réception est entre autres configuré dans le tampon 5 afin de réceptionner la position maître via le bus système.
- Si la position du maître doit être transmise via le bus système, ce dernier doit être synchronisé. Le télégramme de synchronisation du SBus nécessaire est à configurer à l'aide du bloc fonction "MC_SetSync_CAN" (qui fait partie de la bibliothèque MPLCCommunication_CAN). Si le codeur virtuel de la commande est utilisé comme maître, le SBus est synchronisé automatiquement à l'appel du bloc fonction "MC_LinkTecAxis_Virtual" (qui fait partie de la bibliothèque MPLCTecVirtualEncoder). Les réglages de base nécessaires dans l'arborescence des paramètres sous "Communication" sont effectués automatiquement lors de l'initialisation du bloc fonction *LinkTecGear*.
- Le bloc fonction "MC_SetGearConfig_MX" permet dans un premier temps de régler les principaux paramètres de synchronisme. Ce bloc fonction doit être ouvert par un front montant sur l'entrée Execute à chaque nouveau démarrage de la commande MOVI-PLC® ou lorsqu'on souhaite modifier les paramètres de synchronisme généraux.

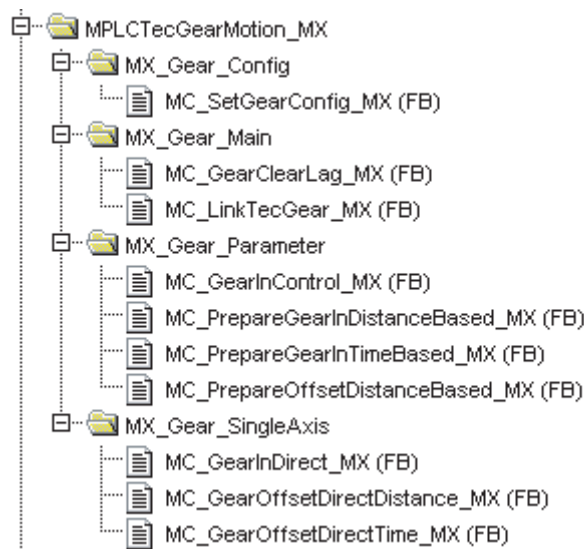


- Les blocs fonction Prepare
 - permettent de définir le mode de synchronisation,
 - permettent de définir le mode de démarrage de l'offset,
 - permettent de configurer par quel événement cela doit être déclenché,
 - permettent de régler les paramètres respectifs nécessaires.
- La synchronisation s'effectue ensuite via "MC_GearInDirect_MX" ou via un autre événement configuré.
- Durant le fonctionnement synchronisé, il est possible de lancer un déplacement d'offset par un front montant sur l'entrée Execute de "MC_GearOffsetDirectDistance_MX" ou de "MC_Gear OffsetDirectTime_MX".



2.4 La bibliothèque MPLCTecGearMotion_MX

2.4.1 Aperçu via la bibliothèque MPLCTecGearMotion_MX



63305axx

MC_SetGearConfig_MX

Ce bloc fonction permet d'effectuer la configuration de base de la synchronisation. Avant de lancer la synchronisation, les paramètres de synchronisation doivent avoir été transférés via ce bloc fonction.

MC_LinkTecGear_MX

Ce bloc fonction a diverses fonctionnalités :

- Configuration des fonctions technologiques du MOVIAXIS® et activation des fonctions nécessaires pour la fonction technologique Gear.
- Configuration de l'interface de données-process MOVIAXIS®.
- Indication de l'état de la fonction technologique Gear et de l'état d'initialisation.

MC_GearClearLag_MX

Ce bloc fonction permet de réinitialiser la différence entre la consigne Gear (signal maître) et la mesure Gear (position actuelle de l'esclave).

MC_GearInControl_MX

Ce bloc fonction contient des fonctions de pilotage supplémentaires pour les événements de synchronisation gérés par interruption ou par compteur. La synchronisation peut ainsi être libérée ou verrouillée à plusieurs reprises.

MC_PrepateGearInTimeBased_MX

Ce bloc fonction Prepare prépare la synchronisation en vitesse/temps. Il permet de définir et de configurer les paramètres de dynamisme et l'événement de synchronisation souhaités.



MC_PrepareGearInDistanceBased_MX

Ce bloc fonction Prepare permet de sélectionner et de régler la synchronisation en distance. Il permet de définir et de configurer les distances de synchronisation du maître et de l'esclave et l'événement de synchronisation.

MC_PrepareOffsetDistanceBased_MX

Ce bloc fonction permet de préparer un déplacement d'offset par rattrapage en distance. Il configure les déplacements d'offset maître / esclave ainsi que l'événement de démarrage de l'offset.

MC_GearInDirect_MX

Ce bloc fonction permet de lancer la synchronisation directe par un front montant sur l'entrée Execute. La synchronisation s'effectue avec les paramètres réglés par le bloc *PrepareGearIn*.

MC_GearOffsetDirectDistance_MX

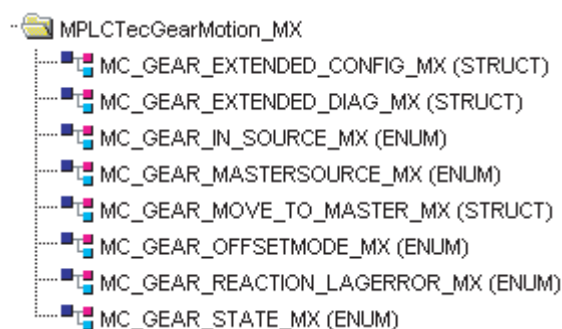
Ce bloc fonction permet de lancer un déplacement d'offset par rattrapage en distance avec les valeurs définies *OffsetMasterDistance* / *OffsetSlaveDistance*.

MC_GearOffsetDirectTime_MX

Ce bloc fonction active, en amont de la synchronisation, un positionnement à l'aide des paramètres transmis. Ceci déclenche la réalisation d'un offset relatif ou absolu selon l'"OffsetMode" ("Phasing").



2.4.2 Types de données de la bibliothèque MPLCTecGearMotion_MX



63306AXX

Configuration standard

- **Type de donnée MC_GEAR_IN_SOURCE_MX**

Ce type de donnée permet, via les blocs fonction "PrepareGearIn" et "PrepareOffset", de définir l'événement pour le lancement de la synchronisation ou du déplacement d'offset. Les événements gérés par interruption possibles sont le changement de front sur les entrées binaires 02 / 03 ou sur la voie C d'une des trois entrées codeur. Les modes de pilotage avec compteur de distance (maître) se distinguent par la possibilité supplémentaire de remettre à zéro le compteur de distance maître actuel en sélectionnant le bloc fonction "Prepare" correspondant.

```

TYPE MC_GEAR_IN_SOURCE_MX :
(
    MX_GEAR_DIRECT,
    MX_GEAR_INTERRUPT_DI01_POS_FLAG
    MX_GEAR_INTERRUPT_DI01_NEG_FLAG
    MX_GEAR_INTERRUPT_DI01
    MX_GEAR_INTERRUPT_DI02_POS_FLAG
    MX_GEAR_INTERRUPT_DI02_NEG_FLAG
    MX_GEAR_INTERRUPT_DI02
    MX_GEAR_INTERRUPT_DI03_POS_FLAG
    MX_GEAR_INTERRUPT_DI03_NEG_FLAG
    MX_GEAR_INTERRUPT_DI03
    MX_GEAR_INTERRUPT_DI04_POS_FLAG
    MX_GEAR_INTERRUPT_DI04_NEG_FLAG
    MX_GEAR_INTERRUPT_DI04
    MX_GEAR_INTERRUPT_DI05_POS_FLAG
    MX_GEAR_INTERRUPT_DI05_NEG_FLAG
    MX_GEAR_INTERRUPT_DI05
    MX_GEAR_INTERRUPT_DI06_POS_FLAG
    MX_GEAR_INTERRUPT_DI06_NEG_FLAG
    MX_GEAR_INTERRUPT_DI06
    MX_GEAR_INTERRUPT_DI07_POS_FLAG
    MX_GEAR_INTERRUPT_DI07_NEG_FLAG
    MX_GEAR_INTERRUPT_DI07
    MX_GEAR_INTERRUPT_DI08_POS_FLAG
    MX_GEAR_INTERRUPT_DI08_NEG_FLAG
    MX_GEAR_INTERRUPT_DI08
    MX_GEAR_INTERRUPT_C_TRACK_ENC1
    MX_GEAR_INTERRUPT_C_TRACK_ENC2
    MX_GEAR_INTERRUPT_C_TRACK_ENC3
    MX_GEAR_DISTANCE_COUNTER
    MX_GEAR_DISTANCE_COUNTER_RESET_COUNT
);
END_TYPE

```

63307AXX



- **Type de données MC_GEAR_MASTERSOURCE_MX**

Ce type de donnée fixe la source du signal du codeur maître. Il est possible de sélectionner les trois entrées codeur MOVIAXIS® (linéaires ou modulo). Il est également possible de sélectionner le codeur virtuel de l'axe (linéaire ou modulo). De plus le signal du maître peut également être reçu via le SBus, p. ex. par la commande MOVI-PLC® ou par un autre axe. Pour cela, deux objets CAN au choix sont disponibles

```
TYPE MC_GEAR_MASTERSOURCE_MX :
(
    MX_GEAR_SYSTEMPOS_ENCODER1,
    MX_GEAR_SYSTEMPOS_ENCODER2,
    MX_GEAR_SYSTEMPOS_ENCODER3,
    MX_GEAR_MODULOPPOS_ENCODER1,
    MX_GEAR_MODULOPPOS_ENCODER2,
    MX_GEAR_MODULOPPOS_ENCODER3,
    MX_GEAR_MODULO_VIRTUAL_ENCODER_MX,
    MX_GEAR_LINEAR_VIRTUAL_ENCODER_MX,
    MX_GEAR_RECEIVE_PDO_1,
    MX_GEAR_RECEIVE_PDO_2
);
END_TYPE
```

63308axx

- **Type de données MC_GEAR_STATE_MX**

Affichage de l'état actuel de la synchronisation

```
TYPE MC_GEAR_STATE_MX :
(
    MX_GEAR_OUTGEAR, (* Synchronous state : Axis is geared out *)
    MX_GEAR_ENGAGING_GEAR_IN, (* Synchronous state : Axis engaging to gear *)
    MX_GEAR_IN_GEAR, (* Synchronous state : Axis is in gear *)
    MX_GEAR_OFFSET_GEAR, (* Synchronous state : Offset move while in gear *)
    MX_GEAR_MOVE_TO_MASTER, (* Synchronous state : Slave moves to master after inl *)
    MX_GEAR_NOTLINKED (* Synchronous state : Axis not linked *)
);
END_TYPE
```

63309AXX

- **Type de données MC_GEAR_OFFSETMODE_MX**

Permet d'opter pour un offset relatif ou absolu.

```
TYPE MC_GEAR_OFFSETMODE_MX :
(
    MX_GEAR_OFFSET_RELATIVE, (* Relative Offset move *)
    MX_GEAR_OFFSET_ABSOLUTE (* Absolute Offset move *)
);
END_TYPE
```

63310axx



Configuration avancée

Les fonctions technologiques du MOVIAXIS® offrent de nombreuses possibilités de réglage, des fonctions de filtrage ou de compensation qui ne sont pas nécessaires pour la plupart des applications standard. Pour une bonne visibilité des blocs fonction tout en ayant une possibilité d'accès, ces paramètres sont résumés dans la structure d'entrée "MC_GEAR_EXTENDED_CONFIG_MX".

Cette structure est utilisée à la fois par le bloc fonction "MC_LinkTecGear_MX" et par le bloc fonction "MC_SetGearConfig_MX".

• ExtConfig avec MC_LinkTecGear_MX

```

TYPE MC_GEAR_EXTENDED_CONFIG_MX :
STRUCT
(* ##### Extended LinkTecGear Configurations ##### *)

    UseTecEditorConfig: BOOL;          (* based on the PLCdefault TecEditor project additional
                                        Tec configurations can be made, e.g. CAM2/3, Touchprobe, etc. *)
    TecFunctionsUserUnitSlave: BOOL;   (* True = Tec values of the slave are scaled in user units
                                        False = Tec values are in system units (unscaled) *)

    stInitialParameter_MX : MC_INITIALPARAMETER;
    (* #####
    With this struct several user defined parameters can be written to the axis during the
    initialization of MC_LinkTecGear_MX. DHx11B : 10 parameters, DHx41B 60 parameters

    Add the following code to your program to initialise the struct :

    Example: write DDB variable 2577 and 3420:

    stInitialParameter_MX.aIndex[1] := 20220; (* Index of DDB Variable 2577 *)
    stInitialParameter_MX.aSubIndex[1] := 18; (* SubIndex of DDB Variable 2577 *)
    stInitialParameter_MX.aData[1] := 10;    (* Data to be written to DDB Variable 2577 *)

    stInitialParameter_MX.aIndex[2] := 20226; (* Index of DDB Variable 3420 *)
    stInitialParameter_MX.aSubIndex[2] := 93; (* SubIndex of DDB Variable 3420 *)
    stInitialParameter_MX.aData[2] := 20;    (* Data to be written to DDB Variable 3420 *)

    If there are more than 10 parameters to write, declare NUMOF_INITIALPARAMETER
    in your program.

    VAR_GLOBAL CONSTANT
        NUMOF_INITIALPARAMETER : UDINT := 20;
    END_VAR

    ##### *)


```

63311axx


La variable structure *UseTecEditorConfig* autorise, en parallèle à la configuration Gear avec les modules de bibliothèques, une configuration avancée avec "TecEditor". Ceci permet par exemple de régler la fonction de capteur d'événements ou de configurer en amont un bloc de courbe supplémentaire.

A la place d'une configuration avancée, cette entrée permet également la configuration Gear à l'aide du TecEditor-Wizzard ; le bloc fonction "MC_SetGearConfig_MX" n'est alors pas nécessaire.



	REMARQUES
	<p>Procédure</p> <ul style="list-style-type: none"> • Les plages DDB pouvant éventuellement se décaler, le programme CEI devrait être arrêté dans un premier temps par reset (à froid), afin d'éviter l'apparition de messages de défaut. • Utiliser impérativement comme base le projet TecEditor "MOVI-PLCDefaultprojekt.TecFunction". Ce projet peut ensuite être modifié ou élargi. Il est important de ne pas désactiver les fonctions technologiques qui sont activées dans le projet de base. • Charger le projet TecEditor modifié. • Lorsque la variable <i>UseTecEditorConfig</i> a été mise = True, le programme peut être lancé.

La variable *TecFunctionsUserUnitSlave* agit de sorte que les valeurs et données technologiques concernant l'esclave soient mises à l'échelle en unités utilisateur. Les facteurs de mise à l'échelle nécessaires pour cela sont les facteurs numérateur et dénominateur déterminés à la mise en service, avec réglage adéquat des décimales.

	REMARQUES
	<p>Attention !</p> <p>Les valeurs concernant le maître ne sont pas mises à l'échelle.</p>

Par exemple "MC_PrepareGearInDistanceBased_MX" :

DistanceForStartGearIn : distance maître selon laquelle la synchronisation doit commencer, donc valeurs indiquées non mises à l'échelle.


GearInMasterDistance : distance maître pour la synchronisation, donc valeurs indiquées non mises à l'échelle.

GearInSlaveDistance : distance de synchronisation, mise à l'échelle en unités utilisateur (si sélectionnée en *TecFunctionsUserUnitSlave* = True).

Structure d'entrée "sInitialParameter" :

Cette structure permet de régler sur l'axe des paramètres supplémentaires librement définissables durant l'initialisation de "MC_LinkTecGear_MX" (pour DH.41B, 60 paramètres max.).

Chaque paramètre à écrire doit être défini correctement avec index / sous-index / données.

	REMARQUES
	<p>Attention !</p> <p>Le système ne vérifie pas si les données sont admissibles pour le paramètre concerné ou si un paramètre déjà configuré pour une fonction est à nouveau modifié.</p> <p>C'est pourquoi il faut vérifier soigneusement l'absence de chevauchements, sans quoi la fonction Gear pourrait en être affectée.</p>



- ExtConfig pour MC_SetGearConfig_MX

```
(*##### Extended SetGearConfig configurations #####*)

MoveToMaster: MC_GEAR_MOVE_TO_MASTER_MX;
(*#####*)
(* With this struct it can be activated and adjusted that the slave moves to the master
   after State 16 was interrupted by Inhibit or Safety Stop:
   MoveToMaster.Activate_Function : True = activates the move to master function
                                   False = move to master function deactivated, if slave
                                           or master has moved more than size of
                                           lag window while disabled, drive trips with
                                           lag error when state changes again to 16.

   MoveToMaster.Slave_Speed : Speed setpoint for moving to the master
   MoveToMaster.Slave_Acceleration : Acceleration setpoint for moving to the master
   MoveToMaster.Slave_Deceleration : Deceleration setpoint for moving to the master *)

MSignal_InterpolationTime: DINT; (* InterpolationTime: range 1...60 resolution [0.5ms] (0.5...30ms)*)
MSignal_DelayTime: DINT; (* PositionDelayTime in [µs] *)
AverageFilterTime: DINT; (* AverageFilterTime: range 1...60 resolution [0.5ms] (0.5...30ms)*)
CompFilterTime: DINT; (* CompFilterTime: range 1...60 resolution [0.5ms] (0.5...30ms)*)
DeathTimeCorr_ON: BOOL; (* Activation of the death time correction: false = OFF, true = ON*)
SpeedCorr_ON: BOOL; (* Activation of the velocity related correction: false = OFF, true = ON*)
AccelerationCorr_ON: BOOL; (* Activation of the acceleration related correction: false = OFF, true = ON*)

ReactionLagError: MC_GEAR_REACTION_LAGERROR_MX;
(* defines the reaction after lag detection. Possible selection:
   NO_RESPONSE
   DISPLAY_FAULT
   IMMEDIATE_STOP_WARNING (default)
   EMERGENCY_STOP_WARNING
   APPL_STOP_WARNING
   SYST_STOP_WARNING *)

END_STRUCT
END_TYPE
```

63330axx

Activation et réglage de filtres et de fonctions de compensation :

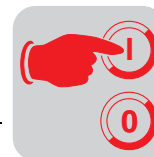
La configuration avancée pour le bloc fonction "MC_SetGearConfig_MX" contient des variables permettant de régler des filtres supplémentaires pour la préparation du signal maître.

Dans les réglages par défaut, les filtres sont désactivés ou, en cas d'utilisation du codeur virtuel de la MOVI-PLC®, la valeur de *MSignal_InterpolationTime* est réglée sur 5 ms.

Il est en plus possible d'activer des dispositifs de compensation afin de compenser la vitesse ou l'accélération. Les informations détaillées concernant le fonctionnement et les effets de ces filtres ainsi que les dispositifs de compensation sont décrits dans la documentation MOVIAXIS® correspondante (manuel "Fonction technologique réducteur électronique pour MOVIAXIS® MX").

Réglage de ReactionLagError :

Permet de régler la réaction de l'entraînement en cas de dépassement de *LagErrorWindow*. Si aucune autre valeur n'est indiquée, un arrêt à la limite application est exécuté en standard (attente).



Fonction spéciale MoveToMaster :

Si à l'état "16" (c.-à-d.. *GearState* = InGear) le verrouillage est activé ou en cas de passage en "Arrêt sécurisé" et si l'esclave ou le maître poursuit son déplacement dans cet état, une erreur de poursuite se produit généralement dès que l'état repasse à "16". Pour éviter cela, il est possible d'activer la fonction "MoveToMaster" pour configurer un positionnement de l'esclave vers la position maître. Après suppression du verrouillage ou de l'arrêt sécurisé, l'esclave se déplace selon les valeurs de vitesse et d'accélération / décélération vers le maître.

• Extended Diagnosis

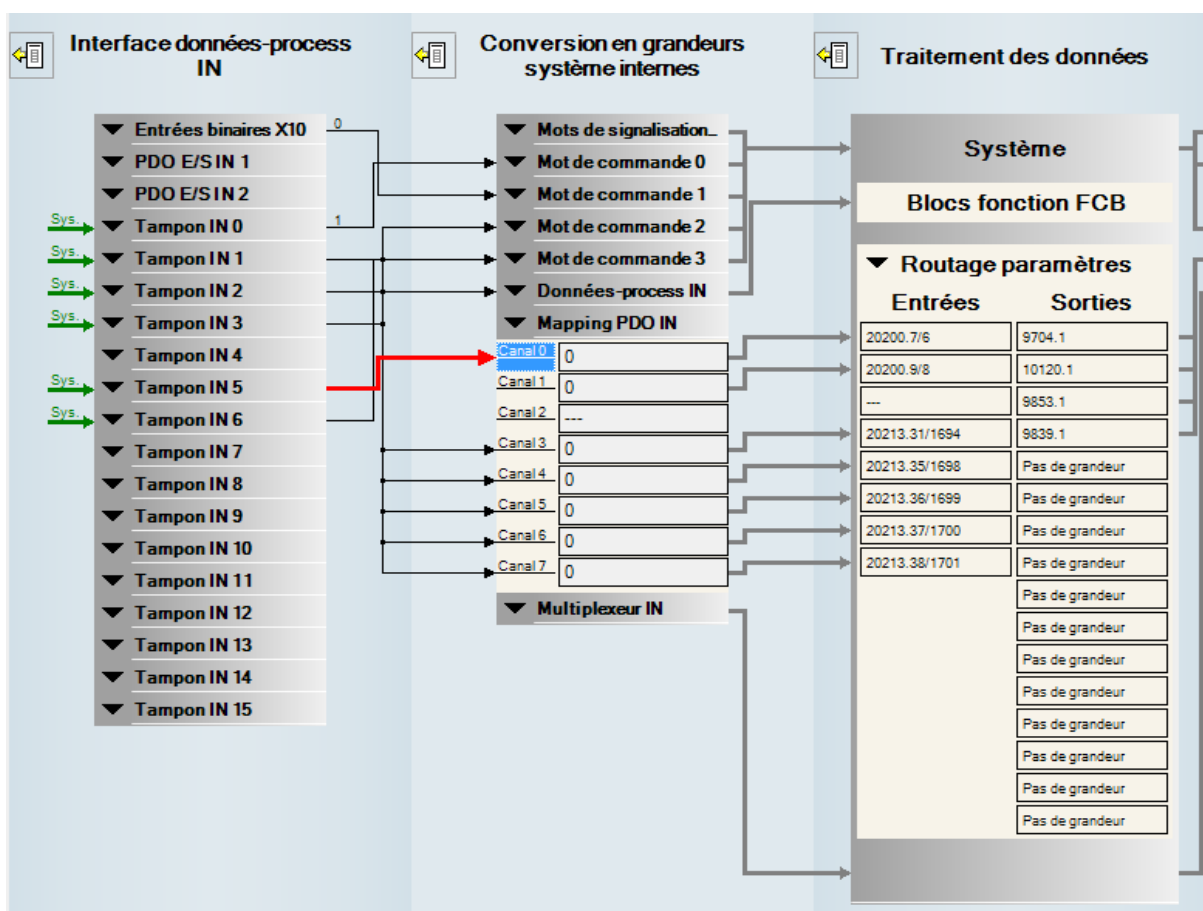
La structure de sortie "MC_GEAR_EXTENDED_DIAG_MX" est prévue comme information de diagnostic supplémentaire de "MC_LinkTecGear_MX".

Pour l'instant ceci est préparé, mais non réalisé.

2.4.3 Configuration PDO

En plus de la configuration effectuée par "MC_ConnectAxis_MX", le bloc fonction "MC_LinkTecGear_MX" effectue également certains réglages sur l'interface données-process.

Même s'il n'est pas nécessaire d'adapter la configuration PDO pour les applications standard, l'illustration suivante donne un aperçu des réglages spécifiques Gear effectués par "MC_LinkTecGear_MX".



63331axx



Le Mapping PDO IN est configuré pour le transfert des paramètres pour l'offset de rattrapage en vitesse/temps (réalisé à l'aide du générateur de profil de la came électronique).

Par ailleurs la liaison du tampon IN 5 vers les variables technologiques correspondantes (DDB) pour la réception de la position maître via le bus CAN est créée ici avec le canal 0. Cet objet est réglé comme valeur maître pour la fonction Gear par la sélection de "MX_GEAR_RECEIVE_PDO_1" comme source maître du bloc fonction "MC_SetGearConfig_MX".

Le canal 1 est configuré de sorte qu'une variable technologique soit réglée comme cible ; cette variable représente la source maître sous forme d'objet "MX_GEAR_RECEIVE_PDO_2" du "MC_SetGearConfig_MX". La source pour le canal 1 peut être choisie et configurée librement par l'utilisateur dans les tampons IN 11 à 15.

Les canaux du Mapping PDO IN se répartissent de la manière suivante :

- Canal 0 : position maître du tampon IN 5
- Canal 1 : position maître configurable, tampons IN 11 à 15
- Canal 2 : réservé
- Canal 3 : OffsetDistance
- Canaux 4 / 5 : OffsetSpeed
- Canal 6 : OffsetAcceleration
- Canal 7 : OffsetDeceleration

Réglages dans le tampon IN 5

63332axx



Le tampon IN 5 est configuré pour la réception de la position maître via le bus CAN. Pour cela, le bloc fonction "MC_LinkTecGear_MX" règle de manière figée la réception synchronisée des données ainsi que le format des données / la longueur de bloc de données.

La source de données et l'ID du message peuvent être réglées via le bloc fonction "MC_InitialConfig_MX" (de la bibliothèque MPLCMotion_MX).

Sans *InitialConfig*, la source de données configurée en standard est "Bus système CAN1" et l'ID de message 2.

Le bloc fonction "MC_InitialConfig_MX" offre entre autres les possibilités suivantes via la structure d'entrée "ExtendedConfiguration_MX" :

- La variable structure "SyncSource" permet de définir quel bus CAN synchronisé doit être utilisé comme source de données pour le tampon IN 5. Les choix suivants sont possibles :
 - MX_SYNCSOURCE_NONE
 - MX_SYNCSOURCE_CAN1
 - MX_SYNCSOURCE_CAN2
 - MX_SYNCSOURCE_COMOPTION
- L'ID message est configurée via la variable *ReceiveID*, avec les possibilités de choix suivantes :
 - MX_VIRTUAL_ENCODER_ID1 (ID 2)
 - MX_VIRTUAL_ENCODER_ID2 (ID 3)
 - MX_PDO_ID1 (ID 4)
 - MX_PDO_ID2 (ID 5)
 - MX_PDO_ID3 (ID 6)
 - MX_PDO_ID4 (ID 7)
 - MX_PDO_ID5 (ID 8)
 - MX_PDO_ID6 (ID 9)
 - MX_PDO_ID7 (ID 10)
 - MX_PDO_ID8 (ID 15)
 - MX_PDO_ID9 (ID 16)
 - MX_PDO_ID10 (ID 17)
 - MX_PDO_ID11 (ID 18)
 - MX_PDO_ID12 (ID 23)

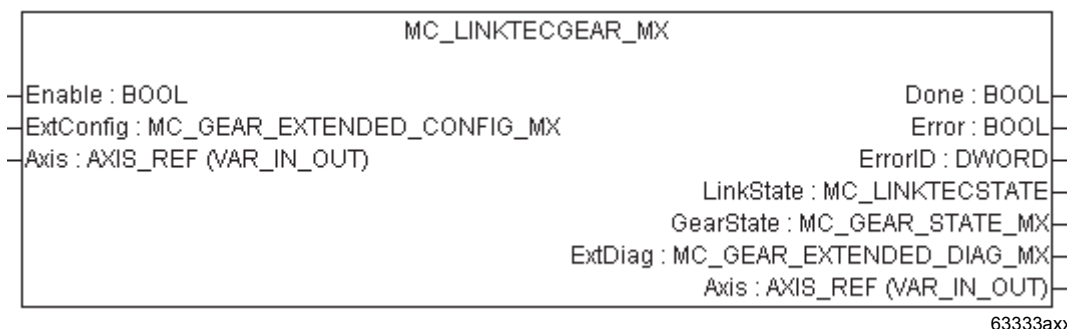
Autre réglage spécifique Gear

- Le tampon IN 6 est configuré pour la réception du mot de commande 3. Les bits de ce mot de commande sont réglés pour le pilotage de fonctions internes intégrées aux fonctions technologiques.
- Le tampon Out 4 contient le mot d'état 3, qui délivre les informations d'état des fonctions technologiques respectives nécessaires aux blocs fonction de la bibliothèque "MPLCTecGearMotion_MX".



2.5 Les blocs fonction de la bibliothèque MPLCTecGearMotion_MX

2.5.1 MC_LinktecGear_MX



63333axx

Description

Le bloc fonction "MC_LinkTecGear_MX" constitue l'interface logique entre l'axe et la fonction technologique Gear. Lorsque le bloc fonction "ConnectAxis" a établi la communication cyclique avec l'axe et que la libération est donnée par l'entrée "Enable", le système configure d'abord les différentes fonctions technologiques MOVIAXIS® et celles nécessaires pour la fonction Gear. L'interface PDO est ensuite configurée (IN-PDO 5 pour la réception de la position maître via le SBus, configuration PDO pour les valeurs d'offset), et les paramètres de communication pour la synchronisation du SBus réglés.

En mode de fonctionnement cyclique, le bloc fonction "MC_LinkTecGear_MX" assure la commutation de la fonction Gear sur une courbe linéaire dès que l'axe est synchronisé et se déplace de façon synchrone (nécessaire afin de rendre possible un déplacement d'offset).

Diverses informations d'état sont également mises à disposition comme sorties.

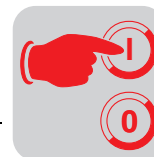
Attention :

Le bloc fonction "MC_LinkTecGear_MX" est indispensable pour l'utilisation des fonctionnalités de synchronisation et doit être ouvert cycliquement dans le programme.

A l'état *LinkState* = NotLinked, il n'est possible d'exécuter aucun bloc fonction de la bibliothèque "MPLCTecGearMotion_MX" ; un message de défaut correspondant apparaît.

Condition :

MOVI-PLC® *advanced* en version technologique T1 ou supérieure.

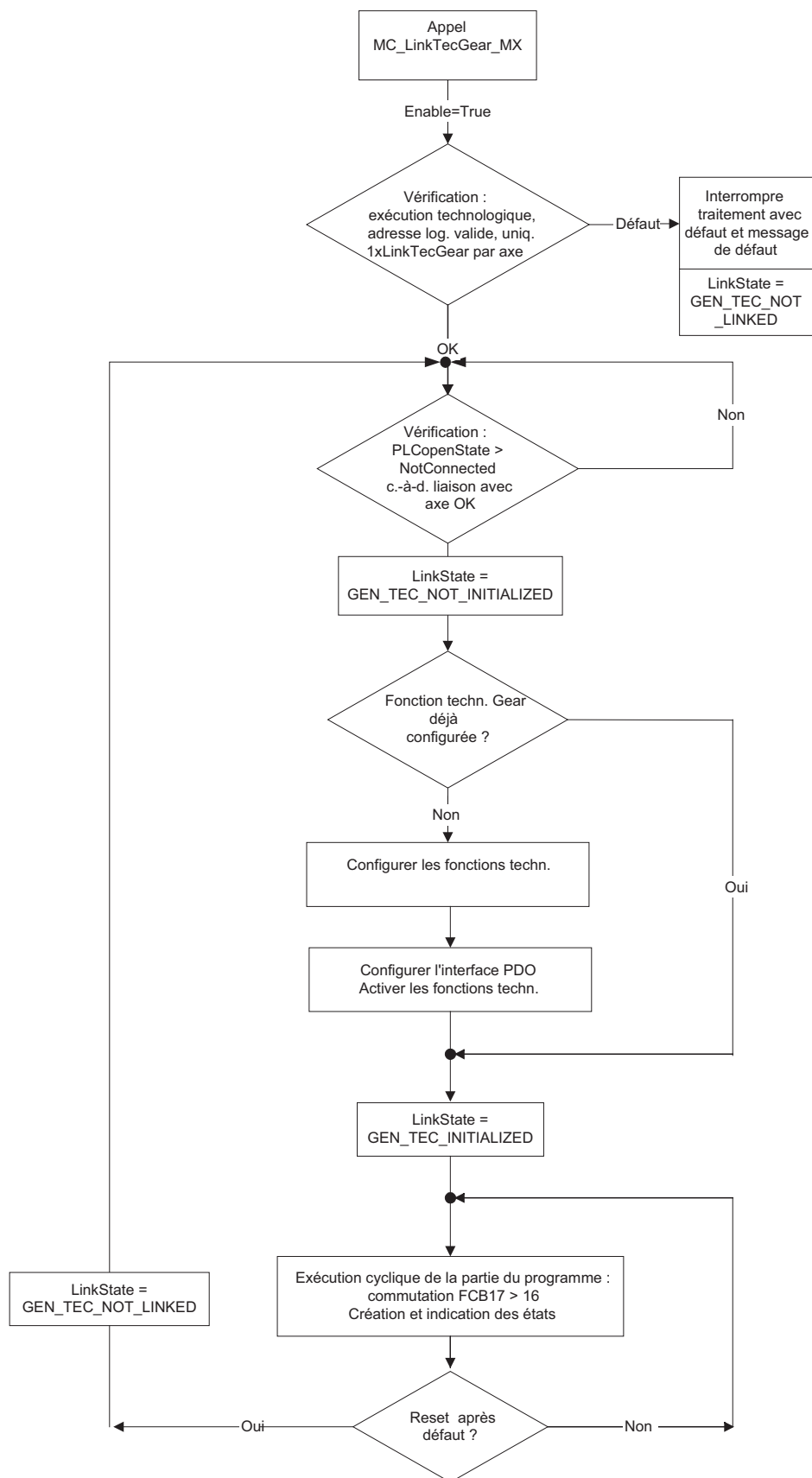


Entrées :

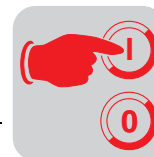
Nom	Type	Signification
Enable	BOOL	Le bloc fonction est traité tant que Enable = "true". L'initialisation est effectuée sur un front montant.
ExtConfig	MC_GEAR_EXTENDED_CONFIG_MX	Configuration avancée, voir page 22
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

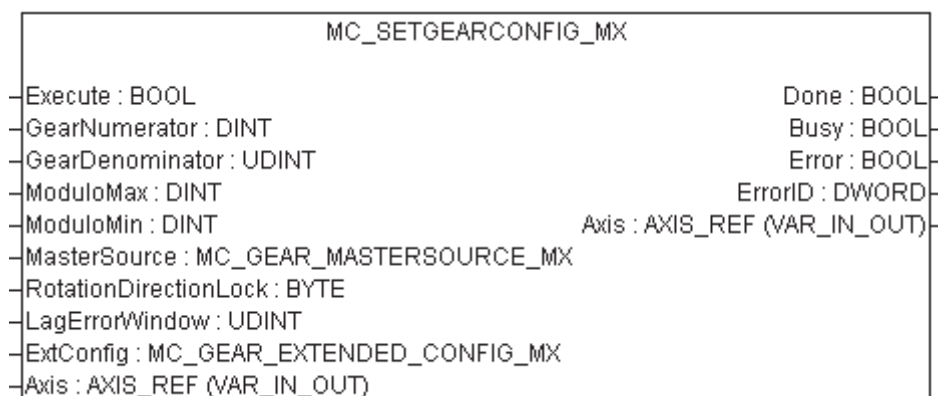
Nom	Type	Signification
Done	BOOL	Initialisation effectuée avec succès, liaison avec fonction technologique Gear établie.
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 54
LinkState	MC_LINKTECSTATE	Etat actuel : GEN_TEC_NOTLINKED (pas de liaison) GEN_TEC_NOTINITIALISED (initialisation non réalisée) GEN_TEC_INITIALISED (fonction technologique Gear initialisée)
GearState	MC_GEAR_STATE_MX	Etat Gear actuel : MX_GEAR_OUTGEAR (désolidarisé) MX_GEAR_ENGAGING_GEAR_IN (mise en synchronisation) MX_GEAR_IN_GEAR (fonctionnement synchronisé) MX_GEAR_OFFSET_GEAR (fonctionnement en offset) MX_GEAR_MOVE_TO_MASTER L'esclave se déplace vers le maître après ARRET D'URGENCE ou verrouillage MX_GEAR_NOTLINKED (pas d'état Gear valide, p. ex. si LinkState = NotLinked)
ExtDiag	MC_GEAR_EXTENDED_DIAG_MX	Prêt pour plus d'informations concernant le diagnostic

**MC_LinktecGear_MX : Initialisation**

63334afr



2.5.2 MC_SetGearConfig_MX



63335axx

Description

Ce bloc fonction règle les paramètres de base de la fonction Gear et les transmet à l'axe concerné. Les valeurs réglées sont prises en compte et envoyées lorsqu'un front montant est appliqué sur l'entrée Execute.

L'entrée *MasterSource* permet d'indiquer la source du signal codeur maître. Il est possible d'opter soit pour des codeurs "réels" raccordés sur l'axe, soit pour des signaux maître "virtuels" envoyés via SBus (voir page 20).

Si le signal maître est transmis au format modulo, régler les entrées ModuloMax / ModuloMin aux valeurs limite du signal maître. Ceci permet d'éviter les sauts de consigne en cas de dépassement de la plage des Modulo. En cas de signal maître linéaire, le réglage de ces valeurs est libre.

Il est possible de transmettre une structure pour le réglage de divers filtres et dispositifs de compensation au niveau de l'entrée *ExtConfig*. Ceci permet p. ex. de filtrer un signal maître "perturbé". Ces filtres sont réglés en interne sur des valeurs standard adéquates, si bien qu'il n'est généralement pas nécessaire de brancher cette entrée.

Condition :

Le bloc fonction "MC_LinkTecGear_MX" a configuré la fonction technologique Gear, l'état *LinkState* ne doit pas passer à GEN_TEC_NOTLINKED.

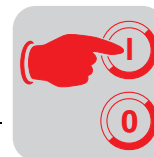


Entrées :

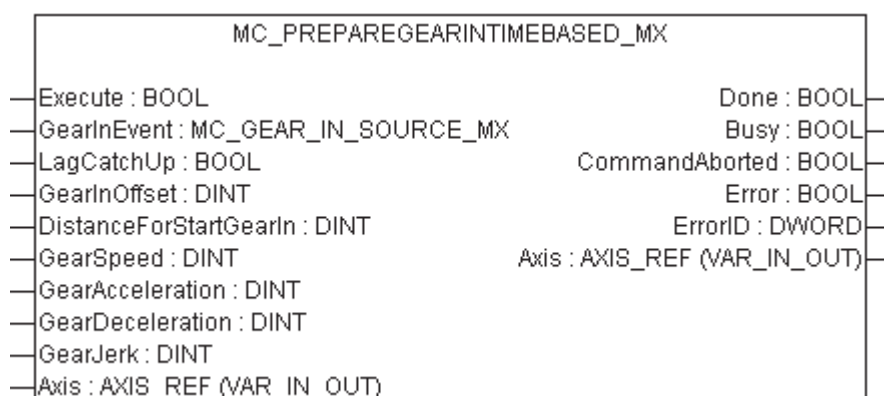
Nom	Type	Signification
Execute	BOOL	Cette entrée lance l'exécution du bloc fonction sur un front montant
GearNumerator	DINT	Numérateur pour la mise à l'échelle du signal maître.
GearDenominator	UDINT	Dénominateur pour la mise à l'échelle du signal maître.
ModuloMax	DINT	Valeur modulo maximale signal maître
ModuloMin	DINT	Valeur modulo minimale signal maître
MasterSource	MC_GEAR_MASTERSOURCE_MX	Source de la position maître MX_GEAR_SYSTEMPOS_ENCODER1 MX_GEAR_SYSTEMPOS_ENCODER2 MX_GEAR_SYSTEMPOS_ENCODER3 MX_GEAR_MODULOPOS_ENCODER1 MX_GEAR_MODULOPOS_ENCODER2 MX_GEAR_MODULOPOS_ENCODER3 MX_GEAR_MODULO_VIRTUAL_ENCODER_MX MX_GEAR_LINEAR_VIRTUAL_ENCODER_MX MX_GEAR_RECEIVE_PDO_1 MX_GEAR_RECEIVE_PDO_2
RotationDirectionLock	BYTE	Verrouillage sens de marche 0 = libération des deux sens 1= libération unq. gauche 2 = libération unq. droite
LagErrorWindow	UDINT	Tolérance erreur de poursuite en [inc]
ExtConfig	MC_GEAR_EXTENDED_CONFIG_MX	Configuration avancée : réglages des filtres, dispositifs de compensation, réaction défaut erreur de poursuite, fonction "MoveToMaster", voir page 22
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
Done	BOOL	Tous les paramètres ont été transmis correctement
Busy	BOOL	Transfert de paramètres en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 54



2.5.3 MC_PrepareGearInTimeBased_MX



63336axx

Description

Ce bloc fonction permet de régler la vitesse, l'accélération, la décélération et le jerk pour la synchronisation en vitesse/temps.

Par ailleurs l'entrée *LagCatchUp* permet de définir si une erreur de poursuite présente doit être supprimée ou non à la mise en synchronisation. Il est également possible d'activer en cours de synchronisation un offset supplémentaire via la valeur *GearInOffset* indiquée pour la distance.

L'événement de synchronisation est défini via *GearInEvent* : avec le réglage "MX_GEAR_DISTANCE_COUNTER", la valeur de l'entrée *DistanceForStartGearIn* règle la distance du maître lorsque la synchronisation a commencé.

Attention :

La valeur *DistanceForStartGearIn* de la trajectoire du maître pour la synchronisation est liée à un sens, c.-à-d. qu'en cas de rotation à gauche du maître, cette valeur doit être négative. Dans le cas de valeurs extrêmement petites, la prise en compte de l'événement n'est assurée que si le maître est à l'arrêt. La trajectoire du maître peut dans certains cas se modifier si rapidement que la durée de scrutation est supérieure à la durée nécessaire pour que la valeur soit atteinte.

Attention :

Si la fonction "LagCatchUp" est activée et si une prise de référence a été effectuée avant la synchronisation, le système peut déclencher une course de rattrapage pendant la synchronisation, même s'il n'y avait au préalable aucune différence entre les positions maître et esclave.

La raison en est la suivante : suite à la prise de référence, l'esclave a une nouvelle position actuelle, c'est pourquoi la différence entre la position maître et esclave est également modifiée et fait donc l'objet d'un rattrapage.

Pour éviter cela, il est possible de désactiver le rattrapage avant la nouvelle synchronisation en appelant une nouvelle fois la fonction "MC_PrepareGearInTimeBased_MX" par *LagCatchUp* = False. Il est également possible de supprimer la différence maître / esclave à l'aide du bloc fonction "MC_GearClearLag_MX".

Condition :

Le bloc fonction "MC_LinkTecGear_MX" a configuré la fonction technologique Gear, *LinkState* ne doit pas passer à GEN_TEC_NOTLINKED.

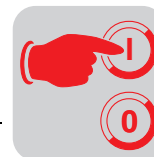


Entrées :

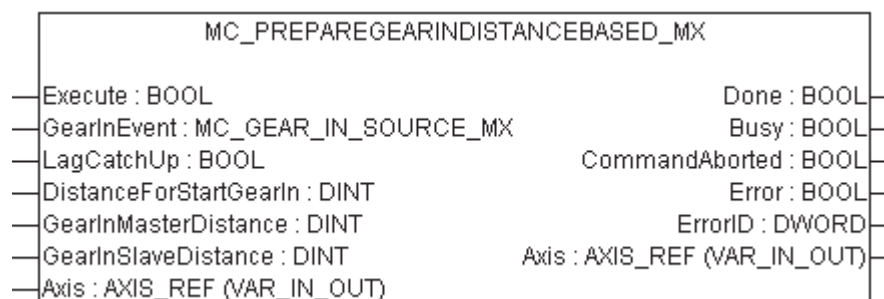
Nom	Type	Signification
Execute	BOOL	Cette entrée sert à lancer l'exécution du bloc fonction.
GearInEvent	MC_GEAR_IN_SOURCE_MX	<p>Les événements de synchronisation possibles sont :</p> <p>MX_GEAR_DIRECTMX_GEAR_INTERRUPT_DI01_POS_FLAG MX_GEAR_INTERRUPT_DI01_NEG_FLAG MX_GEAR_INTERRUPT_DI01 ... MX_GEAR_INTERRUPT_DI08_POS_FLAG MX_GEAR_INTERRUPT_DI08_NEG_FLAG MX_GEAR_INTERRUPT_DI08 MX_GEAR_INTERRUPT_C_TRACK_ENC1 MX_GEAR_INTERRUPT_C_TRACK_ENC2 MX_GEAR_INTERRUPT_C_TRACK_ENC3 MX_GEAR_DISTANCE_COUNTER MX_GEAR_DISTANCE_COUNTER_RESET_COUNT</p> <p>Explication : DIRECT = synchronisation directe POS_FLAG = front positif NEG_FLAG = front négatif uniq. l'entrée = les deux fronts DI01 de MOVIAXISDI01 = entrée DI02 de MOVIAXISDI02 = entrée DI03 de MOVIAXISDI03 = entrée DI04 de MOVIAXISDI04 = entrée DI05 de MOVIAXISDI05 = entrée DI06 de MOVIAXISDI06 = entrée DI07 de MOVIAXISDI07 = entrée DI08 de MOVIAXISDI08 = entrée C_GEAR_ENC1 = voie C codeur 1 C_GEAR_ENC2 = voie C codeur 2 C_GEAR_ENC3 = voie C codeur 3</p>
LagCatchUp	BOOL	True : erreur de poursuite est supprimée. False : erreur de poursuite n'est pas supprimée (synchronisation indexée sur la position maître).
GearInOffset	DINT	Offset de position pour la synchronisation
DistanceForStartGearIn	DINT	Uniquement pour Mode Distance_Counter : déplacement du maître servant pour la synchronisation
GearSpeed	DINT	Vitesse de synchronisation maximale en [tr/min]
GearAcceleration	DINT	Accélération synchronisation en [tr/min/s]
GearDeceleration	DINT	Décélération synchronisation en [tr/min/s]
GearJerk	DINT	Jerk de synchronisation en [tr/min/s²]
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
Done	BOOL	Tous les paramètres ont été transmis correctement
Busy	BOOL	Transfert de paramètres en cours
CommandAborted	BOOL	Le bloc fonction a été interrompu par une autre instance d'un bloc fonction PrepareGearIn
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 54



2.5.4 MC_PrepareGearInDistanceBased_MX



63337axx

Description

Ce bloc fonction sert à préparer la synchronisation par rattrapage en distance/temps. L'esclave se synchronise sur le maître à l'intérieur de la trajectoire *GearInMasterDistance* en parcourant la distance *GearInSlaveDistance*.

L'entrée *LagCatchUp* permet par ailleurs de définir si une erreur de poursuite présente doit être supprimée ou non en cas de mise en synchronisation.

L'événement de synchronisation est défini via *GearInEvent* ; en cas de réglage "MX_GEAR_DISTANCE_COUNTER", la valeur de l'entrée *DistanceForStartGearIn* détermine le réglage de la trajectoire du maître après lancement de la mise en synchronisation.

Attention :

La valeur *DistanceForStartGearIn* ou *GearInMasterDistance* (trajectoire maître pour synchronisation) de la trajectoire du maître pour la synchronisation est liée à un sens, c.-à-d. qu'en cas de rotation à gauche du maître, ces valeurs doivent par conséquent être négatives.

MX_GEAR_DISTANCE_COUNTER :

Dans le cas de valeurs extrêmement petites, la prise en compte de l'événement n'est assurée que si le maître est à l'arrêt. La trajectoire du maître peut dans certains cas se modifier si rapidement que la durée de scrutation est supérieure à la durée nécessaire pour que la valeur *DistanceForStartGearIn* soit atteinte.

Le calcul du profil de synchronisation s'effectue selon un polynôme 5^{ème} degré.

Par conséquent, la distance *GearInSlaveDistance* doit être égale à environ la moitié de la trajectoire mise à l'échelle du maître, sans quoi l'esclave risquerait de dépasser la position cible ou de modifier brièvement son sens de rotation durant la phase de synchronisation (voir aussi page 11).

Attention :

Si la fonction "LagCatchUp" est activée et si une prise de référence a été effectuée avant la synchronisation, le système peut déclencher une course de rattrapage pendant la synchronisation, même s'il n'y avait au préalable aucune différence entre les positions maître et esclave.

La raison en est la suivante : suite à la prise de référence, l'esclave a une nouvelle position actuelle, c'est pourquoi la différence entre la position maître et esclave est également modifiée et fait donc l'objet d'un rattrapage.

Pour éviter cela, il est possible de désactiver le rattrapage avant la nouvelle synchronisation en appelant une nouvelle fois la fonction "MC_PrepareGearInTimeBased_MX" par *LagCatchUp* = False. Il est également possible de supprimer la différence maître / esclave à l'aide du bloc fonction "MC_GearClearLag_MX".

Condition :

"MC_LinkTecGear_MX" a configuré la fonction technologique Gear, *LinkState* ne doit pas passer à "GEN_TEC_NOTLINKED".

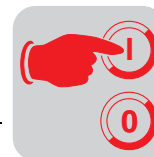


Entrées :

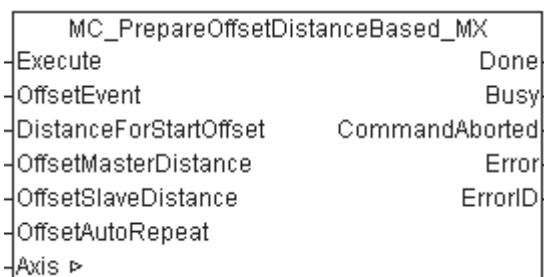
Nom	Type	Signification
Execute	BOOL	Cette entrée lance l'exécution du bloc fonction.
GearInEvent	MC_GEAR_IN_SOURCE_MX	<p>Les événements de synchronisation possibles sont :</p> <p>MX_GEAR_DIRECT MX_GEAR_INTERRUPT_DI01_POS_FLAG MX_GEAR_INTERRUPT_DI01_NEG_FLAG MX_GEAR_INTERRUPT_DI01 ... MX_GEAR_INTERRUPT_DI08_POS_FLAG MX_GEAR_INTERRUPT_DI08_NEG_FLAG MX_GEAR_INTERRUPT_DI08 MX_GEAR_INTERRUPT_C_TRACK_ENC1 MX_GEAR_INTERRUPT_C_TRACK_ENC2 MX_GEAR_INTERRUPT_C_TRACK_ENC3 MX_GEAR_DISTANCE_COUNTER MX_GEAR_DISTANCE_COUNTER_RESET_COUNT</p> <p>Explication :</p> <p>DIRECT = synchronisation directe POS_FLAG = front positif NEG_FLAG = front négatif uniq. l'entrée = les deux fronts DI01 de MOVIAXISDI01 = entrée DI02 de MOVIAXISDI02 = entrée DI03 de MOVIAXISDI03 = entrée DI04 de MOVIAXISDI04 = entrée DI05 de MOVIAXISDI05 = entrée DI06 de MOVIAXISDI06 = entrée DI07 de MOVIAXISDI07 = entrée DI08 de MOVIAXISDI08 = entrée C_GEAR_ENC1 = voie C codeur 1 C_GEAR_ENC2 = voie C codeur 2 C_GEAR_ENC3 = voie C codeur 3</p>
LagCatchUp	BOOL	True : l'erreur de poursuite est supprimée. False : l'erreur de poursuite n'est pas supprimée (synchronisation indexée sur la position maître)
DistanceForStartGearIn	DINT	Uniquement pour Mode Distance_Counter : trajectoire du maître après synchronisation
GearInMasterDistance	DINT	Distance maître dans laquelle s'effectue la synchronisation
GearInSlaveDistance	DINT	Trajectoire esclave durant la phase de synchronisation
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
Done	BOOL	La synchronisation est préparée correctement.
Busy	BOOL	Transfert de paramètres en cours
CommandAborted	BOOL	Le bloc fonction a été interrompu par une autre instance d'un bloc fonction PrepareGearIn.
Error	BOOL	Défaut dans le bloc fonction durant l'exécution.
ErrorID	UINT	Voir page 54



2.5.5 MC_PrepareOffsetDistanceBased_MX



63340axx

Description

Ce bloc fonction sert à préparer un offset en rattrapage distance. L'entrée *OffsetMasterDistance* est utilisée pour définir la trajectoire maître qui sera utilisée pour appliquer l'offset, avec *OffsetSlaveDistance* comme trajectoire esclave supplémentaire.

L'événement de lancement d'offset est défini via le bloc *OffsetEvent* ; avec le réglage "MX_GEAR_DISTANCE_COUNTER", la valeur de l'entrée *DistanceForStartOffset* règle la trajectoire du maître lorsque le déplacement d'offset a commencé. L'entrée *OffsetAutoRepeat* permet de sélectionner la répétition automatique de l'activation d'un mode d'offset géré par interruption ou par compteur.

Attention :

La valeur *DistanceForStartOffset* ou *GearInMasterDistance* (trajectoire maître pour synchronisation) de la trajectoire du maître pour le lancement de l'offset est liée à un sens, c.-à-d. qu'en cas de rotation à gauche du maître, ces valeurs doivent par conséquent être négatives.

MX_GEAR_DISTANCE_COUNTER :

Dans le cas de valeurs extrêmement petites, la prise en compte de l'événement n'est assurée que si le maître est à l'arrêt. La trajectoire du maître peut dans certains cas se modifier si rapidement que la durée de scrutation est supérieure à la durée nécessaire pour que la valeur *DistanceForStartOffset* soit atteinte.

Condition :

"MC_LinkTecGear_MX" a configuré la fonction technologique Gear, *LinkState* ne doit pas passer à "GEN_TEC_NOTLINKED".

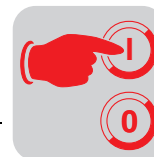


Entrées :

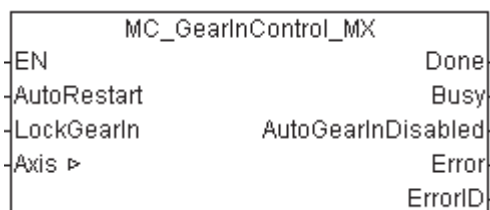
Nom	Type	Signification
Execute	BOOL	Cette entrée lance l'exécution du bloc fonction.
OffsetEvent	MC_GEAR_IN_SOUR CE_MX	<p>Les événements de lancement d'offset possibles sont :</p> <p>MX_GEAR_DIRECT MX_GEAR_INTERRUPT_DI01_POS_FLAG MX_GEAR_INTERRUPT_DI01_NEG_FLAG MX_GEAR_INTERRUPT_DI01 ... MX_GEAR_INTERRUPT_DI08_POS_FLAG MX_GEAR_INTERRUPT_DI08_NEG_FLAG MX_GEAR_INTERRUPT_DI08 MX_GEAR_INTERRUPT_C_TRACK_ENC1 MX_GEAR_INTERRUPT_C_TRACK_ENC2 MX_GEAR_INTERRUPT_C_TRACK_ENC3 MX_GEAR_DISTANCE_COUNTER</p> <p>Explication : DIRECT = synchronisation directe POS_FLAG = front positif NEG_FLAG = front négatif uniq. l'entrée = les deux fronts DI01 de MOVIAXISDI01 = entrée DI02 de MOVIAXISDI02 = entrée DI03 de MOVIAXISDI03 = entrée DI04 de MOVIAXISDI04 = entrée DI05 de MOVIAXISDI05 = entrée DI06 de MOVIAXISDI06 = entrée DI07 de MOVIAXISDI07 = entrée DI08 de MOVIAXISDI08 = entrée C_GEAR_ENC1 = voie C codeur 1 C_GEAR_ENC2 = voie C codeur 2 C_GEAR_ENC3 = voie C codeur 3</p>
DistanceForStartOffset	DINT	Uniquement pour Mode Distance_Counter : distance maître après laquelle l'offset est lancé
OffsetMasterDistance	DINT	Distance maître sur laquelle l'offset est exécuté.
OffsetSlaveDistance	DINT	Distance esclave supplémentaire durant le déplacement d'offset
OffsetAutoRepeat	BOOL	Fonction de répétition des modes d'offset gérés par interruption / par compteur
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
Done	BOOL	La synchronisation est préparée correctement
Busy	BOOL	Transfert de paramètres en cours
CommandAborted	BOOL	Le bloc fonction a été interrompu par une autre instance d'un bloc fonction PrepareOffset ou par OffsetDirectDistance
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 54



2.5.6 MC_GearInControl_MX



63341axx

Description

Le bloc fonction "MC_GearInControl_MX" offre des fonctions de pilotage supplémentaires pour les événements de synchronisation gérés par interruption ou par compteur. L'entrée *AutoRestart* permet de répéter la synchronisation sans nouvelle préparation à l'aide du bloc fonction "Prepare-GearIn". La synchronisation gérée par interruption / compteur peut par ailleurs être verrouillée en forçant l'entrée *LockGearIn* sur "True". La synchronisation est également verrouillée après un défaut sur l'axe (Errorstop). Dans ce cas, il faudra réactiver la synchronisation gérée par interruption / compteur à l'aide des blocs fonction "PrepareGearIn". Le verrouillage de la synchronisation est affiché par la sortie *AutoGearInDisabled*.

Attention :

Le verrouillage de la synchronisation n'est possible qu'avec les événements de synchronisation gérés par interruption / compteur ; la synchronisation via "MC_GearInDirect_MX" reste possible.

Condition :

"MC_LinkTecGear_MX" a configuré la fonction technologique Gear, *LinkState* ne doit pas passer à "GEN_TEC_NOTLINKED".

Procédure :

Afin de pouvoir activer la fonction "AutoRestart", il faut préparer une fois l'événement de synchronisation à l'aide du bloc fonction "PrepareGearIn". Si l'*AutoRestart* est désactivé, la synchronisation gérée par interruption / compteur est verrouillée, jusqu'à une nouvelle préparation via le bloc fonction "PrepareGearIn".

Si *LockGearIn* = True, la synchronisation gérée par interruption / compteur est verrouillée et peut à nouveau être lancée par suppression du verrouillage (*LockGearIn* = False), sans nécessiter un nouvel appel des blocs fonction "PrepareGearIn".

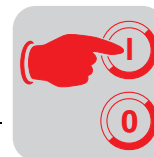


Entrées :

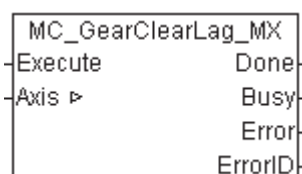
Nom	Type	Signification
EN	BOOL	Le bloc fonction est traité tant que cette entrée = True.
AutoRestart	BOOL	True : fonction répétition automatique de la synchronisation activée. False : synchronisation désactivée
LockGearIn	BOOL	True : synchronisation verrouillée. False : synchronisation libérée
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
Done	BOOL	Fonction souhaitée verrouillée
Busy	BOOL	Transfert de paramètres en cours
AutoGearInDisabled	BOOL	Synchronisation verrouillée
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 54



2.5.7 MC_GearClearLag_MX



63342axx

Description

Avec ce bloc fonction, la consigne Gear (position maître préparée) est mise à la même valeur que la mesure Gear (position esclave actuelle) par un front montant sur l'entrée Execute ; la différence maître-esclave est ainsi remise à zéro (une seule fois).

Ceci peut être nécessaire lorsque, pour la synchronisation, la fonction "Rattraper l'erreur de poursuite" (entrée *LagCatchUp* du bloc fonction "PrepareGearIn") est sélectionnée, mais que, pour une raison quelconque, l'erreur de poursuite ne doit pas être rattrapée (p. ex. après une prise de référence).

Si le rattrapage de l'erreur de poursuite est désactivé (*LagCatchUp* = False dans le bloc fonction "PrepareGearIn"), la synchronisation se fait par rapport à la position momentanée, c.-à-d. que "GearClearLag" n'est alors pas nécessaire.

Attention :

Le bloc fonction a besoin d'environ deux tours de cycle pour l'exécution ; c'est pourquoi il est recommandé de n'activer cette fonction que lorsque le maître / l'esclave est à l'arrêt.

Condition :

"MC_LinkTecGear_MX" a configuré la fonction technologique Gear, *LinkState* ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

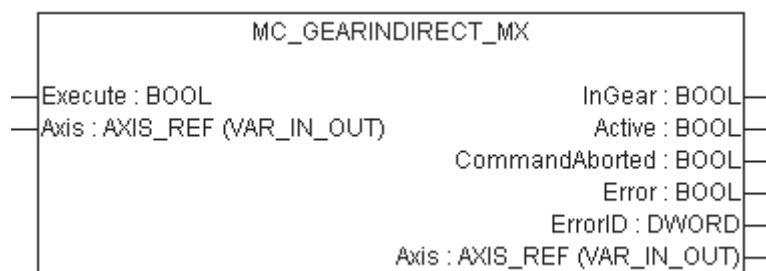
Nom	Type	Signification
Execute	BOOL	Cette entrée sert à lancer l'exécution du bloc fonction.
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
Done	BOOL	Différence maître-esclave mise à zéro
Busy	BOOL	Transfert de paramètres en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 54



2.5.8 MC_GearInDirect_MX



63343axx

Description

Un front montant sur l'entrée Execute active une synchronisation directe avec les paramètres qui ont été réglés via *PrepareGearIn*.

Ce bloc fonction interrompt les blocs "Move" et aussi "PrepareGearIn" encore actifs.

Condition :

"MC_LinkTecGear_MX" a configuré la fonction technologique Gear, *LinkState* ne doit pas passer à "GEN_TEC_NOTLINKED".

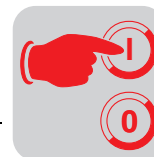
GearState = MX_GEAR_OUTGEAR

Entrées :

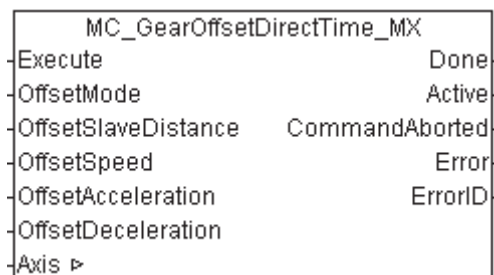
Nom	Type	Signification
Execute	BOOL	Cette entrée lance la mise en synchronisation.
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
InGear	BOOL	Synchronisation effectuée correctement, l'axe est synchronisé sur le maître.
Active	BOOL	L'axe est à l'état de synchronisation.
CommandAborted	BOOL	La tâche du bloc fonction a été interrompue par l'appel d'un autre bloc fonction Move.
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 54



2.5.9 MC_GearOffsetDirectTime_MX



63345axx

Description

Ce bloc fonction lance un offset par rattrapage en vitesse/temps avec les valeurs réglées. Après exécution de l'offset, l'axe repasse en mode synchronisé.

Le réglage du type d'offset *OffsetMode* permet de définir si la distance *OffsetSlaveDistance* doit être parcourue vers le maître sous forme de valeur relative ou absolue. Dans le cas d'un déplacement absolu, le point de référence est mis à zéro une seule fois, les autres valeurs d'offset se réfèrent à cette valeur de démarrage (tant que le statut "16" est affiché pour le MOVIAXIS®).

Condition :

"MC_LinkTecGear_MX" a configuré la fonction technologique Gear, *LinkState* ne doit pas passer à "GEN_TEC_NOTLINKED".

L'axe est en mode synchronisé (*GearState* = In_Gear).

Entrées :

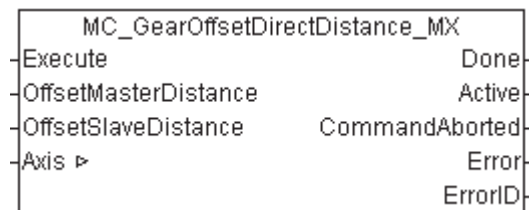
Nom	Type	Signification
Execute	BOOL	Cette entrée active le déplacement d'offset.
OffsetMode	MC_GEAR_OFFSETMODE_MX	Mode d'offset, réglage : MX_GEAR_OFFSET_RELATIVE MX_GEAR_OFFSET_ABSOLUTE
OffsetDistance	DINT	Offset que l'esclave doit parcourir
OffsetSpeed	DINT	Vitesse maximale pour le déplacement d'offset en [tr/min]
OffsetAcceleration	DINT	Accélération pour le déplacement d'offset en [tr/min/s]
OffsetDeceleration	DINT	Décélération pour le déplacement d'offset en [tr/min/s]
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	L'offset a été traité, l'axe est à nouveau synchronisé
Active	BOOL	Gestion d'offset en cours
CommandAborted	BOOL	La tâche du bloc fonction a été interrompue par l'appel d'un autre bloc fonction Move
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 54



2.5.10 MC_GearOffsetDirectDistance_MX



63346axx

Description

Ce bloc fonction permet de lancer un déplacement d'offset avec les valeurs réglées pour *OffsetMasterDistance* et *OffsetSlaveDistance*. Après exécution de l'offset, l'axe repasse en mode synchronisé.

Condition :

"MC_LinkTecGear_MX" a configuré la fonction technologique Gear, *LinkState* ne doit pas passer à "GEN_TEC_NOTLINKED".

L'axe est en mode synchronisé (*GearState* = In_Gear).

Entrées :

Nom	Type	Signification
Execute	BOOL	Cette entrée active le déplacement d'offset.
OffsetMasterDistance	DINT	Distance maître sur laquelle l'offset est exécuté.
OffsetSlaveDistance	DINT	Trajectoire esclave supplémentaire pendant la gestion d'offset
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
Done	BOOL	L'offset a été traité, l'axe est à nouveau synchronisé
Active	BOOL	Gestion d'offset en cours
CommandAborted	BOOL	La tâche du bloc fonction a été interrompue par l'appel d'un autre bloc fonction Move
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 54



2.6 Exemples

2.6.1 1. Le maître est un codeur virtuel, synchronisation directe par rattrapage en vitesse / temps

L'exemple de projet simple "AxisControl_MX_Technology" suivant décrit une mise en synchronisation simple avec synchronisation directe par rattrapage en vitesse/temps, sans rattrapage d'une erreur de poursuite.

Le codeur virtuel de la bibliothèque "MPLCTecVirtualEncoder" doit servir de maître.

Remarque :

Seules les séquences de programme impliquées dans la fonction Gear sont illustrées et expliquées ci-après. Les exemples de projets "AxisControl...." sont décrits en détail dans la documentation correspondant à ces exemples.

Solution :

1. Les structures d'entrée pour le réglage des paramètres Gear* pour le bloc fonction "MC_SetGearConfig_MX", ainsi que les valeurs de synchronisation pour le bloc "MC_PrepareGearInTimeBased_MX" sont configurés dans la routine d'initialisation :

```
IF NOT bInit THEN
  (*#####*)
  Initialisation of AxisControlVirtual
  (*#####*)

  (* Configuration virtual encoder *)
  gVirtualEncoderConfiguration.StartRisingEdge := FALSE;

  gVirtualEncoderConfiguration.LinkTecAxisVirtual.CanNode := SBUS_NODE_1; (* SBUS_NODE_1
                                                                           SBUS_NODE_2 *)

  gVirtualEncoderConfiguration.LinkTecAxisVirtual.SendObjectID := MDX_VIRTUAL_ENCODER_ID1;
                                                                           (* name of the message which is used to transmit
                                                                           MDX_VIRTUAL_ENCODER_ID1
                                                                           MDX_VIRTUAL_ENCODER_ID2 *)

  (*#####*)
  Initialisation of AxisControl axis 3 (gear motion)
  (*#####*)

  (* general axis control parameters *)
  gAxisConfiguration_MX[3].StartRisingEdge := FALSE;

  (* configuration modulo parameter --> only necessary if modulo mode is used *)
  gAxisConfiguration_MX[3].SetModuloParameters.Overflow := 1;
  gAxisConfiguration_MX[3].SetModuloParameters.Underflow := 1;

  (* Configuration gear parameters *)
  gAxisConfiguration_MX[3].SetGearConfig.GearNumerator := 1; (* Numerator for scaling of the master increments *)
  gAxisConfiguration_MX[3].SetGearConfig.GearDenominator := 1; (* Denominator for scaling of the master increments *)
  gAxisConfiguration_MX[3].SetGearConfig.ModuloMax := 0; (* Modulo maximum Value of the master position *)
  gAxisConfiguration_MX[3].SetGearConfig.ModuloMin := 0; (* Modulo minimum Value of the master position *)
  gAxisConfiguration_MX[3].SetGearConfig.MasterSource := MX_GEAR_RECEIVE_PDO_1;
  gAxisConfiguration_MX[3].SetGearConfig.LagErrorWindow := 10000; (* lag error window in [inc] *)
  gAxisConfiguration_MX[3].SetGearConfig.RotationDirectionLock := 0; (* 0 = both directions, 1 = only CW enabled, 2 = only CCW enabled *)

  gAxisConfiguration_MX[3].PrepareGearIn.GearSpeed := 1000; (* gear in speed in [rpm] *)
  gAxisConfiguration_MX[3].PrepareGearIn.GearAcceleration := 10000; (* gear in acceleration in [rpm/s] *)
  gAxisConfiguration_MX[3].PrepareGearIn.GearDeceleration := 10000; (* gear in deceleration in [rpm/s] *)

  gAxisConfiguration_MX[3].SetGearConfig.ExtConfig.MSignal_InterpolationTime := 10; (* Interpolation time: range 1...60 means 0.5...30ms filter time *)
  gAxisConfiguration_MX[3].SetGearConfig.ExtConfig.AverageFilterTime := 10; (* AverageFilter: range 1...60 means 0.5...30ms filter time *)

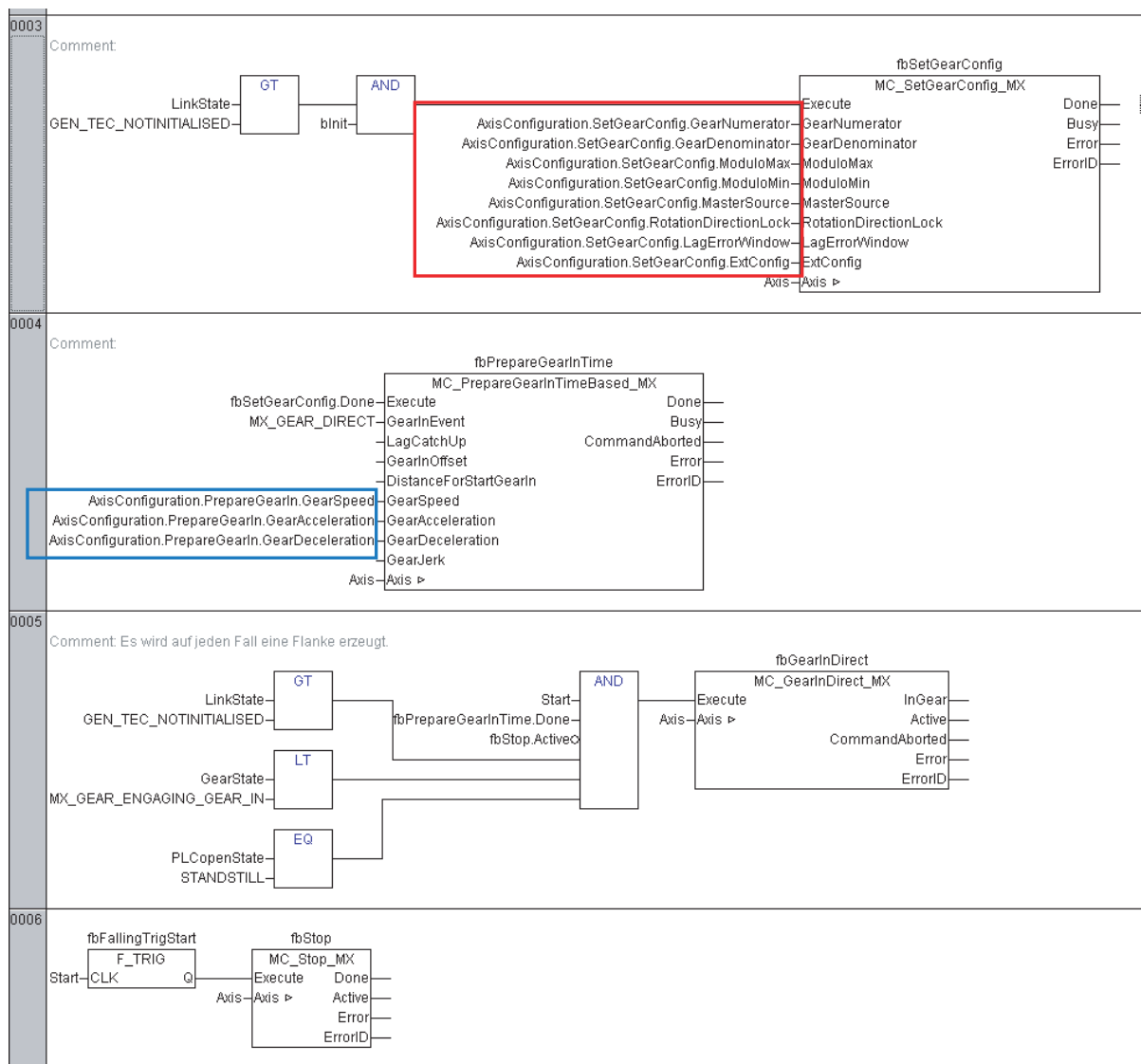
  bInit:=TRUE;
END_IF;
```

63347axx

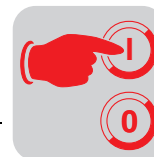


2. Lors de l'appel de AxisMode "Gearing", le système règle d'abord les principaux paramètres Gear, puis les valeurs concernant la synchronisation. La synchronisation s'effectue à l'aide du bloc fonction "MC_GearInDirect_MX", si tout est à l'état "prepared" et si *Start* = True.

Pour mettre fin au mode synchronisé, un front descendant sur Start permet d'exécuter un arrêt aux limites application ; le statut de l'entraînement passe à "26". Lorsque l'entraînement s'est arrêté, le bloc "PLCopenState" passe à l'état "Standstill".



63349axx



2.6.2 2. MOVIAXIS® maître, synchronisation par interruption DI02

Un MOVIAXIS® doit être configuré comme maître pour un esclave Gear MOVIAXIS®. La synchronisation doit s'effectuer par un front montant sur l'entrée DI02.

Afin de réduire la charge du bus système CAN1, sur lequel transite l'ensemble de la communication, l'envoi de la position de codeur 1 de l'axe maître s'effectue via le bus CAN2 synchronisé.

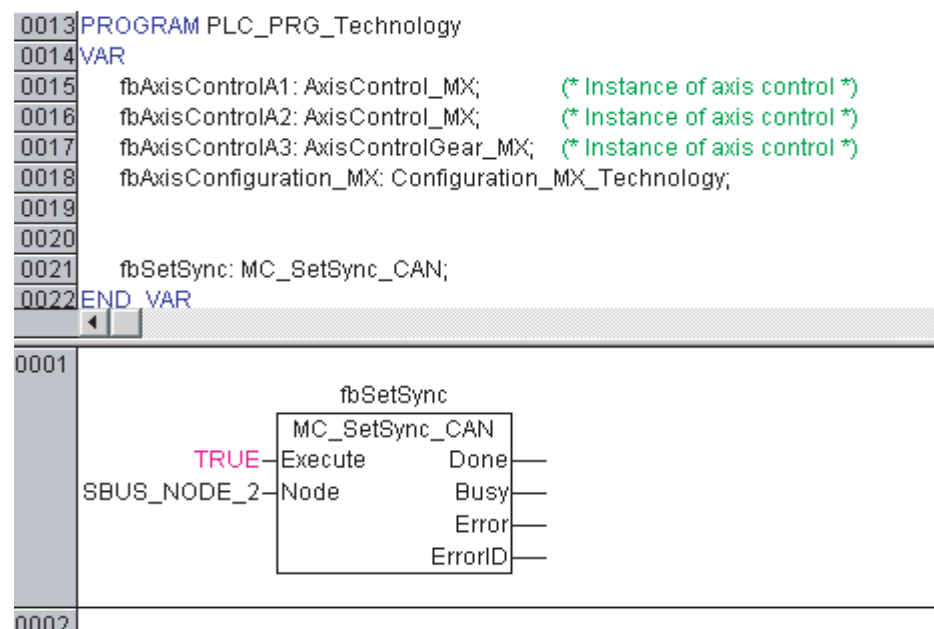
Dans ce cas également l'exemple de projet "AxisControl_MX_Technology" sert de base et est modifié de manière adéquate en fonction des nécessités.

Remarque :

Seules les séquences de programme impliquées dans la fonction Gear sont illustrées et expliquées ci-après. Les exemples de projets "AxisControl...." sont décrits en détail dans la documentation correspondant à ces exemples.

Solution :

La position maître devant être envoyée via le bus CAN2, le bus CAN2 de la MOVI-PLC® (X32) est raccordé au bus CAN2 des axes MOVIAXIS® (X12) et un télégramme de synchronisation cyclique est configuré via "MC_SetSync_CAN".



63350axx

Les réglages correspondants sont effectués dans la partie initialisation puis envoyés par les blocs fonction concernés aux entraînements dans la partie cyclique du programme.

Pour configurer au niveau de l'axe moteur un objet d'envoi cyclique avec la position souhaitée et au niveau de l'esclave l'identifiant correspondant comme objet de réception, la configuration de données-process des deux entraînements est adaptée via le bloc fonction "MC_InitConfig_MX".



IF NOT blnit THEN

```
(*#####*)
  Initialisation of AxisControlVirtual
  (*#####*)
```

(* Configuration virtual encoder *)

gVirtualEncoderConfiguration.StartRisingEdge := FALSE;

gVirtualEncoderConfiguration.LinkTecAxisVirtual.CanNode := SBUS_NODE_1; (* SBUS_NODE_1
SBUS_NODE_2 *)

gVirtualEncoderConfiguration.LinkTecAxisVirtual.SendObjectID := MDX_VIRTUAL_ENCODER_ID1;
(* name of the message which is used to transmit
MDX_VIRTUAL_ENCODER_ID1
MDX_VIRTUAL_ENCODER_ID2 *)

```
(*#####*)
  Initialisation of AxisControl axis 1 (single axis)
  (*#####*)
```

(* general axis control parameters *)

gAxisConfiguration_MX[1].StartRisingEdge := FALSE;

(* configuration modulo parameter --> only necessary if modulo mode is used *)

gAxisConfiguration_MX[1].SetModuloParameters.Overflow := 1;

gAxisConfiguration_MX[1].SetModuloParameters.Underflow := 1;

(* Extended Configuration by use of InitialConfig for send the encoder 1 position via CAN2 with ID "MX_PDO_ID1" *)
gAxisConfiguration_MX[1].bUseInitialConfig := TRUE;
gAxisConfiguration_MX[1].InitialConfig.ExtendedConfiguration_MX.SendSource := MX_SEND_ENCODER_1_SYS;
gAxisConfiguration_MX[1].InitialConfig.ExtendedConfiguration_MX.SendID := MX_PDO_ID1;
gAxisConfiguration_MX[1].InitialConfig.ExtendedConfiguration_MX.SyncSource := MX_SYNC_SOURCE_CAN2;

```
(*#####*)
  Initialisation of AxisControl axis 3 (gear motion)
  (*#####*)
```

(* general axis control parameters *)

gAxisConfiguration_MX[3].StartRisingEdge := FALSE;

(* configuration modulo parameter --> only necessary if modulo mode is used *)

gAxisConfiguration_MX[3].SetModuloParameters.Overflow := 1;

gAxisConfiguration_MX[3].SetModuloParameters.Underflow := 1;

(* Extended Configuration by use of InitialConfig for receive data with ID "MX_PDO_ID1" via CAN2 *)
gAxisConfiguration_MX[3].bUseInitialConfig := TRUE;
gAxisConfiguration_MX[3].InitialConfig.ExtendedConfiguration_MX.ReceiveID := MX_PDO_ID1;
gAxisConfiguration_MX[3].InitialConfig.ExtendedConfiguration_MX.SyncSource := MX_SYNC_SOURCE_CAN2;

(* Configuration gear parameters *)

gAxisConfiguration_MX[3].SetGearConfig.GearNumerator := 1; (* Numerator for scaling of the master increments *)

gAxisConfiguration_MX[3].SetGearConfig.GearDenominator := 1; (* Denominator for scaling of the master increments *)

gAxisConfiguration_MX[3].SetGearConfig.ModuloMax := 0; (* Modulo maximum Value of the master position *)

gAxisConfiguration_MX[3].SetGearConfig.ModuloMin := 0; (* Modulo minimum Value of the master position *)

gAxisConfiguration_MX[3].SetGearConfig.MasterSource := MX_GEAR_RECEIVE_PDO_1;

gAxisConfiguration_MX[3].SetGearConfig.LagErrorWindow := 10000; (* lag error window in [inc] *)

gAxisConfiguration_MX[3].SetGearConfig.RotationDirectionLock := 0; (* 0 = both directions, 1 = only CW enabled, 2 = only CCW enabled *)

gAxisConfiguration_MX[3].PrepareGearIn.GearSpeed := 1000; (* gear in speed in [rpm] *)

gAxisConfiguration_MX[3].PrepareGearIn.GearAcceleration := 10000; (* gear in acceleration in [rpm/s] *)

gAxisConfiguration_MX[3].PrepareGearIn.GearDeceleration := 10000; (* gear in deceleration in [rpm/s] *)

gAxisConfiguration_MX[3].SetGearConfig.ExtConfig.MSignal_InterpolationTime := 10; (* Interpolation time: range 1...60 means 0.5...30ms filter time *)

gAxisConfiguration_MX[3].SetGearConfig.ExtConfig.AverageFilterTime := 10; (* AverageFilter: range 1...60 means 0.5...30ms filter time *)

blnit:=TRUE;

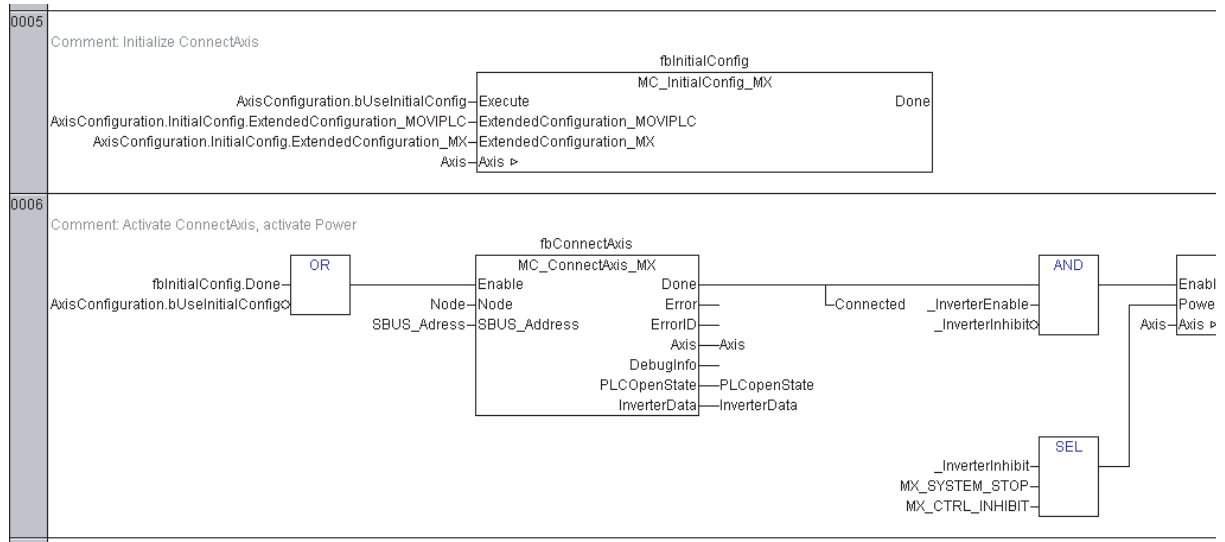
END_IF;

63351axx

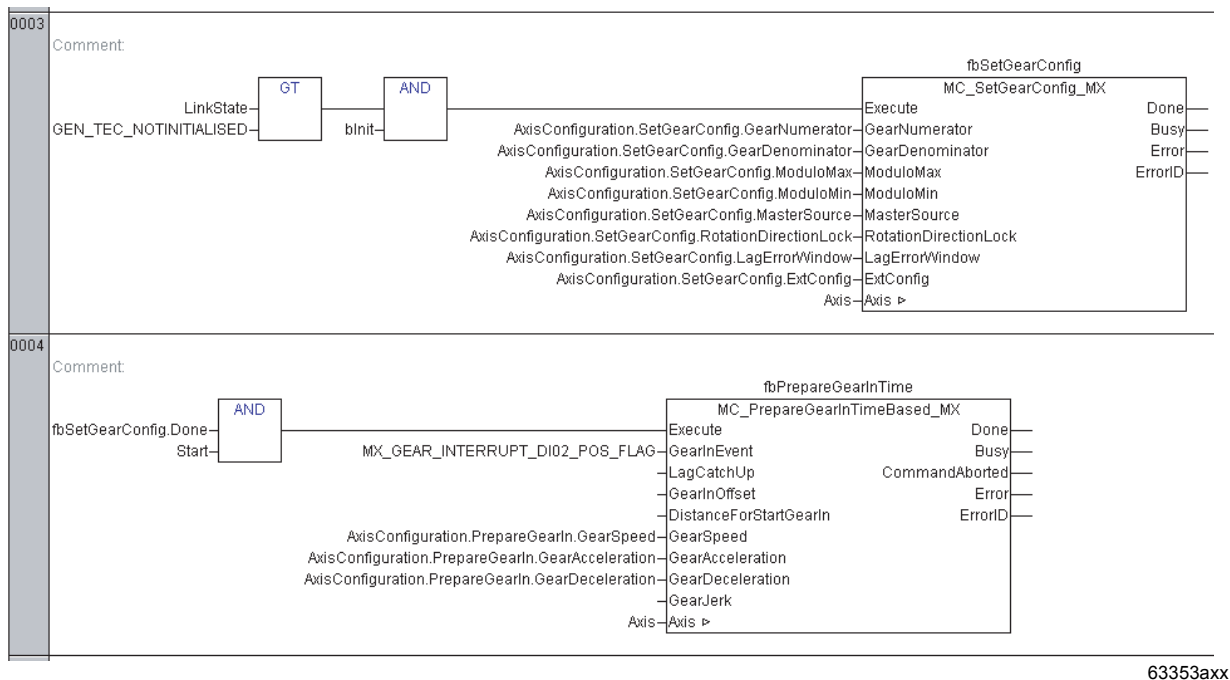
Partie initialisation pour la configuration des objets d'envoi / de réception du maître et de l'esclave



Le bloc fonction *InitialConfig* doit être appelé par le bloc fonction *ConnectAxis* ; la configuration des données process est alors activée à l'initialisation de *ConnectAxis*.



Dans le bloc "AxisModeGear", le *GearInEvent* de "MC_PrepareGearInTimeBased_MX" est réglé en conséquence et envoyé si *Start* = True. La synchronisation gérée par interruption est ainsi préparée. Si l'événement se produit, l'esclave se met en synchronisation ; en cas de front descendant sur *Start*, un arrêt à la limite application est également déclenché.





2.6.3 3. Gestion d'offset en distance, activée selon une trajectoire maître définie, répétition automatique

Dans cet exemple, l'exemple 2 est enrichi d'une gestion d'offset en distance.

Il doit être possible de définir respectivement comme entrées sur AxisControl la distance maître-esclave pour l'offset et la trajectoire maître après le début du déplacement d'offset. Les valeurs sont transmises par activation de l'entrée *SetOffsetData*.

Solution :

Dans un premier temps, le bloc fonction "AxisModeGear_MX" est élargi en conséquence :

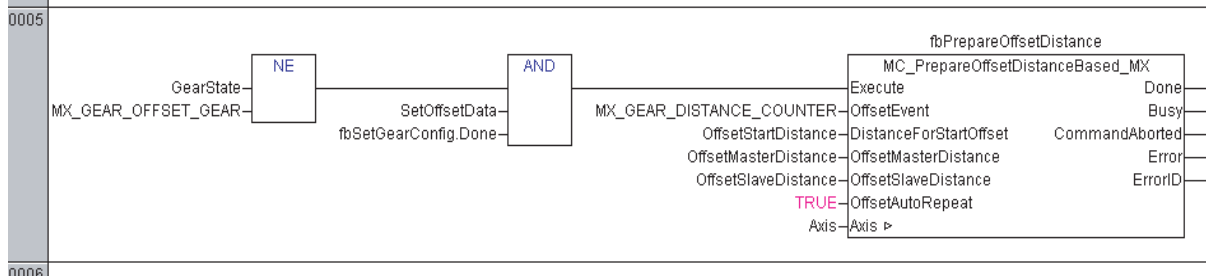
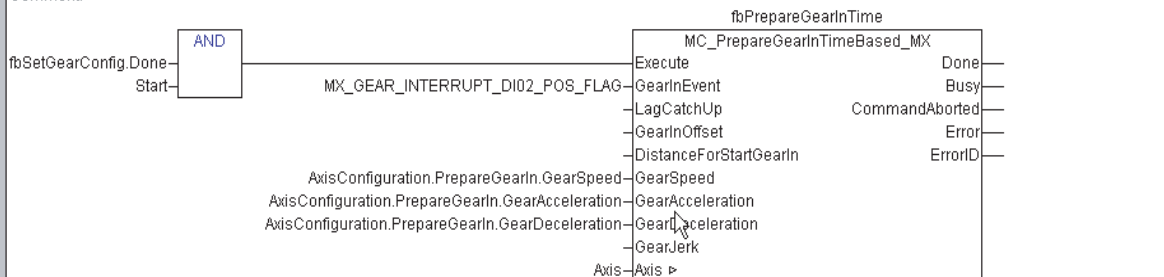
insertion de *PrepareOffsetDistance* avec la configuration adéquate de l'événement, les consignes pour l'offset distance maître/esclave et *DistanceForStartOffset*, ainsi que *SetOffsetData* sont utilisés comme entrées d'AxisModeGear. Le bloc fonction "PrepareOffset" ne doit pas être exécuté durant le déplacement d'offset, c'est pourquoi il reste verrouillé.

```

0013 FUNCTION_BLOCK AxisModeGear_MX
0014
0015 VAR
0016   fbSetGearConfig: MC_SetGearConfig_MX;
0017   fbGearInDirect: MC_GearInDirect_MX;
0018   bInit: BOOL;
0019   fbStop: MC_Stop_MX;
0020   fbFallingTrigStart: F_TRIG;
0021   fbPrepareGearInTime: MC_PrepareGearInTimeBased_MX;
0022   fbPrepareOffsetDistance: MC_PrepareOffsetDistanceBased_MX;
0023 END_VAR
0024 VAR_INPUT
0025   Enable: BOOL;
0026   Start: BOOL;
0027   AxisConfiguration: ST_AxisConfiguration_MX;
0028   SetOffsetData: BOOL;
0029   OffsetMasterDistance: DINT;
0030   OffsetSlaveDistance: DINT;
0031   OffsetStartDistance: DINT;
0032 END_VAR

```

Comment:

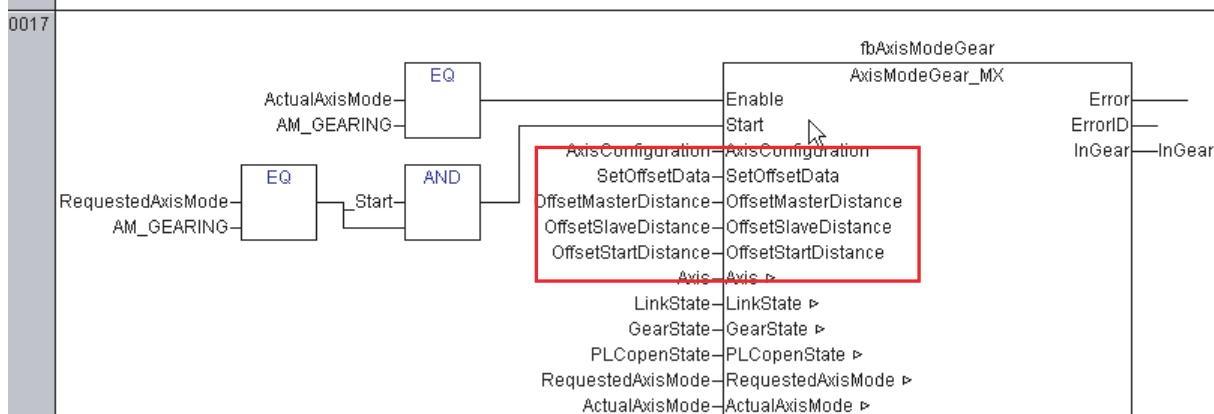


63354axx

```

0064
0065 END_VAR
0066 VAR_INPUT
0067     HControl: BOOL;
0068     Node: UINT;
0069     SBUS_Address: UINT;
0070     InverterInhibit: BOOL;
0071     InverterEnable: BOOL;
0072     Reset: BOOL;
0073     AxisMode: ENUM_AXISMODE;
0074     Start: BOOL;
0075     JogPos: BOOL;
0076     JogNeg: BOOL;
0077     Position: DINT;
0078     Velocity: DINT;
0079     Acceleration: DINT;
0080     Deceleration: DINT;
0081     ModuleMode: MC_MODULEMODE_MX;
0082     AxisConfiguration : ST_AxisConfiguration_MX;
0083     SetOffsetData: BOOL;
0084     OffsetStartDistance: DINT;
0085     OffsetMasterDistance: DINT;
0086     OffsetSlaveDistance: DINT;
0087 END_VAR
0088 VAR_OUTPUT

```



Manuel Bibliothèques MPLCTec.. MX



Afin que la configuration de l'offset puisse également être gérée via l'interface d'entrée, cette possibilité de réglage est également ajoutée à la structure "ST_AxisInterfaceInType_MX".

TYPE ST_AxisInterfaceInType_MX :

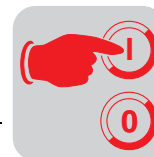
STRUCT

HControl: BOOL ;	(* FALSE: fbAxisControl is controlled by the input variables TRUE: fbAxisCor
InverterInhibit: BOOL ;	(* TRUE: inverter is in state inverter inhibit *)
InverterEnable: BOOL ;	(* TRUE: inverter is enabled (if no inverter inhibit is set, no digital input is pr
Reset: BOOL ;	(* rising edge resets and inverter fault *)
AxisMode: ENUM_AXISMODE ;	(* choose the requested axis mode
	AM_DEFAULT, (* 0 *)
	AM_VELOCITY, (* 1 *)
	AM_POSITIONING, (* 2 *)
	AM_POSITIONINGMODULO, (* 3 *)
	AM_POSITIONINGRELATIVE, (* 4 *)
	AM_JOG_5, (* 5 *)
	AM_JOG_A, (* 6 *)
	AM_HOMING, (* 7 *)
	AM_CAMING, (* 8 *)
	AM_GEARING, (* 9 *)
	AM_POSITIONINGMODULO_RELATIVE, (* 10 *)
	AM_USER_SELECTION (* 11 *) *)
Start: BOOL ;	(* starts a movement in the choosen axis mode (except jogging modes *)
JogPos: BOOL ;	(* jog axis in positiv direction *)
JogNeg: BOOL ;	(* jog axis in negativ direction *)
Position: DINT ;	(* target position for positioning axis modes (4096 incr. / motor rotation)*)
Velocity: DINT ;	(* velocity for all movements (rpm) *)
Acceleration: DINT ;	(* ramp up time (rpm/s) for all movements *)
Deceleration: DINT ;	(* ramp down time (rpm/s) for all movements *)
ModuloMode: MC_MODULOMODE_MX ;	(* modulo mode for the axis mode modulo
	MX_SHORT, Shortest way to target
	MX_CW, Clockwise direction to target
	MX_CCW, Counterclockwise direction to target*)
SetOffsetData: BOOL ;	(* Sets the selected offset data *)
OffsetStartDistance: DINT ;	(* MasterDistance the Offset will be started *)
OffsetMasterDistance: DINT ;	(* Master distance within the offset will be executed *)
OffsetSlaveDistance: DINT ;	(* additional Slave distance during the offset move*)

END_STRUCT

END_TYPE

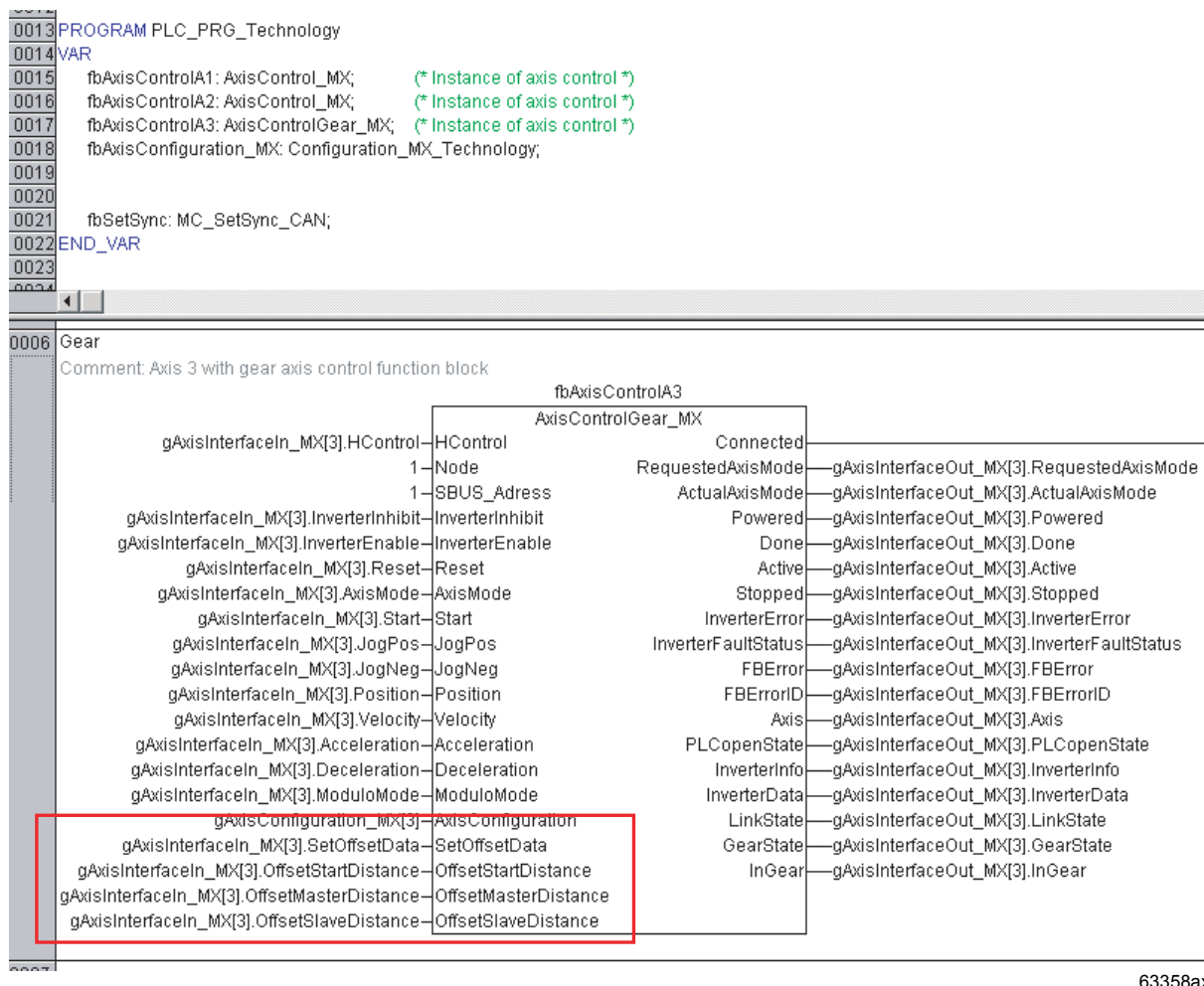
63556axx



Les entrées supplémentaires de "AxisControlGear" peuvent à présent être affectées avec la structure d'entrée élargie.

Remarque :

Dans cet exemple, nous avons renoncé à reproduire également les entrées supplémentaires via les variables système, c'est pourquoi il n'y a pour cette fonction élargie pas d'interface vers l'"ApplicationBuilder" ou vers un pupitre opérateur DOP.





2.7 Annexes

2.7.1 Détection des défauts (ErrorID)

Code défaut	Nom du défaut	Description du défaut
FA0010	E_IEC_GENERAL_INVALID_TECHNOLOGIE_OPTION	La fonction technologique demandée n'est pas supportée
FA020...	General Link Tec Error	
FA0200	E_TEC_GENERAL_MULTIPLE_TECLINKS	Les appels multiples du bloc fonction LinkTec ne sont pas autorisés
FA0201	E_TEC_GENERAL_INVALID_LINKSTATE	LinkState non admissible pour appel de fonction
FA0202	E_TEC_GENERAL_NOT_LINKED	Le bloc fonction LinkTec n'a pas encore de liaison logique.
FA0203	E_TEC_GENERAL_NOT_INITIALIZED	Fonction technologique pas encore activée
FA0204 ¹⁾	E_TEC_GENERAL_SERVICE_NOT_IMPLEMENTED	La fonction demandée n'est pas supportée
FA024	Gear Technologyfunction Error	
FA0240 ¹⁾	E_TEC_GEAR_CLEAR_LAG_FAILED	La remise à zéro du compteur d'écart avec MC_GearClearLag_MDX n'a pas fonctionné
FA0241	E_TEC_GEAR_INVALID_GEAR_MODE	Appel de la fonction non autorisé avec GearState actuel
	Messages de défaut généraux CEI	
FA0070	E_IEC_PARAMETER_VALUE_OUT_OF_RANGE	Valeur de paramètre saisie non autorisée
FB0071	E_MDX_MOTIONBLOCK_LOG_ADR_NOT_INITIALIZED	MC_ConnectAxis_MDX n'a pas encore attribué d'adresse log
FB0072	E_MDX_MOTIONBLOCK_INVALID_LOG_ADR	Adresse log non valide
FB0073	E_MDX_MOTIONBLOCK_INVALID_STATE	Appel de la fonction depuis PLCopenState actuel non autorisé
FB0074 ¹⁾	E_MDX_MOTIONBLOCK_INVALID_OPERATING_MODE	La fonction technologique demandée n'est pas supportée par le mode d'exploitation MOVIDRIVE® réglé.

1) Ce message de défaut n'est pas pertinent avec un MOVIAXIS® (uniquement avec un MOVIDRIVE®)

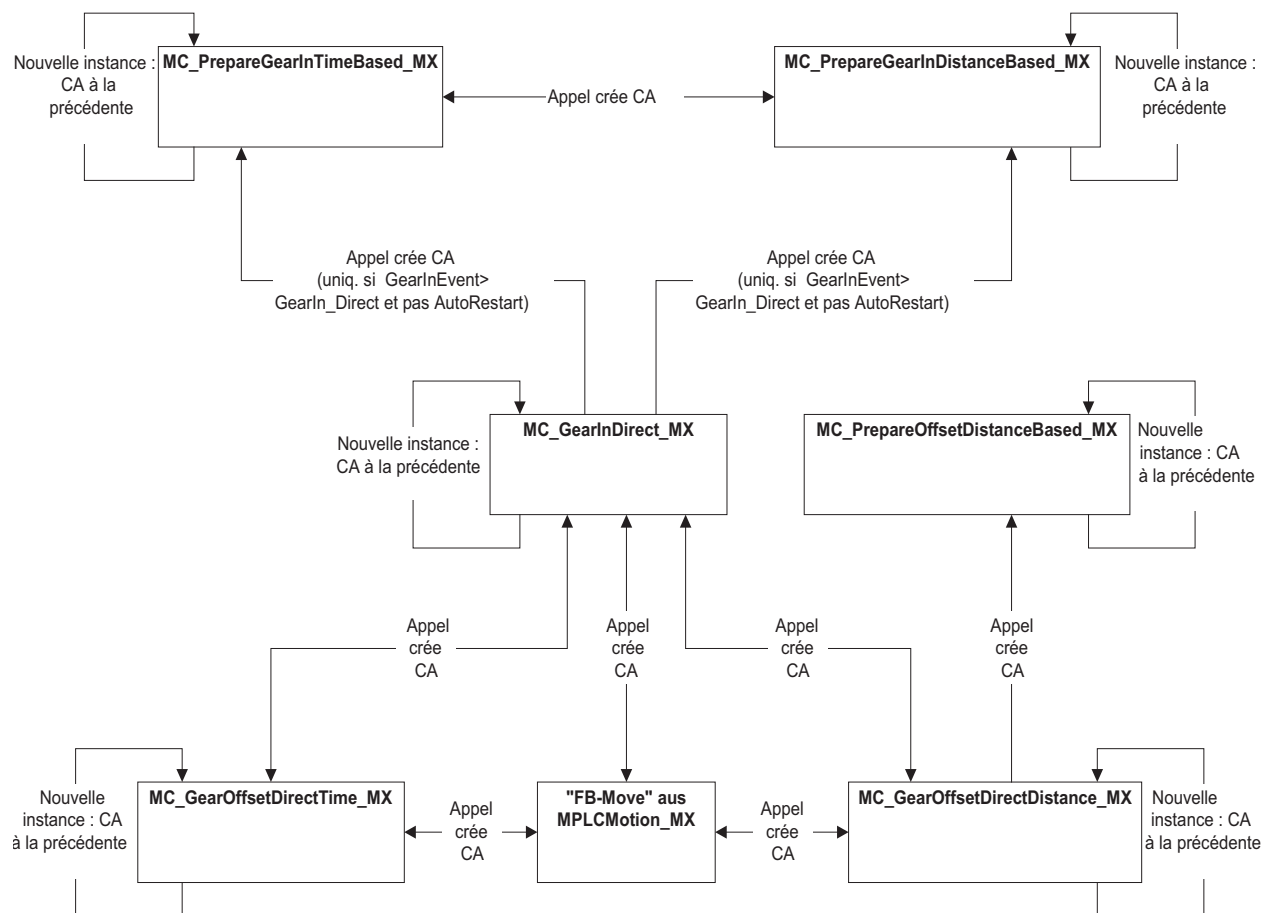
2.7.2 Interruption des tâches des blocs fonction (CommandAborted)

La sortie "CommandAborted" est utilisée pour signaler les appels multiples d'un bloc fonction de déplacement ou l'appel d'un autre bloc fonction de déplacement. La tâche du bloc fonction interrompu ne sera pas menée à terme (pour des informations détaillées à ce sujet, voir le manuel "Bibliothèque MPLCMotion_MDX / MX pour MOVI-PLC®").

Bloc fonction activé	provoque CommandAborted sur :
MC_GearInDirect_MX	- Instances de MC_GearInDirect_MX - MC_PrepareGearIn..... (si Event<>Direct et pas AutoRestart) - "Move-FB"
MC_GearOffsetDirectTime_MX	- Instances de MC_GearOffsetDirectTime_MX - MC_GearInDirect_MX - "Move-FB"
MC_GearOffsetDirectDistance_MX	- Instances de MC_GearOffsetDirectDistance_MX - MC_PrepareOffsetDistanceBased_MX - MC_GearInDirect_MX - "Move-FB"
MC_PrepareGearInTimeBased_MX	- Instances de MC_PrepareGearIn.....
MC_PrepareGearInDistanceBased_MX	- Instances de MC_PrepareGearIn.....
MC_PrepareOffsetDistanceBased_MX	- Instances de MC_PrepareOffsetDistanceBased_MX
"Move-FB"	- MC_GearInDirect_MX - MC_GearOffsetDirectTime_MX - MC_GearOffsetDirectDistance_MX



Schéma de "CommandAborted" :



63359axx

Remarque :

Les blocs fonction "Move-FB" de la bibliothèque MPLCMotion_MX sont les suivants :

- MC_MoveAbsolute_MX
- MC_MoveRelative_MX
- MC_MoveTargetPosition_MX
- MC_MoveTargetSpeed_MX
- etc.



3 MPLCTecCamMotion_MX

3.1 Introduction

Les servovariateurs multi-axes MOVIAXIS® offrent des fonctions technologiques très complètes, telles que :

- Réducteur électronique
- Cames électroniques
- Capteurs d'événements
- Gestion d'événements
- Boîte à cames

L'assistant de mise en route "Motion Technologie-Editor" de MotionStudio permet de configurer toutes les fonctions technologiques. Il est alors possible de procéder à un réglage de base. Mais il est fréquent que certaines valeurs ou réglages nécessitent une modification en cours de process, ou alors diverses fonctions devront être combinées et se dérouler de manière coordonnée dans le temps.

Les blocs fonction de la bibliothèque "MPLCTecCamMotion_MX" règlent les paramètres Cam généraux et configurent également d'autres fonctions technologiques nécessaires pour la fonction de Cam souhaitée.

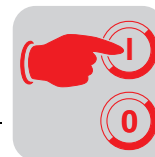
Une initiation approfondie aux paramètres respectifs des fonctions technologiques n'est ainsi pas nécessaire. La configuration avec l'éditeur technologique n'est pas nécessaire, tous les réglages nécessaires étant effectués par les blocs fonction de la bibliothèque "MPLCTecCamMotion_MX".

Les avantages de la bibliothèque "MPLCTecCamMotion_MX" sont :

- Paramétrage selon CEI61131, permettant l'implémentation aisée des applications spécifiques
- Bibliothèque pour le pilotage de la fonction technologique "Cam". Celle-ci permet la configuration et le pilotage aisés des fonctionnalités Cam sans nécessiter de connaissances approfondies des différentes fonctions technologiques.
- Pilotage centralisé de plusieurs axes de came électronique.
- Fonctionnalités complémentaires, comme p. ex. la définition de la valeur maître par un codeur maître virtuel, commandées de manière centralisée via la commande MOVI-PLC®
- Possibilité de sélectionner divers événements et types de synchronisation
- Possibilité d'activer une gestion d'offset en mode came électronique
- Changement entre plusieurs courbes facilité
- Possibilité de relier jusqu'à dix courbes maximum avec des passages de courbes
- Disponibilité de blocs fonction de calcul de points de came pour déplacements Rest to Rest
- Possibilité d'adapter les profils de courbe à l'aide de blocs fonction de mise à l'échelle
- L'éditeur Cam permet de définir 10 courbes. Possibilité de transférer les courbes du PC au variateur et du variateur au PC séparément

D'autres informations figurent dans les documentations suivantes :

- Les remarques générales concernant le fonctionnement de la bibliothèque figurent dans le document "Bibliothèques MPLCMotion_MDX et MPLCMotion_MX pour MOVI-PLC®".
- Manuel "Servovariateurs multi-axes MOVIAXIS® MX"



3.2 Domaines d'application

La bibliothèque "MPLCTecCamMotion_MX" convient à toutes les applications nécessitant des déplacements d'axes synchronisés.

Exemples d'application

- Plusieurs axes synchronisés avec un maître virtuel.
- Nombreuses possibilités d'applications dans l'emballage.
 - Ensacheuses, fermetures d'emballages et mesureurs
 - Formeuses, poinçonnage et translation de cartons synchronisés par codeur virtuel
 - Cartonnage, confection d'attaches et transport de carton synchronisés
 - Alignement de produits par banderolage
 - Rotation, positionnement et translation synchronisées
- Machines à trancher / à coller les livres
- Coupe à la volée, par exemple pour la découpe d'un matériau continu
- Transport de matériel synchronisé (p. ex. synchronisation de plusieurs bandes transporteuses sur un codeur maître)
- Dispositifs de translation et de levage multi-axes
- Dispositifs de remplissage dans l'industrie des boissons
- Tous les déplacements synchronisés avec référence maître-esclave définie.

Propriétés

- Possibilité d'exploitation de maximum 64 axes moteur (avec MOVI-PLC[®] *advanced*) en mode came électronique
- Nombreuses possibilités de configuration, permettant une programmation claire à l'aide des blocs fonctions correspondants
- Possibilité de régler différentes sources de codeur maître
- Choix entre divers modes de synchronisation
- Choix entre divers types d'offset
- Lancement simplifié de l'offset à la mise en synchronisation ou en mode synchronisé
- Possibilité de réaliser diverses combinaisons maître / esclave et de configurer plusieurs maîtres de came électronique



3.3 Configuration

3.3.1 Conditions

- Pour pouvoir utiliser la bibliothèque `MPLCTecCamMotion`, une commande `MOVI-PLC®` en exécution technologique T1 ou supérieure est nécessaire.
- La bibliothèque `MPLCMotion_MX` doit être intégrée. Les conditions préalables indiquées dans le manuel "Bibliothèques `MPLCMotion_MDX` et `MPLCMotion_MX` pour `MOVI-PLC®`" sont également applicables.

3.3.2 Principe de base et remarques

- Réglage des fonctions technologiques adéquates de l'axe via blocs fonction CEI.
- Mise en synchronisation et resynchronisation via la fonction technologique "Cam", puis basculement automatique sur la fonction Cam linéaire. Grâce à cela, la gestion d'offset est possible durant le fonctionnement synchronisé.
- Synchronisation rapide gérée par interruption grâce à l'utilisation de la fonction capteur d'événements et de la gestion d'événements (la configuration correspondante s'effectue automatiquement).
- Le bloc fonction "MC_LinkTecCam_MX" est indispensable comme interface de données centrale pour la fonction technologique "Cam".
- L'éditeur technologique de MotionStudio n'est pas nécessaire, mais il est possible pour le diagnostic d'établir une liaison online du moniteur avec l'exemple de projet "MOVI-PLC® Defaultprojekt.....TecFunction". Cependant l'exemple de projet ne doit pas être chargé dans le variateur, sans quoi les réglages effectués par la bibliothèque seraient à nouveau écrasés.
- L'éditeur Cam permet de définir et de charger dans le `MOVIAXIS®` jusqu'à dix profils de courbe. Lors du transfert vers le variateur, la courbe est sauvegardée avec 512 points de came. Lors du chargement dans le PC, un tableau avec 512 points de came est généré. Afin d'obtenir une meilleure résolution, les points de came sont mis à une échelle plus grande via le facteur de décalage. Le facteur Shift est déterminé automatiquement pour toutes les courbes. Il est également possible de définir le facteur de décalage maximal. En cas de nouveau facteur suite à une nouvelle définition de courbe, il n'est pas possible de charger une courbe seule, toutes les courbes devront alors être chargées.



3.3.3 Synchronisation directe

Le bloc fonction "MC_CamInDirekt_MX" permet de passer directement en mode came électronique. Le déplacement s'effectue selon la courbe sélectionnée. En cas de verrouillage (arrêt d'urgence), l'axe n'est plus calé. Pour recalibrer l'axe après un verrouillage, il est possible d'activer la fonction "Move-to-Master" à l'aide du bloc fonction "MC_PrepareCamIn_MX".

Le bloc fonction "MC_CamTableSelect_MX" permet de passer d'une courbe à l'autre ; la commutation s'effectue lorsque la fin de la courbe est atteinte.

3.3.4 Calage de l'axe et synchronisation

Le bloc fonction "MC_CamInAdjust_MX" permet dans un premier temps de caler l'axe (FCB 9), puis de passer en mode came électronique. Le calage de l'axe peut être défini via l'entrée *Adjustmode*. Le déplacement s'effectue selon la courbe sélectionnée. Après un verrouillage (arrêt d'urgence), l'axe est recalé par la trajectoire la plus courte.

Le bloc fonction "MC_CamTableSelect_MX" permet de passer d'une courbe à l'autre ; la commutation s'effectue lorsque la fin de la courbe est atteinte.

3.3.5 Synchronisation avec interruption

Le bloc fonction "MC_PrepareCamIn_MX" permet de définir l'événement "CamInEvent". Il est également possible d'activer la fonction *MovetoMaster* pour la mise en synchronisation.

Événements de synchronisation possibles :

Toutes les entrées du MOVIAXIS® ainsi que les signaux de la voie C des codeurs 1 à 3 peuvent être paramétrés comme événements d'interruption.

La synchronisation est lancée par un événement d'interruption déclenché par une modification du signal par exemple sur l'entrée binaire DI02 du MOVIAXIS®, par un front montant / descendant (Pos / Neg Flag) ou par une modification du signal sur DI02.

- MX_CAM_DIRECT

Le bloc fonction "MC_CamInDirect_MX" permet de lancer directement la synchronisation.

- MX_CAM_INTERRUPT_DI01_POS_FLAG

- MX_CAM_INTERRUPT_DI01_NEG_FLAG

- MX_CAM_INTERRUPT_DI01

...

- MX_CAM_INTERRUPT_DI08_POS_FLAG

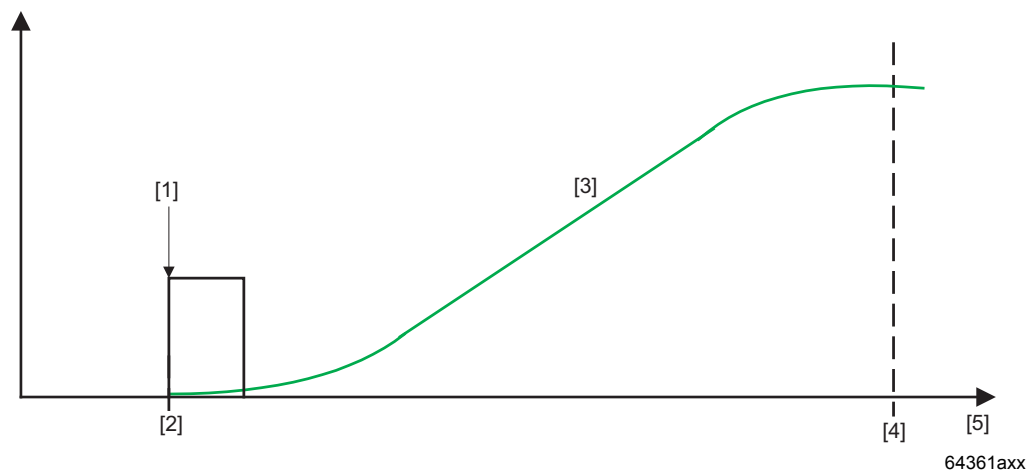
- MX_CAM_INTERRUPT_DI08_NEG_FLAG

- MX_CAM_INTERRUPT_DI08

- MX_CAM_INTERRUPT_C_TRACK_ENC1

- MX_CAM_INTERRUPT_C_TRACK_ENC2

- MX_CAM_INTERRUPT_C_TRACK_ENC3



- [1] Interruption DI02 (front montant)
- [2] Démarrage de la synchronisation
- [3] n esclave
- [4] L'esclave est synchronisé
- [5] Position maître

3.3.6 Gestion de l'offset

Lorsque la synchronisation est achevée, il est possible de lancer une gestion d'offset au niveau de l'esclave en cours de fonctionnement synchronisé. Le bloc fonction "MC_OffsetDirekt_MX" permet de lancer la gestion d'offset par rattrapage en distance de la trajectoire maître. Par conséquent l'offset résiduel de l'esclave par rapport à la courbe actuelle est additionné. Ceci permet de compenser la référence maître-esclave.

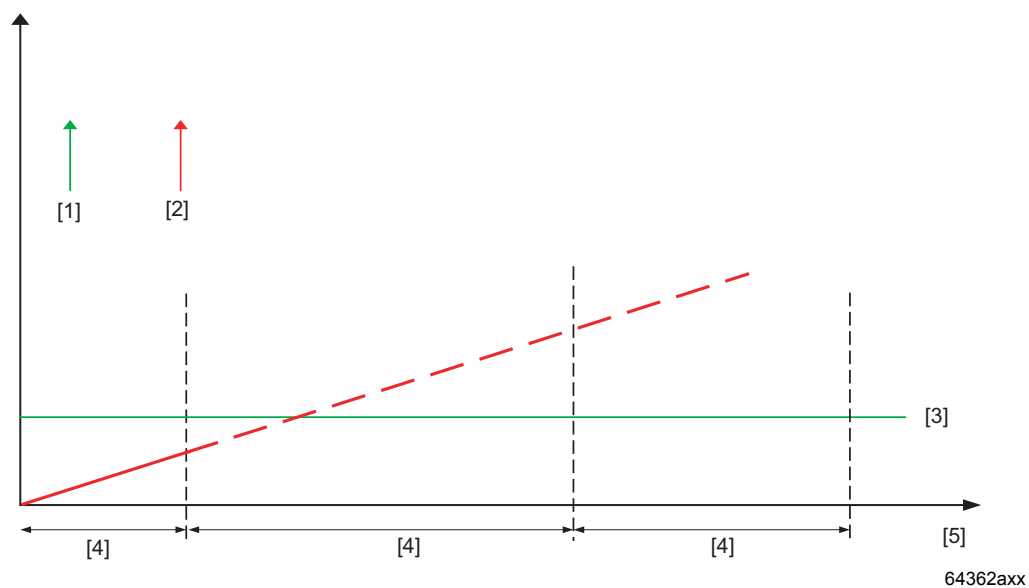
Possibilité de renseigner un offset négatif ou positif pour l'esclave.

L'offset est calculé avec une transition de type polynôme_5.

Exemple : pour améliorer la compréhension, la gestion d'offset est représentée par deux courbes de synchronisation.



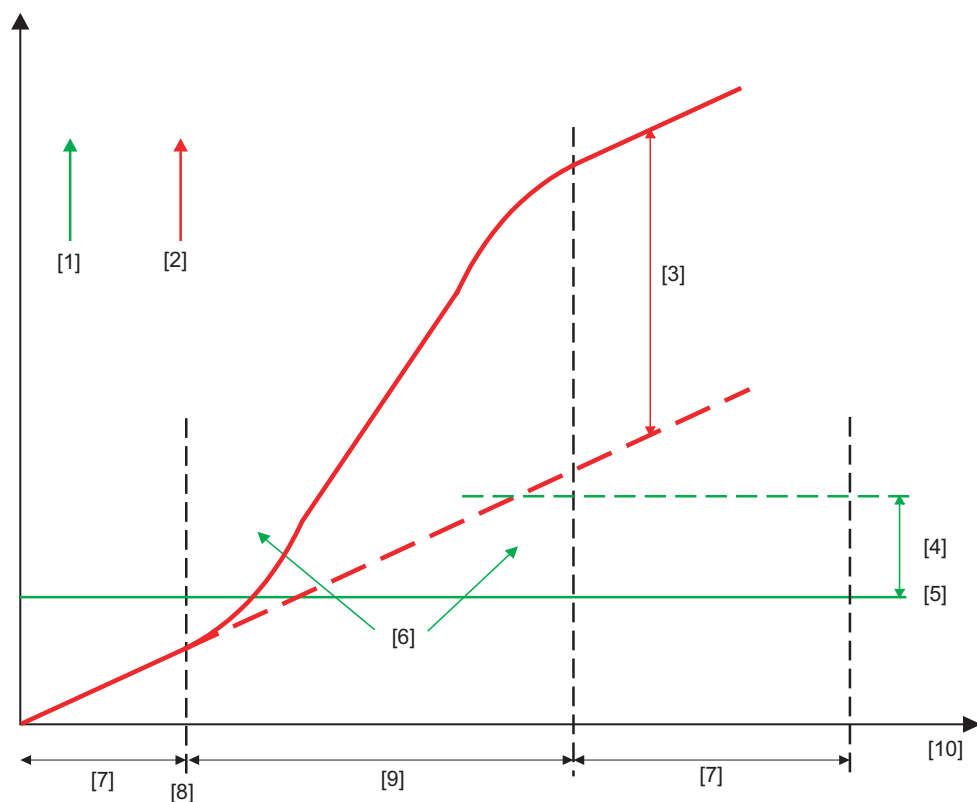
Profil de courbe sans offset



- [1] Vitesse esclave
- [2] Position esclave
- [3] n esclave
- [4] Synchrone
- [5] Position maître

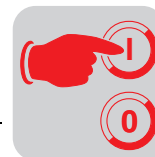


Profil de courbe avec offset :



64363axx

- [1] Vitesse esclave
- [2] Position esclave
- [3] Offset esclave
- [4] Vitesse offset
- [5] n esclave
- [6] Offset accélération / décélération
- [7] Synchrone
- [8] Offset de démarrage
- [9] Cycle maître de la gestion d'offset
- [10] Position maître



3.3.7 Logique de détermination

Procédure

- La bibliothèque MPLCTecCamMotion_MX peut être insérée dans le projet avec le gestionnaire des bibliothèques en complément à la bibliothèque MPLCMotion_MX.
- Le bloc fonction "MC_LinkTecCam_MX" est l'interface de la fonction technologique de l'axe et doit donc être appelé de manière cyclique dans le programme. Lorsque le bloc fonction "MC_ConnectAxis_MX" a établi la connexion avec l'axe (Done = TRUE), le bloc fonction "LinkTecCam" active les fonctions technologiques du MOVIAXIS® nécessaires pour la fonction Cam. Les réglages nécessaires sont également effectués dans l'éditeur PDO. Un objet de réception destiné à réceptionner la position maître via le bus système est notamment configuré dans le tampon In 5.
- Si la position maître doit être transmise via le bus système, il est recommandé de synchroniser le bus système. Le télégramme de synchronisation du SBus nécessaire se configure à l'aide du bloc fonction "MC_SetSync_MX", qui fait partie de la bibliothèque MPLCMotion_MX). Si le codeur virtuel est utilisé comme maître de la commande, le SBus est synchronisé automatiquement à l'appel du bloc fonction "MC_LinkTecAxis_Virtual", qui fait partie de la bibliothèque MPLCTecVirtualEncoder. Les réglages de base nécessaires pour cela se trouvent dans l'arborescence paramètres sous Communication. L'initialisation du bloc "LinkTecCam" est effectuée automatiquement.
- Le bloc fonction "MC_SetCamConfig_MX" permet dans un premier temps de régler les principaux paramètres. Ce bloc fonction peut être ouvert par un front montant sur l'entrée Execute à chaque nouveau démarrage de la commande MOVI-PLC® ou lorsqu'on souhaite modifier les paramètres généraux.
- Le bloc fonction "MC_CamInDirect_MX" permet de lancer directement la synchronisation. Après un arrêt d'urgence, l'axe devra être calé par l'application. Si l'axe doit automatiquement à nouveau être déplacé vers le point de courbe après un arrêt d'urgence, utiliser la fonction *Move_to_Master* (Prepare). Si la synchronisation s'effectue via "MC_CamInAdjust_MX", l'axe sera d'abord calé (FCB 9) en fonction du mode d'ajustement réglé. Le calage après un arrêt d'urgence ou un verrouillage est effectué à la mise sous tension via le générateur de profil. Ceci s'effectue toujours par le chemin le plus court DX / DV.



3.3.8 Stockage des données

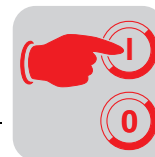
Stockage des courbes

Le nombre maximal de courbes est de 10.

Le nombre de points par courbe est de 512.

La bibliothèque configure une courbe GEAR et une courbe WAIT.

Offset	Index / Sous-index	Description
0	20000.1	Début courbe GEAR
..		
4	20000.5	Fin courbe GEAR
5	20000.6	Début courbe WAIT
...		
9	20000.10	Fin courbe WAIT
10	20000.11	Début courbe 1
...		
521	20004.10	Fin courbe 1
522	20004.11	Début courbe 2
...		
1033	20008.10	Fin courbe 2
1034	20008.11	Début courbe 3
...		
1545	20012.10	Fin courbe 3
1546	20012.11	Début courbe 4
...		
2057	20016.10	Fin courbe 4
2058	20016.11	Début courbe 5
...		
2569	20020.10	Fin courbe 5
2570	20020.11	Début courbe 6
...		
3081	20024.10	Fin courbe 6
3082	20024.11	Début courbe 7
...		
3593	20028.10	Fin courbe 7
3594	20028.11	Début courbe 8
...		
4105	20032.10	Fin courbe 8
4106	20032.11	Début courbe 9
...		
4617	20036.10	Fin courbe 9
4618	20036.11	Début courbe 10
...		
5129	20040.10	Fin courbe 10

**Sauvegarde des caractéristiques des courbes**

Interface de données entre came électronique et bibliothèque CAM

Offset	Index / Sous-index	Description
4095	20231.128	Version et sous-version de l'éditeur
4094	20231.127	Drapeaux pour courbes chargées (un bit par courbe) Bit 0 = bit global
4093	20231.126	Nombre de courbes
4092	20231.125	Nombre de points de came (512)
4091	20231.124	Facteur de décalage esclave
4090	20231.123	Facteur de décalage maître
4089	20231.122	Longueur maître courbe 1
4088	20231.121	Longueur maître courbe 2
4087	20231.120	Longueur maître courbe 3
4086	20231.119	Longueur maître courbe 4
4085	20231.118	Longueur maître courbe 5
4084	20231.117	Longueur maître courbe 6
4083	20231.116	Longueur maître courbe 7
4082	20231.115	Longueur maître courbe 8
4081	20231.114	Longueur maître courbe 9
4080	20231.113	Longueur maître courbe 10

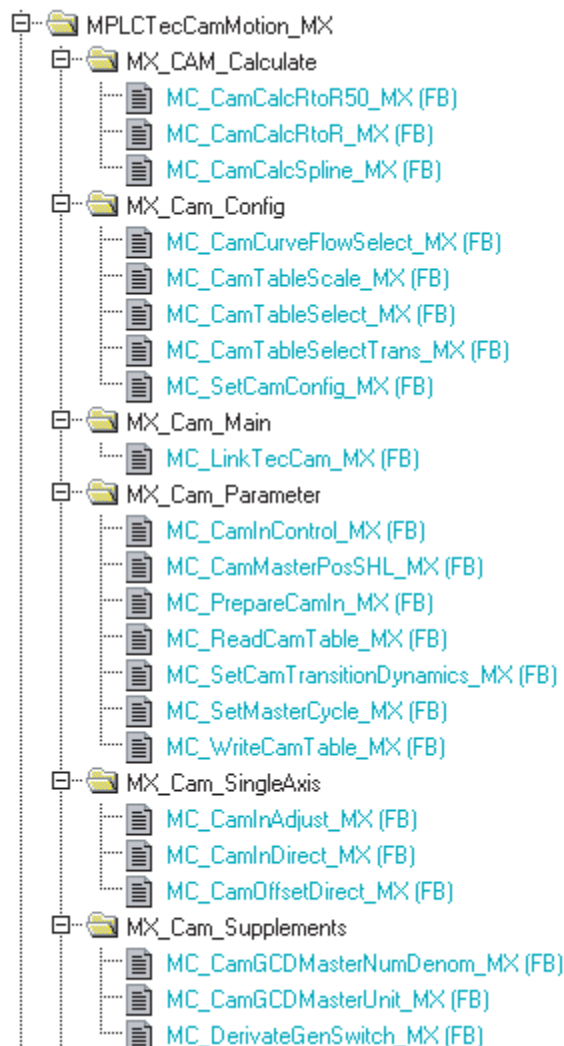
L'éditeur de came électronique modifie l'interface de données (sauf décalage maître), le bloc fonction "MC_LinkTecCam_MX" copie, en cas de modification par l'éditeur, les valeurs des points correspondants dans le DDB.

Le bloc fonction "MC_CamMasterPosSHL_MX" permet d'adapter le facteur de mise à l'échelle du maître. Les paramètres de la courbe sont modifiés automatiquement par le bloc fonction.



3.4 La bibliothèque MPLCTecCamMotion_MX

3.4.1 Bref aperçu de la bibliothèque MPLCTecCamMotion_MX



64364axx

MC_LinkTecCam_MX

Le bloc fonction "MC_LinkTecCam_MX" dispose de diverses fonctions :

- Configuration des fonctions technologiques du MOVIAXIS® et activation des fonctions nécessaires pour Cam.
- Configuration de l'interface de données-process MOVIAXIS®.
- Indication de l'état actuel de la fonction technologique "Cam"
- Indication de l'état d'initialisation

MC_SetCamConfig_MX

Ce bloc fonction permet d'effectuer la configuration de base de la synchronisation. Avant de lancer la synchronisation, les paramètres doivent avoir été transférés via ce bloc fonction.



MC_CamInDirect_MX

Ce bloc fonction permet de lancer la synchronisation directe par un front montant sur l'entrée Execute. Après un arrêt d'urgence, l'axe doit être orienté par l'application. Si, après un arrêt d'urgence, l'axe doit automatiquement à nouveau se déplacer vers un point de courbe, utiliser la fonction *Move_to_Master* (Prepare).

MC_CamInAdjust_MX

Ce bloc fonction permet de lancer la synchronisation directe par un front montant sur l'entrée Execute. L'axe est d'abord calé en fonction du mode d'ajustement réglé (FCB 9).

Après un arrêt d'urgence ou un verrouillage, le calage s'effectue à la mise sous tension avec le générateur de profil. Ceci s'effectue toujours par le chemin le plus court DX / DV.

MC_CamOffsetDirect_MX

Le bloc fonction "MC_CamOffsetDirect_MX" active, à l'aide des paramètres transmis, un positionnement additionné au profil de courbe. La trajectoire esclave est ajoutée à la trajectoire maître par rattrapage en distance.

MC_PrepareCamIn_MX

Le bloc fonction "MC_PrepareCamIn_MX" peut configurer un événement d'interruption pour la synchronisation. La fonction *Move_To_Master* peut être sélectionnée ou désélectionnée. La fonction d'interruption peut être libérée ou verrouillée via le bloc fonction "MC_CamInControl_MX".

MC_CamInControl_MX

Ce bloc fonction permet de libérer ou de verrouiller l'événement d'interruption. Le redémarrage automatique peut être activé.

MC_CamTableSelect_MX

Le bloc fonction "MC_CamTableSelect_MX" permet de sélectionner une courbe (0 - 10). Lorsqu'une courbe est activée, la commutation sur la courbe nouvellement sélectionnée ne s'effectue que lorsque la fin de la courbe est atteinte. La sortie de la courbe est liée à la fin de la courbe, la courbe est alors parcourue sans fin (courbe 0 = courbe d'attente).

MC_CamTableSelectTrans_MX

Le bloc fonction "MC_CamTableSelectTrans_MX" permet de sélectionner une courbe (1 - 10) avec une transition. Lorsqu'une courbe est activée, la commutation sur la courbe nouvellement sélectionnée ne s'effectue que lorsque la fin de la courbe est atteinte. La sortie de la courbe passe sur une transition. La fin de la transition est liée au début de la courbe, si bien que la courbe et la transition sont parcourus en continu.

MC_CamCurveFlowSelect_MX

Ce bloc fonction permet de lier entre elles jusqu'à 10 courbes.

Chaque sortie de courbe peut être liée avec une transition.

**MC_CamTableScale_MX**

Le bloc fonction "MC_CamTableScale" permet d'adapter les paramètres Numérateur, Dénominateur pour une courbe.

MC_SetMasterCycle_MX

Le bloc fonction "MC_SetMasterCycle_MX" permet d'adapter la longueur du cycle maître pour une courbe.

MC_CAM_MasterPosSHL_MX

Afin d'obtenir une meilleure résolution, il est possible de mettre à l'échelle le cycle maître en fonction du facteur de décalage du maître.

Exemple : cycle maître = 10000 => facteur décalage maître = 6, le cycle maître est alors de $10000 \times 2^6 = 64000$.

Ce bloc fonction permet de modifier la mise à l'échelle des cycles maître de toutes les courbes.

MC_SetCamTransitionDynamics_MX

Ce bloc fonction permet de modifier les paramètres de dynamisme des courbes de transition de la courbe sélectionnée.

MC_ReadCamTable_MX

Ce bloc fonction permet de lire les 512 points de came de la courbe sélectionnée depuis le MOVIAXIS® et de les écrire dans un tableau.

MC_WriteCamTable_MX

Ce bloc fonction permet d'écrire pour la courbe sélectionnée un Array (tableau) avec 512 points de came dans le MOVIAXIS®.

MC_CamGCDMasterNumDenom_MX

Ce bloc fonction recherche le plus grand dénominateur commun et raccourcit les interruptions en conséquence. Aucun paramètre n'est modifié.

MC_CamGCDMasterUnit_MX

Ce bloc fonction calcule, à partir de la distance maître, un numérateur et un dénominateur. Aucun paramètre n'est modifié.

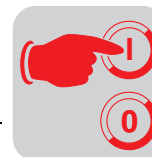
MC_Derivate_Gen_Switch_MX

Cette bibliothèque fonctionne avec CamFlow1 et CamFlow2 et le mode Position.

Ce bloc fonction permet de commuter la source du générateur de dérivation.

MC_CamCalRtoR_MX

Calcul de courbes imbriquées à partir de deux à six points de coordonnées. Le début de la courbe est toujours à 0 / 0 (X1 / Y1).



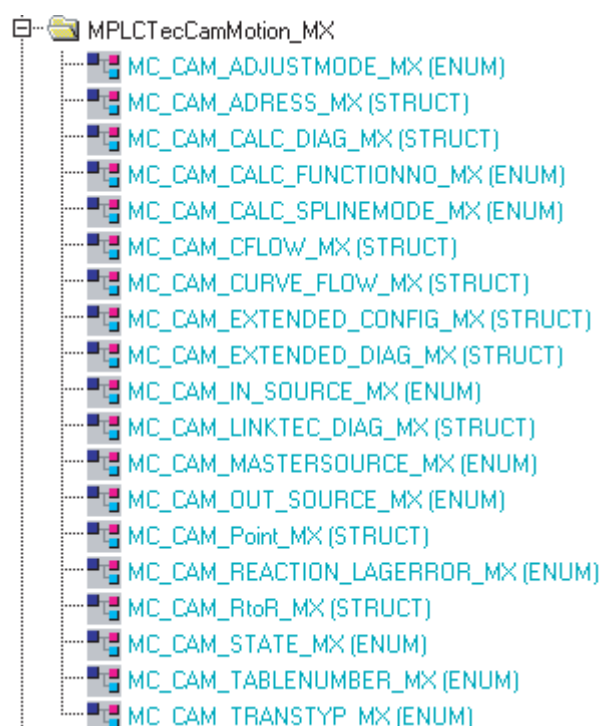
MC_CamCalRtoR50_MX

Calcul de courbes Rest to Rest à partir de deux à 50 points de coordonnées.
L'offset du maître et l'offset de l'esclave sont réglables.

MC_CamCalcSpline_MX

Calcul d'une courbe Spline constituée de 3 à 29 points de coordonnées.

3.4.2 Types de données de la bibliothèque MPLCTecCamMotion_MX



64366axx

MC_CAM_ADJUSTMODE_MX

```
TYPE MC_CAM_ADJUSTMODE_MX :(  
    MX_CAMINDIRECT      := 0,  (*cam in direct without adjust, init with XMasterPosZero*)  
    MX_ADJUST            := 1   (*move the slave to XMasterPosAct *)  
);  
END_TYPE
```

64367axx

**MC_CAM_ADRESS_MX**

TYPE MC_CAM_ADRESS_MX :

STRUCT

Variable_IO	: DINT := 3;	(*DDB Startadress Vatiabile IO*)
PosSetPointGen	: DINT := 24;	(*DDB Startadress PositionSetpointGenerator*)
Gear	: DINT := 61;	(*DDB Startadress Gear*)
SystemData	: DINT := 104;	(*DDB Startadress Systemdata*)
VirtualEncoderMX	: DINT := 149;	(*DDB Startadress Virtual Encoder*)
Messtaster_1	: DINT := 189;	(*DDB Startadress Messtaster 1*)
Messtaster_2	: DINT := 226;	(*DDB Startadress Messtaster 2*)
Messtaster_3	: DINT := 263;	(*DDB Startadress Messtaster 3*)
Messtaster_4	: DINT := 300;	(*DDB Startadress Messtaster 4*)
Databuffer_1	: DINT := 339;	(*DDB Startadress Databuffer 1*)
Databuffer_2	: DINT := 430;	(*DDB Startadress Databuffer 2*)
Databuffer_3	: DINT := 521;	(*DDB Startadress Databuffer 3*)
Databuffer_4	: DINT := 612;	(*DDB Startadress Databuffer 4*)
EventControl	: DINT := 705;	(*DDB Startadress Event Control*)
Cam_Controller	: DINT:= 815;	(* 0*) (*DDB Startadress Cam Control*)
Cam_1	: DINT :=1401;	(* 819*) (*DDB Startadress CAMFlow1 *)
Cam_2	: DINT :=2021;	(*1439*) (*DDB Startadress CAMFlow2 *)
Cam_3	: DINT :=2173;	(*159;*) (*DDB Startadress CAMFlow3 *)
ProfGen	: DINT :=2271;	(*1689*) (*DDB Startadress CAM Profilgenerator*)
DerivateGen	: DINT :=2304;	(*1722*) (*DDB Startadress CAM DerivateGenerators/MotorManager*)
Motormangment	: DINT :=2317;	(*1735*) (*DDB Startadress CAM Motormanagment*)

END_STRUCT

64368axx

MC_CAM_CALC_DIAG_MX

TYPE MC_CAM_CALC_DIAG_MX :

STRUCT

(*parameter fbcamAnalyzer*)

Vmax	:DINT;	(*maximum speed*)
Vmin	:DINT;	(*minimum speed*)
AbsVmax	:DINT;	(*absolut maximum speed*)
Amax	:DINT;	(*maximum acceleration*)
Amin	:DINT;	(*minimum acceleration*)
AbsAmax	:DINT;	(*absolut maximum acceleration*)
Mastercycle	:DINT;	(*mastercyclescale*)
Shiftfaktor	:UINT;	(*Slave RightShift Faktor Default 10*)
State	:BYTE;	(*state machine*)
MasterShiftTab	:DINT;	(* the table point of the shift masterposition only used by RtoR50*)

END_STRUCT

END_TYPE

64369axx



MC_CAM_CALC_FUNCTIONNO_MX

```
TYPE MC_CAM_CALC_FUNCTIONNO_MX :(  
  MX_CAM_STRAIGHT           :=0, (*straight line*)  
  MX_CAM_POLYNOMIAL_5       :=1, (*polynomial 5th order *)  
  MX_CAM_SINUS              :=2, (*sine curve*)  
  MX_CAM_POLYNOMIAL_3       :=3, (*polynomial 3rd order *)  
  MX_CAM_BESTEHORN_SINOIDE  :=4, (*sinoide Bestehorn*)  
  MX_CAM_ACC_TRAPEZOID      :=5); (*modified trapezoid*)  
END_TYPE
```

64370axx

MC_CAM_CALC_SPLINEMODE_MX

```
TYPE MC_CAM_CALC_SPLINEMODE_MX :(  
  MX_CAM_Spline_0           :=0, (* Spline-Interpolation *)  
  MX_CAM_Spline_B           :=1, (*Approximation - Submode 0 - polynomial 3rd order  
                                   - Submode 1 - not used  
                                   - Submode 2 - polynomial 2nd order  
                                   - Submode 3 - polynomial 3rd order  
                                   - Submode 4 - polynomial 4th order  
                                   - Submode 5 - polynomial 5th order  
                                   - Submode 6 - polynomial 6th order  
                                   - Submode 7 - polynomial 7th order*)  
  MX_CAM_Spline_1           :=2 (* fast, piecewise Interpolation *)  
);  
END_TYPE
```

64371axx

MC_CAM_CFLOW_MX

```
TYPE MC_CAM_CFLOW_MX :  
STRUCT  
  Curvenumber: MC_CAM_TABLENUMBER_MX:=1; (*number of the curve*)  
  Nextcurve: MC_CAM_TABLENUMBER_MX:=1; (*next curve pos direction*)  
  Transitionmaster: DINT; (*transision during ...master units*)  
  YSlaveoffsetadd: DINT; (*slaveoffset between the curve included the difference off the old curve*)  
END_STRUCT  
END_TYPE
```

64372axx

MC_CAM_CURVE_FLOW_MX

```
TYPE MC_CAM_CURVE_FLOW_MX :  
STRUCT  
  Camsegment: ARRAY[1..10] OF MC_CAM_CFLOW_MX;  
END_STRUCT  
END_TYPE
```

64373axx



MC_CAM_EXTENDED_CONFIG_MX

```

TYPE MC_CAM_EXTENDED_CONFIG_MX :
STRUCT
(* Extended LinkTecCam Configurations *)
  UseTecEditorConfig:      BOOL;          (* based on the PLCdefault TecEditor project additional
                                           Tec configurations can be made, e.g. CAM2/3, Touchprobe, etc. *)
                                           (* True = Tec values are scaled in user units
                                           False = Tec values are in system units (unscaled) *)
  TecFunctionsUserUnitSlave: BOOL:=1;
  TecFunctionCamExpan:     BOOL;          (* False = it is only to compress a cam curve possible
                                           True = it is also allowed to expan a cam curve *)
  TecFunctionTransition:   MC_CAM_TRANS_TYP_MX:=4; (*
                                           MX_CAM_TransPolynomial3:=1, Transition with Ploynom 3
                                           MX_CAM_TransPolynomial5:=2, Transition with Ploynom 5
                                           MX_CAM_LPG2:=3,          Transition with LPG2
                                           MX_CAM_LPG1:=4          Transition with LPG1
                                           *)
  Calc_SlaveShift_max      : DINT := 10;  (* limit of the calculate SlaveShift *)

  stInitialParameter_MX : MC_INITIALPARAMETER;
  (*#####*)
  (* With this struct several user defined parameters can be written to the axis during the
  initialization of MC_ConnectAxis_MX. DHx11B : 10 parameters, DHx41B 60 parameters

  Add the following code to your program to initialise the struct :

  Example: write DDB variable 2577 and 3420:

  stInitialParameter_MX.aIndex[1] := 20220; (* Index of DDB Variable 2577 *)
  stInitialParameter_MX.aSubIndex[1] := 18; (* SubIndex of DDB Variable 2577 *)
  stInitialParameter_MX.aData[1] := 10;    (* Data to be written to DDB Variable 2577 *)

  stInitialParameter_MX.aIndex[2] := 20226; (* Index of DDB Variable 3420 *)
  stInitialParameter_MX.aSubIndex[2] := 93; (* SubIndex of DDB Variable 3420 *)
  stInitialParameter_MX.aData[2] := 20;    (* Data to be written to DDB Variable 3420 *)

  If there are more than 10 parameters to write, declare NUMOF_INITIALPARAMETER
  in your program.

  VAR_GLOBAL CONSTANT
    NUMOF_INITIALPARAMETER : UDINT := 20;
  END_VAR

  #####*)

(*##### Extended SetCamConfig configurations #####*)
(* Extended SetCamConfig configurations *)
  MSignal_InterpolationTime: DINT;      (* InterpolationTime: range 1...60 resolution [0.5ms] (0.5....30ms) *)
  MSignal_DelayTime:         DINT;      (* PositionDelayTime in [µs] *)
  AverageFilterTime:         DINT;      (* AverageFilterTime: range 1...60 resolution [0.5ms] (0.5....30ms) *)
  CompFilterTime:            DINT;      (* CompFilterTime: range 1...60 resolution [0.5ms] (0.5....30ms) *)
  DeathTimeCorr_ON:          BOOL;      (* Activation of the death time correction: false = OFF, true = ON *)
  SpeedCorr_ON:              BOOL;      (* Activation of the velocity related correction: false = OFF, true = ON *)
  AccelerationCorr_ON:       BOOL;      (* Activation of the acceleration related correction: false = OFF, true = ON *)
  ReactionLagError:          MC_CAM_REACTION_LAGERROR_MX := 10; (* defines the reaction after lag detection. Possible se

  MX_CAM_NO_RESPONSE := 0,
  MX_CAM_DISPLAY_ONLY := 1,
  MX_CAM_STOP_APPL_LIMIT_AUTORESET_NO_HISTORY := 21,
  MX_CAM_STOP_APPL_LIMIT_AUTORESET := 17,
  MX_CAM_STOP_APPL_LIMIT_WAITING_NO_HISTORY := 13,
  MX_CAM_STOP_APPL_LIMIT_WAITING := 8,
  MX_CAM_STOP_APPL_LIMIT_LOCKED := 9,

  MX_CAM_IMMEDIATE_STOP_AUTORESET_NO_HISTORY := 22,
  MX_CAM_IMMEDIATE_STOP_AUTORESET := 18,
  MX_CAM_IMMEDIATE_STOP_WAITING_NO_HISTORY := 14,
  MX_CAM_IMMEDIATE_STOP_WAITING := 6,
  MX_CAM_IMMEDIATE_STOP_LOCKED := 3,

  MX_CAM_STOP_SYSTEM_LIMIT_AUTORESET_NO_HISTORY := 23,
  MX_CAM_STOP_SYSTEM_LIMIT_AUTORESET := 19,
  MX_CAM_STOP_SYSTEM_LIMIT_WAITING_NO_HISTORY := 15,
  MX_CAM_STOP_SYSTEM_LIMIT_WAITING := 10,
  MX_CAM_STOP_SYSTEM_LIMIT_LOCKED := 11,

  MX_CAM_INHIBIT_AUTORESET_NO_HISTORY := 24,
  MX_CAM_INHIBIT_AUTORESET := 20,
  MX_CAM_INHIBIT_WAITING_NO_HISTORY := 16,
  MX_CAM_INHIBIT_WAITING := 5,
  MX_CAM_INHIBIT_LOCKED := 2,
  MX_CAM_INHIBIT_LOCKED_OVERRIDE := 25,

  MX_CAM_SYSTEM_INTERNAL_WAITING := 12,
  MX_CAM_STOP_WAITING := 7,
  MX_CAM_STOP_LOCKED := 4,
  MX_CAM_DISPLAY_ERRORHISTORY := 26
  *)
END_STRUCT
END_TYPE

```

64374axx



MC_CAM_EXTENDED_DIAG_MX

```

TYPE MC_CAM_EXTENDED_DIAG_MX :
STRUCT
  CamFlow1ActCurve      :UINT;  (*Cam Flow 1 status actual curve *)
  CamFlow2ActCurve      :UINT;  (*Cam Flow 2 status actual curve *)
  Modulo_underflow      :DINT;  (*Modulo limit underflow*)
  Modulo_overflow       :DINT;  (*Modulo limit overflow*)

  Calc_SlaveShift_max   :DINT;  (* limit of the calculate SlaveShift*)
  number_mastercycles   :UINT:=1; (*number of mastercycles after tableselect*)
  CamEditor_VersionSubversion :DINT; (*version off the cam editor*)
  CurveChanged          :DINT; (* Bit-0 one or more Curve are changed
                                Bit-1 Curve 1 is changed
                                Bit-2 Curve 2 is changed
                                Bit-3 Curve 3 is changed
                                Bit-4 Curve 4 is changed
                                Bit-5 Curve 5 is changed
                                Bit-6 Curve 6 is changed
                                Bit-7 Curve 7 is changed
                                Bit-8 Curve 8 is changed
                                Bit-9 Curve 9 is changed
                                Bit-10 Curve 10 is changed*)

  NumberofCurve         :DINT; (*number off designed curve*)
  Tablepoints           :DINT; (*number of tablepoints*)
  SlaveShift            :DINT; (*all curve points are shifted with the Slave Shift *)
  MasterShift           :DINT; (*the master position is shifted *)
  Mastercycle_lenght :
  ARRAY[1..10] OF DINT; (*Mastercyclelenght Curve 1 -10
                        Mastercycle_lenght_1 :DINT; Mastercyclelenght Curve 1
                        Mastercycle_lenght_2 :DINT; Mastercyclelenght Curve 2
                        Mastercycle_lenght_3 :DINT; Mastercyclelenght Curve 3
                        Mastercycle_lenght_4 :DINT; Mastercyclelenght Curve 4
                        Mastercycle_lenght_5 :DINT; Mastercyclelenght Curve 5
                        Mastercycle_lenght_6 :DINT; Mastercyclelenght Curve 6
                        Mastercycle_lenght_7 :DINT; Mastercyclelenght Curve 7
                        Mastercycle_lenght_8 :DINT; Mastercyclelenght Curve 8
                        Mastercycle_lenght_9 :DINT; Mastercyclelenght Curve 9
                        Mastercycle_lenght_10 :DINT; Mastercyclelenght Curve 10*)

  Transition_mastercycle :
  ARRAY [11..20] OF DINT;(*Mastercyclelenght Transition Block 11 -20
                        Transition_Mastercycle_11 :DINT; Mastercyclelenght Curve 11
                        Transition_Mastercycle_12 :DINT; Mastercyclelenght Curve 12
                        Transition_Mastercycle_13 :DINT; Mastercyclelenght Curve 13
                        Transition_Mastercycle_14 :DINT; Mastercyclelenght Curve 14
                        Transition_Mastercycle_15 :DINT; Mastercyclelenght Curve 15
                        Transition_Mastercycle_16 :DINT; Mastercyclelenght Curve 16
                        Transition_Mastercycle_17 :DINT; Mastercyclelenght Curve 17
                        Transition_Mastercycle_18 :DINT; Mastercyclelenght Curve 18
                        Transition_Mastercycle_19 :DINT; Mastercyclelenght Curve 19
                        Transition_Mastercycle_20 :DINT; Mastercyclelenght Curve 20*)

  Slave_deviation:
  ARRAY[1..10] OF DINT; (*Slave deviation from the statpoint to the endpoint of a curve
                        Slave_deviation_1 - difference curve 1
                        Slave_deviation_2 - difference curve 2
                        Slave_deviation_3 - difference curve 3
                        Slave_deviation_4 - difference curve 4
                        Slave_deviation_5 - difference curve 5
                        Slave_deviation_6 - difference curve 6
                        Slave_deviation_7 - difference curve 7
                        Slave_deviation_8 - difference curve 8
                        Slave_deviation_9 - difference curve 9
                        Slave_deviation_10 - difference curve 10*)

  Slave_OffsetAdd:
  ARRAY[1..10] OF DINT; (*Slave deviation from the statpoint to the endpoint of a curve
                        Slave_OffsetAdd_1 - offset add difference curve 1
                        Slave_OffsetAdd_2 - offset add difference curve 2
                        Slave_OffsetAdd_3 - offset add difference curve 3
                        Slave_OffsetAdd_4 - offset add difference curve 4
                        Slave_OffsetAdd_5 - offset add difference curve 5
                        Slave_OffsetAdd_6 - offset add difference curve 6
                        Slave_OffsetAdd_7 - offset add difference curve 7
                        Slave_OffsetAdd_8 - offset add difference curve 8
                        Slave_OffsetAdd_9 - offset add difference curve 9
                        Slave_OffsetAdd_10 - offset add difference curve 10*)

  Cam_Adress            :MC_CAM_ADRESS_MX; (*Startadress DDB*)
END_STRUCT
END_TYPE

```

64375axx

**MC_CAM_IN_SOURCE_DIAG_MX**

```

TYPE MC_CAM_IN_SOURCE_MX :
(
    MX_CAM_DIRECT,                                (* CAM in direct with MC_CamInDirect_MX *)
    MX_CAM_INTERRUPT_DI01_POS_FLAG:=16#30,         (* CamIn(Offset) by DI01 interrupt rising edge *)
    MX_CAM_INTERRUPT_DI01_NEG_FLAG:=16#31,         (* CamIn(Offset) by DI01 interrupt falling edge *)
    MX_CAM_INTERRUPT_DI01:=16#32,                 (* CamIn(Offset) by DI01 interrupt both edges *)
    MX_CAM_INTERRUPT_DI02_POS_FLAG:=16#40,         (* CamIn(Offset) by DI02 interrupt rising edge *)
    MX_CAM_INTERRUPT_DI02_NEG_FLAG:=16#41,         (* CamIn(Offset) by DI02 interrupt falling edge *)
    MX_CAM_INTERRUPT_DI02:=16#42,                 (* CamIn(Offset) by DI02 interrupt both edges *)
    MX_CAM_INTERRUPT_DI03_POS_FLAG:=16#50,         (* CamIn(Offset) by DI03 interrupt rising edge *)
    MX_CAM_INTERRUPT_DI03_NEG_FLAG:=16#51,         (* CamIn(Offset) by DI03 interrupt falling edge *)
    MX_CAM_INTERRUPT_DI03:=16#52,                 (* CamIn(Offset) by DI03 interrupt both edges *)
    MX_CAM_INTERRUPT_DI04_POS_FLAG:=16#60,         (* CamIn(Offset) by DI04 interrupt rising edge *)
    MX_CAM_INTERRUPT_DI04_NEG_FLAG:=16#61,         (* CamIn(Offset) by DI04 interrupt falling edge *)
    MX_CAM_INTERRUPT_DI04:=16#62,                 (* CamIn(Offset) by DI04 interrupt both edges *)
    MX_CAM_INTERRUPT_DI05_POS_FLAG:=16#70,         (* CamIn(Offset) by DI05 interrupt rising edge *)
    MX_CAM_INTERRUPT_DI05_NEG_FLAG:=16#71,         (* CamIn(Offset) by DI05 interrupt falling edge *)
    MX_CAM_INTERRUPT_DI05:=16#72,                 (* CamIn(Offset) by DI05 interrupt both edges *)
    MX_CAM_INTERRUPT_DI06_POS_FLAG:=16#80,         (* CamIn(Offset) by DI06 interrupt rising edge *)
    MX_CAM_INTERRUPT_DI06_NEG_FLAG:=16#81,         (* CamIn(Offset) by DI06 interrupt falling edge *)
    MX_CAM_INTERRUPT_DI06:=16#82,                 (* CamIn(Offset) by DI06 interrupt both edges *)
    MX_CAM_INTERRUPT_DI07_POS_FLAG:=16#90,         (* CamIn(Offset) by DI07 interrupt rising edge *)
    MX_CAM_INTERRUPT_DI07_NEG_FLAG:=16#91,         (* CamIn(Offset) by DI07 interrupt falling edge *)
    MX_CAM_INTERRUPT_DI07:=16#92,                 (* CamIn(Offset) by DI07 interrupt both edges *)
    MX_CAM_INTERRUPT_DI08_POS_FLAG:=16#A0,         (* CamIn(Offset) by DI08 interrupt rising edge *)
    MX_CAM_INTERRUPT_DI08_NEG_FLAG:=16#A1,         (* CamIn(Offset) by DI08 interrupt falling edge *)
    MX_CAM_INTERRUPT_DI08:=16#A2,                 (* CamIn(Offset) by DI08 interrupt both edges *)
    MX_CAM_INTERRUPT_C_TRACK_ENC1:=16#2,           (* CamIn(Offset) by C-track interrupt of encoder1 *)
    MX_CAM_INTERRUPT_C_TRACK_ENC2:=16#12,          (* CamIn(Offset) by C-track interrupt of encoder2 *)
    MX_CAM_INTERRUPT_C_TRACK_ENC3:=16#22,          (* CamIn(Offset) by C-track interrupt of encoder3 *)
    MX_CAM_DISTANCE_COUNTER:=16#01FF,              (* CamIn(Offset) by master distance without reset actual counter value *)
    MX_CAM_DISTANCE_COUNTER_RESET_COUNT:=16#02FF   (* CamIn(Offset) by master distance , reset actual counter value *)
);
END_TYPE

```

64376axx

MC_CAM_LINKTEC_DIAG_MX

```

TYPE MC_CAM_LINKTEC_DIAG_MX :
STRUCT
    CamFlow1ActCurve      :UINT;    (*Cam Flow 1 status actual curve number *)
    CamFlow2ActCurve      :UINT;    (*Cam Flow 2 status actual curve number *)

```

64377axx

MC_CAM_MasterPosSHL_MX

```

TYPE MC_CAM_MASTERSOURCE_MX :
(
    MX_CAM_SYSTEMPOS_ENCODER1,    (* Master position is the linear position of encoder 1 *)
    MX_CAM_SYSTEMPOS_ENCODER2,    (* Master position is the linear position of encoder 2 *)
    MX_CAM_SYSTEMPOS_ENCODER3,    (* Master position is the linear position of encoder 3 *)
    MX_CAM_MODULOPOS_ENCODER1,    (* Master position is the modulo position of encoder 1 *)
    MX_CAM_MODULOPOS_ENCODER2,    (* Master position is the modulo position of encoder 2 *)
    MX_CAM_MODULOPOS_ENCODER3,    (* Master position is the modulo position of encoder 3 *)
    MX_CAM_MODULO_VIRTUAL_ENCODER_MX, (* Master position is the modulo position of the MX virtual encoder *)
    MX_CAM_LINEAR_VIRTUAL_ENCODER_MX, (* Master position is the linear position of the MX virtual encoder *)
    MX_CAM_RECEIVE_PDO_1,          (* Master position is the SBus receive objekt PDO1 *)
    MX_CAM_RECEIVE_PDO_2          (* Master position is the SBus receive objekt PDO2 *)
);
END_TYPE

```

64378axx



MC_CAM_POINT_MX

```
TYPE MC_CAM_POINT_MX :
STRUCT
  X      : DINT;    (*masterpoints*)
  Y      : DINT;    (*slavepoints*)
END_STRUCT
END_TYPE
```

64379axx

MC_CAM_REACTION_LAGEERROR_MX

```
TYPE MC_CAM_REACTION_LAGEERROR_MX :
(
  MX_CAM_NO_RESPONSE := 0,
  MX_CAM_DISPLAY_ONLY := 1,
  MX_CAM_STOP_APPL_LIMIT_AUTORESET_NO_HISTORY := 21,
  MX_CAM_STOP_APPL_LIMIT_AUTORESET := 17,
  MX_CAM_STOP_APPL_LIMIT_WAITING_NO_HISTORY := 13,
  MX_CAM_STOP_APPL_LIMIT_WAITING := 8,
  MX_CAM_STOP_APPL_LIMIT_LOCKED := 9,

  MX_CAM_IMMEDIATE_STOP_AUTORESET_NO_HISTORY := 22,
  MX_CAM_IMMEDIATE_STOP_AUTORESET := 18,
  MX_CAM_IMMEDIATE_STOP_WAITING_NO_HISTORY := 14,
  MX_CAM_IMMEDIATE_STOP_WAITING := 6,
  MX_CAM_IMMEDIATE_STOP_LOCKED := 3,

  MX_CAM_STOP_SYSTEM_LIMIT_AUTORESET_NO_HISTORY := 23,
  MX_CAM_STOP_SYSTEM_LIMIT_AUTORESET := 19,
  MX_CAM_STOP_SYSTEM_LIMIT_WAITING_NO_HISTORY := 15,
  MX_CAM_STOP_SYSTEM_LIMIT_WAITING := 10,
  MX_CAM_STOP_SYSTEM_LIMIT_LOCKED := 11,

  MX_CAM_INHIBIT_AUTORESET_NO_HISTORY := 24,
  MX_CAM_INHIBIT_AUTORESET := 20,
  MX_CAM_INHIBIT_WAITING_NO_HISTORY := 16,
  MX_CAM_INHIBIT_WAITING := 5,
  MX_CAM_INHIBIT_LOCKED := 2,
  MX_CAM_INHIBIT_LOCKED_OVERRIDE := 25,

  MX_CAM_SYSTEM_INTERNAL_WAITING := 12,
  MX_CAM_STOP_WAITING := 7,
  MX_CAM_STOP_LOCKED := 4,
  MX_CAM_DISPLAY_ERRORHISTORY := 26
);
END_TYPE
```

64380axx

**MC_CAM_RtoR_MX**

```

TYPE MC_CAM_RtoR_MX :
STRUCT
    Point: ARRAY[1..50] OF MC_CAM_Point_MX;
END_STRUCT
END_TYPE

```

64381axx

MC_CAM_STATE_MX

```

TYPE MC_CAM_STATE_MX :
(
    MX_CAM_OUTCAM           :=0,    (* Synchronous state : Axis is out caming *)
    MX_CAM_INCAM            :=1,    (* Synchronous state : Axis is in caming *)
    MX_CAM_INCAM_ADJUST     :=2,    (* Synchronous state : Axis is in caming *)
    MX_CAM_INCAM_OFFSET     :=3,    (* Synchronous state : Offset move while in cam direct *)
    MX_CAM_INCAM_ADJUST_OFFSET :=4, (* Synchronous state : Offset move while in cam adjust *)
    MX_CAM_MOVE_TO_MASTER   :=5,    (* Synchronous state : Axis is moving to Master after inhibit or safety stop init DX / DV *)
    MX_CAM_NOTLINKED        :=6,    (* Synchronous state : Axis not linked *)
);
END_TYPE

```

64382bxx

MC_CAM_TABLENUMBER_MX

```

TYPE MC_CAM_TABLENUMBER_MX :
(
    CAM_WAIT_MX      :=0,    (*Curve Wait Startpoint 5 Lenght 5*)
    CAM_1_MX         :=1,    (*CamFLoW 1 Curve 1 Startpoint 10 Lenght 512*)
    CAM_2_MX         :=2,    (*CamFLoW 1 Curve 2 Startpoint 522 Lenght 512*)
    CAM_3_MX         :=3,    (*CamFLoW 1 Curve 3 Startpoint 1034 Lenght 512*)
    CAM_4_MX         :=4,    (*CamFLoW 1 Curve 4 Startpoint 1546 Lenght 512*)
    CAM_5_MX         :=5,    (*CamFLoW 1 Curve 5 Startpoint 2058 Lenght 512*)
    CAM_6_MX         :=6,    (*CamFLoW 1 Curve 6 Startpoint 2570 Lenght 512*)
    CAM_7_MX         :=7,    (*CamFLoW 1 Curve 7 Startpoint 3082 Lenght 512*)
    CAM_8_MX         :=8,    (*CamFLoW 1 Curve 8 Startpoint 3594 Lenght 512*)
    CAM_9_MX         :=9,    (*CamFLoW 1 Curve 9 Startpoint 4106 Lenght 512*)
    CAM_10_MX        :=10,   (*CamFLoW 1 Curve 10 Startpoint 2058 Lenght 512*)
    CAM_GEAR_MX      :=11,   (*Curve GEAR Startpoint 0 Lenght 5*)
);
END_TYPE

```

64383axx

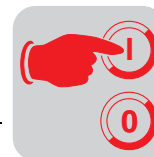
MC_CAM_TRANSTYP_MX

```

TYPE MC_CAM_TRANSTYP_MX :
(
    MX_CAM_TRANSPOLYNOMINAL3:=1, (*Transition with Ploynom 3 / Ya,Va,Ye,Ve*)
    MX_CAM_TRANSPOLYNOMINAL5:=2, (*Transition with Ploynom 5 /Ya,Va,Aa,Ye,Ve,Ae*)
    MX_CAM_LPG2:=3,               (*Transition with LPG2 / Amax*)
    MX_CAM_LPG1:=4                (*Transition with LPG1 / Amax,VmaxP,VmaxN*)
);
END_TYPE

```

64384axx



3.4.3 Configuration avancée

Les fonctions technologiques du MOVIAXIS® offrent de nombreuses possibilités de réglage, de filtrage ou de fonctions de compensation qui ne sont pas nécessaires pour la plupart des applications standard. Pour conserver une bonne vue d'ensemble des blocs fonction tout en ayant une possibilité d'accès, ces paramètres sont résumés dans la structure d'entrée "MC_CAM_EXTENDED_CONFIG_MX".


Cette structure est utilisée tant par le bloc fonction "MC_LinkTecCam_MX" que par le bloc fonction "MC_SetCamConfig_MX".

3.4.4 ExtConfig avec MC_LinkTecCam_MX

UseTecEditorConfig

La variable structure *UseTecEditorConfig* permet, en parallèle à la configuration CAM à l'aide des blocs fonction des bibliothèques, une configuration avancée avec le TecEditor.

Ceci permet p. ex. de régler la fonction de capteur d'événements ou de configurer un bloc de courbe supplémentaire superposé.

	REMARQUES
	<p>Procédure</p> <ul style="list-style-type: none"> • Les plages DDB pouvant dans certaines circonstances se décaler, afin d'éviter les messages de défaut, il faut d'abord stopper le programme CEI par reset (à froid). • Utiliser comme base le projet TecEditor "MOVI-PLCDefaultprojekt.... TecFunction". Le projet peut ensuite être modifié ou élargi selon la nécessité. Il est important dans ce cas de ne pas désactiver les fonctions Tec qui sont activées dans le projet de base. • Charger dans le variateur le projet TecEditor modifié. • Lorsque la variable <i>UseTecEditorConfig</i> a été forcée sur TRUE, le programme peut être lancé.

TecFunctionUserUnitSlave

La variable est initialisée sur TRUE et signifie que toutes les variables système sont interprétées comme unités utilisateur.

En cas de réglage sur FALSE, ce sont les unités système qui sont utilisées (paramètre 20300.1).

TecFunctionCamExpant

Réglage par défaut = FALSE. En raison de l'augmentation automatique de la mise à l'échelle des positions esclave, il n'est normalement pas possible de dilater une courbe à l'échelle numérateur / dénominateur, elle peut uniquement être comprimée. Si cependant, à la configuration de la courbe, la courbe est volontairement maintenue à une taille inférieure via le paramètre "facteur de décalage maximum", il est également possible de dilater la courbe en activant le bit "TecFunctionCamExpant".



TecFunctionTransition

Réglage par défaut = MX_CAM_LPG1. Les fonctions de commutation (transitions), pouvant suivre une courbe mathématique, sont calculées avec un générateur de profil linéaire 1. Il est par conséquent possible de définir une vitesse maximale et une accélération maximale. Cette fonction peut être commutée en conséquence, voir "MC_CAM_TRANSTYP_MX".

stInitialParameter_MX

Permet d'intégrer les paramètres MOVIAXIS® dans un array (tableau) ; ceux-ci sont ensuite transférés dans le variateur via le bloc fonction "MC_LinkTecCam_MX".

3.4.5 ExtConfig avec MC_SetCamConfig_MX

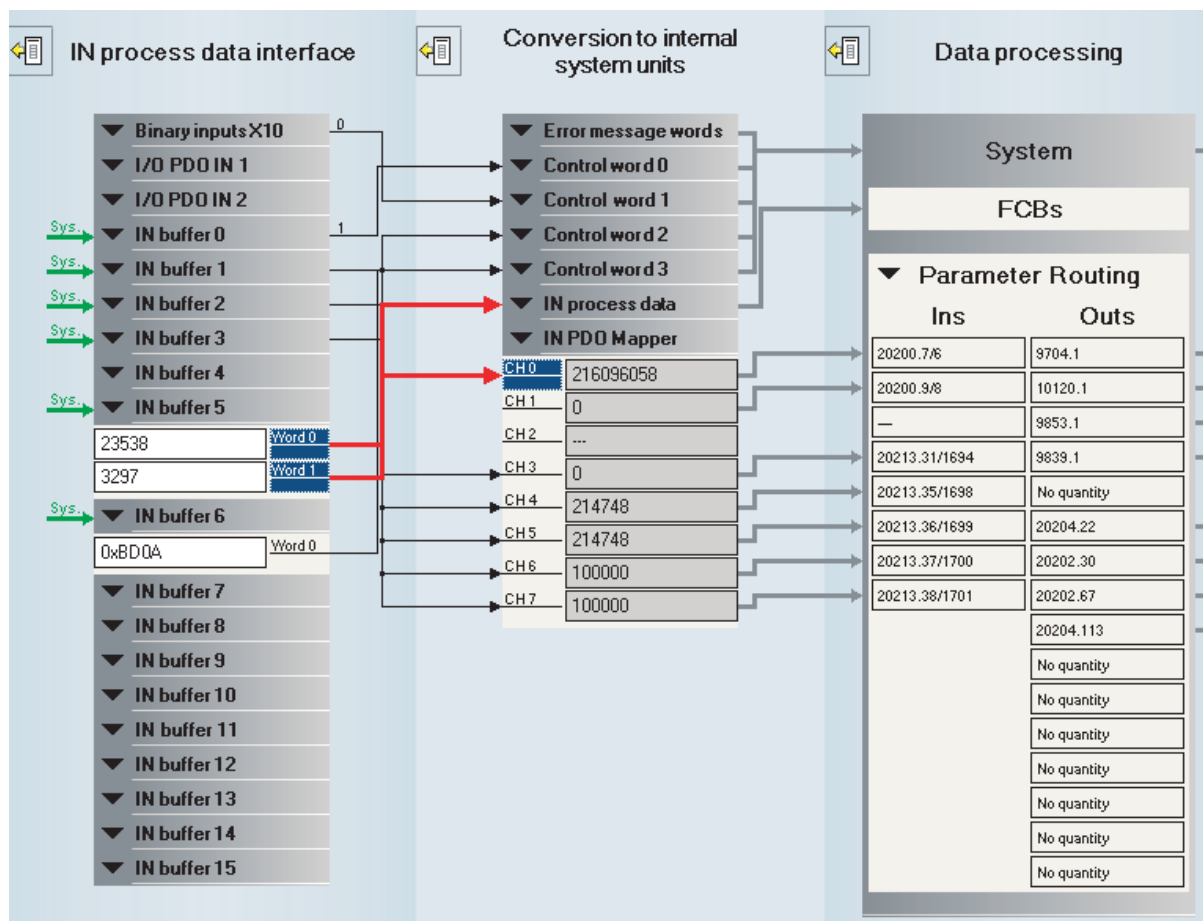
La configuration avancée avec "SetCamConfig" contient des variables permettant de régler des filtres supplémentaires pour la préparation du signal maître.

Les informations détaillées concernant les fonctions et les effets de ces filtres figurent dans la documentation MOVIAXIS® "Fonction technologique réducteur électronique pour MOVIAXIS® MX".

Dans les réglages par défaut, les filtres sont désactivés. En cas d'utilisation du codeur virtuel de la MOVI-PLC®, la valeur de *MSignal_InterpolationTime* est réglée sur 5 ms.

3.4.6 Configuration PDO

En plus de la configuration effectuée par le bloc fonction "MC_ConnectAxis_MX", le bloc fonction "MC_LinkTecCam_MX" effectue également certains réglages sur l'interface de données-process.



64385axx

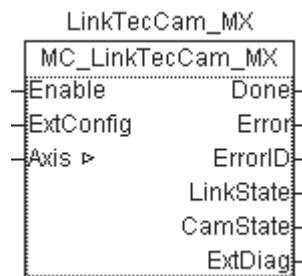


Le tampon IN 5 est réglé pour réceptionner un objet cyclique avec l'ID 2, qui est affectée au canal 0 du mapping PDO IN. Dans la mesure où elle n'est pas configurée différemment, la MOVI-PLC® envoie avec cet identifiant la position réelle du codeur virtuel.

En cas de sélection de "MX_CAM_RECEIVE_PDO_1" comme source maître pour le bloc fonction "MC_SetCamConfig_MX", cet objet est réglé comme valeur maître pour la fonction Cam.

3.5 Les blocs fonction de la bibliothèque MPLCTecCamMotion_MX

3.5.1 MC_LinktecCam_MX



64386axx

Description

Le bloc fonction "MC_LinkTecCam_MX" constitue l'interface logique entre l'axe et la fonction technologique "Cam". Les différentes fonctions technologiques du MOVIAXIS® sont configurées après établissement et activation de la communication cyclique avec l'axe via ConnectAxis. Parmi ces fonctions technologiques, seules celles nécessaires pour la fonction Cam sont activées.

L'interface PDO est également configurée (IN-PDO 5 pour la réception de la position maître via le SBus), et les paramètres de communication pour la synchronisation du SBus réglés.

En mode cyclique, le bloc fonction "MC_LinkTecCam_MX" assure la commutation des courbes mathématiques dès que l'axe est synchronisé. Ceci est nécessaire pour la réalisation du déplacement d'offset.

Diverses informations d'état sont également mises à disposition comme sorties.

Attention :

Le bloc fonction "MC_LinkTecCam_MX" est indispensable pour l'utilisation des fonctionnalités de came électronique et doit être appelé cycliquement dans le programme.

Si *LinkState* = NotLinked, il n'est possible d'exécuter aucun bloc fonction de la bibliothèque "MPLCTecCamMotion_MX" ; un message de défaut correspondant apparaît alors.

Condition :

MOVI-PLC® *advanced* en version technologique T1 ou supérieure.

ConnectAxis pour l'axe correspondant intégré dans le programme cyclique.



Entrées :

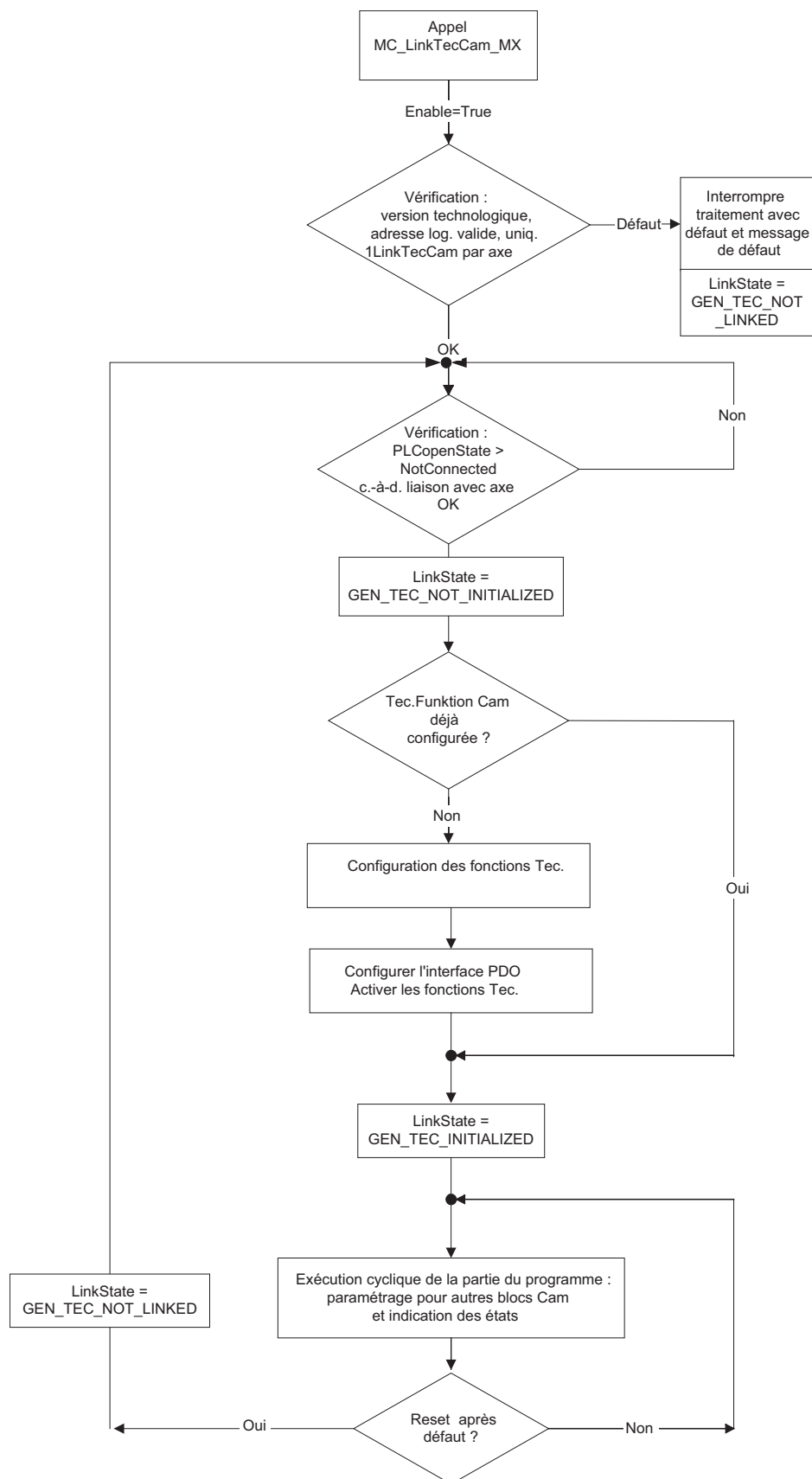
Nom	Type	Signification
Enable	BOOL	Le bloc est traité tant que Enable est = true. L'initialisation est effectuée sur un front montant.
ExtConfig	MC_Cam_EXTENDED_CONFIG_MX	Configuration avancée, voir page 126
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
Done	BOOL	Initialisation effectuée avec succès, liaison avec fonction technologique "Cam" établie.
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 126
LinkState	MC_LINKTECSTATE	Etat actuel : GEN_TEC_NOTLINKED (pas de liaison) GEN_TEC_NOTINITIALISED (pas d'initialisation effectuée) GEN_TEC_INITIALISED (fonction tec. Cam initialisée)
CamState	MC_Cam_STATE_MX	Etat actuel : MX_CAM_OUTCAM : =0, MX_CAM_INCAM : =1 MX_CAM_INCAM_ADJUST : =2, MX_CAM_INCAM_OFFSET : =3 MX_CAM_INCAM_ADJUST_OFFSET : = 4 MX_CAM_MOVE_TO_MASTER : = 5 MX_CAM_NOTLINKED : = 6
ExtDiag	MC_Cam_EXTENDED_DIAG_MX	Information de diagnostic CamFlow1ActCurve : UINT; Affichage du numéro de courbe activé dans le profil de courbe 1 0 = courbe Cam_Wait_MX activée 1 = courbe Cam_1_MX activée 2 = courbe Cam_2_MX activée 3 = courbe Cam_3_MX activée 4 = courbe Cam_4_MX activée 5 = courbe Cam_5_MX activée 6 = courbe Cam_6_MX activée 7 = courbe Cam_7_MX activée 8 = courbe Cam_8_MX activée 9 = courbe Cam_9_MX activée 10 = courbe Cam_10_MX activée 11 = transition Cam_1_MX activée 12 = transition Cam_2_MX activée 13 = transition Cam_3_MX activée 14 = transition Cam_4_MX activée 15 = transition Cam_5_MX activée 16 = transition Cam_6_MX activée 17 = transition Cam_7_MX activée 18 = transition Cam_8_MX activée 19 = transition Cam_9_MX activée 20 = transition Cam_10_MX activée CamFlow2ActCurve:UINT; Affichage du numéro de courbe activé dans le profil de courbe 2 0 = Cam_Wait_MX activé 1 = offset de transition activé 2 = Cam_Wait_MX activé



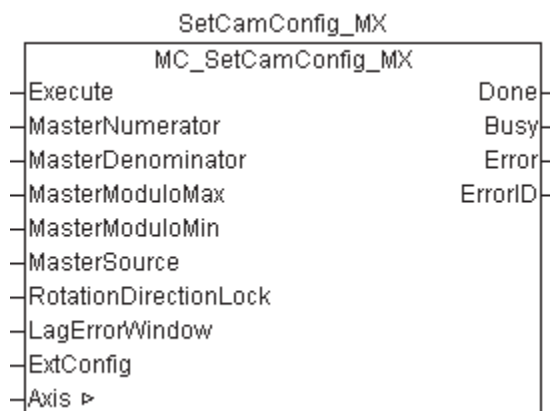
MC_LinktecCam_MX : Initialisation



64388afr



3.5.2 MC_SetCamConfig_MX



64389axx

Description

Ce bloc fonction règle les paramètres de base de la fonction Cam et les transmet à l'axe concerné. Les valeurs réglées sont prises en compte et envoyées lorsqu'un front montant est appliqué sur l'entrée Execute.

L'entrée *MasterSource* permet d'indiquer la source du signal codeur maître. Il est possible d'opter soit pour des codeurs "réels" raccordés sur l'axe, soit pour des signaux maître "virtuels" envoyés via le SBus. Voir également à ce sujet le chapitre "Types de données de la bibliothèque MPLCTecCamMotion_MX", page 69 et suivantes.

Le signal du maître peut être mis à l'échelle via les paramètres "MasterNumerator" et "MasterDenominator".

Exemple :

Signal de 65536 incr par cycle maître

- MasterNumerator = 62500
- MasterDenominator = 4096

c.-à-d. qu'un cycle maître correspond à 1000000 unités utilisateur [mm].

Les courbes peuvent être configurées avec des unités-maître en mm.

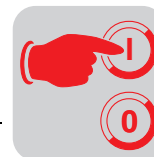
Si le signal maître est transmis au format modulo, régler les entrées *ModuloMax / ModuloMin* aux valeurs maximales respectives du signal maître. Ceci permet d'éviter des sauts de consigne en cas de dépassement de la plage modulo. En cas de signal maître linéaire, le réglage de ces valeurs est libre.

Il est possible de transférer une structure pour le réglage de divers filtres via l'entrée *ExtConfig*. Il est ainsi p. ex. possible de filtrer un signal maître "brouillé". Ces filtres sont réglés en interne sur des valeurs standard adéquates, si bien qu'il n'est généralement pas nécessaire d'affecter cette entrée.

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

LinkState ne doit pas passer à GEN_TEC_NOTLINKED.



Entrées :

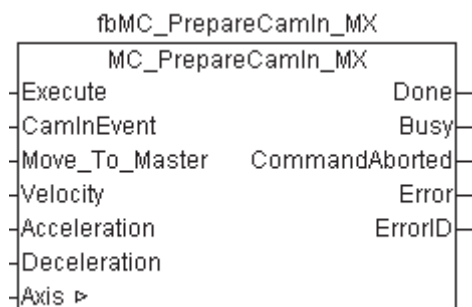
Nom	Type	Signification
Execute	BOOL	Cette entrée lance l'exécution du bloc fonction sur un front montant
MasterNumerator	DINT	Numérateur pour la mise à l'échelle du signal maître
MasterDenominator	UDINT	Dénominateur pour la mise à l'échelle du signal maître
ModuloMax	DINT	Valeur modulo maximale signal maître
ModuloMin	DINT	Valeur modulo minimale signal maître
MasterSource	MC_CAM_MASTERSOURCE_MX	Source de la position maître MX_CAM_SYSTEMPOS_ENCODER1 MX_CAM_SYSTEMPOS_ENCODER2 MX_CAM_SYSTEMPOS_ENCODER3 MX_CAM_MODULOPOS_ENCODER1 MX_CAM_MODULOPOS_ENCODER2 MX_CAM_MODULOPOS_ENCODER3 MX_CAM_MODULO_VIRTUAL_ENCODER_MX MX_CAM_LINEAR_VIRTUAL_ENCODER_MX MX_CAM_RECEIVE_PDO_1 MX_CAM_RECEIVE_PDO_2
RotationDirectionLock	BYTE	Verrouillage sens de marche 0 = libération des deux sens 1 = libération uniq. gauche 2 = libération uniq. droite
LagErrorWindow	UDINT	Tolérance erreur de poursuite en [inc]
ExtConfig	MC_CAM_EXTENDED_CONFIG_MX	Configuration avancée : MSignal_InterpolationTime en [0.5ms] MSignal_DelayTime en [µs] AverageFilterTime en [0.5ms] CompFilterTime en [0.5ms] DeathTimeCorr_On [on / off]
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	Tous les paramètres ont été transmis correctement
Busy	BOOL	Transfert de paramètres en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 126



3.5.3 MC_PrepareCamIn_MX



64390axx

Description

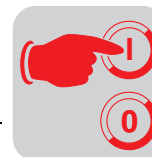
Ce bloc fonction permet d'activer la synchronisation commandé par interruption et la fonction *MovetoMaster*.

Les paramètres de dynamisme pour le calage après un verrouillage à l'aide de la fonction *MovetoMaster* sont également modifiés.

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

LinkState ne doit pas passer à GEN_TEC_NOTLINKED.



Entrées :

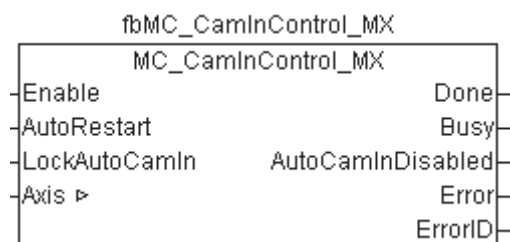
Nom	Type	Signification
Execute	BOOL	Cette entrée lance l'exécution de la commande du bloc fonction
CamInEvent	MC_Cam_IN_SOURCE_MX	<p>Evénements de synchronisation possibles :</p> <p>MX_CAM_DIRECT MX_CAM_INTERRUPT_DI01_POS_FLAG MX_CAM_INTERRUPT_DI01_NEG_FLAG MX_CAM_INTERRUPT_DI01 ... MX_CAM_INTERRUPT_DI08_POS_FLAG MX_CAM_INTERRUPT_DI08_NEG_FLAG MX_CAM_INTERRUPT_DI08 MX_CAM_INTERRUPT_C_TRACK_ENC1 MX_CAM_INTERRUPT_C_TRACK_ENC2 MX_CAM_INTERRUPT_C_TRACK_ENC3</p> <p>Explication : DIRECT = synchronisation directe POS_FLAG = front positif NEG_FLAG = front négatif uniqu. l'entrée = les deux fronts DI01 de MOVIAXISDI01 = entrée DI02 de MOVIAXISDI02 = entrée DI03 de MOVIAXISDI03 = entrée DI04 de MOVIAXISDI04 = entrée DI05 de MOVIAXISDI05 = entrée DI06 de MOVIAXISDI06 = entrée DI07 de MOVIAXISDI07 = entrée DI08 de MOVIAXISDI08 = entrée C_TRACK_ENC1 = voie C codeur 1 C_TRACK_ENC2 = voie C codeur 2 C_TRACK_ENC3 = voie C codeur 3</p>
MovetoMaster	BOOL	True : fonction activée pour MC_CamInDirekt_MX False : fonction activée pour MC_CamInDirekt_MX
Velocity	DINT	Velocity [rpm] or User unit scaling
Acceleration	DINT	Ramp up time in ms relating to 3000 rpm /or User unit scaling
Deceleration	DINT	Ramp down time in ms relating to 3000 rpm /or User unit scaling
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	Tous les paramètres ont été transmis correctement
Busy	BOOL	Transfert de paramètres en cours
CommandAborted	BOOL	L'exécution de la commande du bloc fonction a été interrompue par une autre instance d'un bloc fonction PrepareCamIn
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 126



3.5.4 MC_CamInControl_MX



64395axx

Description

Le bloc fonction "MC_CamInControl_MX" offre des fonctions de pilotage supplémentaires pour les événements de synchronisation gérés par interruption ou par compteur.

L'entrée *AutoRestart* permet de répéter une synchronisation à l'aide du bloc fonction "PrepareCamIn" sans nécessiter une nouvelle préparation. Il est également possible de verrouiller la synchronisation gérée par interruption en forçant l'entrée *LockAutoCamIn* sur True. La synchronisation est également verrouillée après un défaut sur l'axe (Errorstop). Dans ce cas, il faut réactiver la synchronisation gérée par interruption via le bloc fonction "PrepareCamIn".

Le verrouillage de la synchronisation est affiché par la sortie *AutoCamInDisabled*.

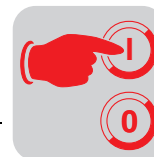
Attention :

Le verrouillage de la synchronisation n'est possible que pour les événements de synchronisation gérés par interruption ; la synchronisation via "MC_CamInDirect_MX" ou "MC_CamInAdjust_MX" reste toujours possible.

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".



Entrées :

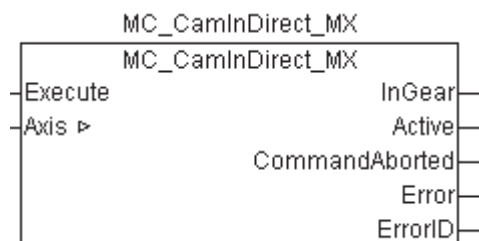
Nom	Type	Signification
Enable	BOOL	Le bloc fonction est traité tant que cette entrée = True
AutoRestart	BOOL	True : réactivation automatique de la synchronisation activée False : synchronisation désactivée
LockAutoCamIn	BOOL	True : synchronisation verrouillée False : synchronisation libérée
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	Fonction souhaitée verrouillée
Busy	BOOL	Transfert de paramètres en cours
AutoCamInDisabled	BOOL	Synchronisation verrouillée
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 126



3.5.5 MC_CamInDirect_MX



64398axx

Description

L'application d'un front montant sur l'entrée Execute provoque une synchronisation directe.

Le bloc fonction interrompt les blocs fonction "Move" encore activés.

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

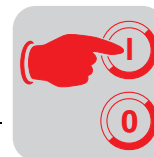
CamState = "MX_CAM_OUTCAM".

Entrées :

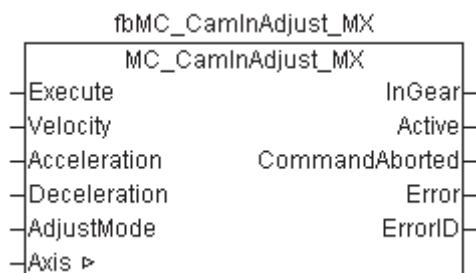
Nom	Type	Signification
Execute	BOOL	Cette entrée lance la synchronisation directe
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
InGear	BOOL	Synchronisation correcte, l'axe est synchronisé avec le maître sur la came électronique.
Active	BOOL	L'axe est en cours de synchronisation.
CommandAborted	BOOL	La tâche du bloc fonction a été interrompue par l'appel d'un autre bloc fonction Move.
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 126



3.5.6 MC_CamInAdjust_MX



64400axx

Description

Si le mode d'ajustement sélectionné est égal à MX_CAMINDIRECT, la came électronique est initialisée par un front montant sur l'entrée Execute et commutée en mode came électronique FCB 16 à sa position actuelle.

Si le mode d'ajustement sélectionné est égal à MX_ADJUST, la came électronique est calée sur la position actuelle de la courbe avec les paramètres de dynamisme sélectionnés par un front montant sur l'entrée Execute et le bloc fonction FCB 16 est commuté en mode came électronique. Après un verrouillage, un arrêt d'urgence etc., l'axe s'oriente en conséquence.

Après une coupure du réseau, l'axe doit à nouveau être déplacé vers la position de synchronisation et être synchronisé via le mode d'ajustement MX_CAMINDIRECT, car l'axe perd sa référence à la courbe. La prochaine synchronisation pourra s'effectuer via le mode d'ajustement MX_Adjust.

Le bloc fonction interrompt l'exécution des blocs fonction "Move" encore activés.

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

CamState = "MX_CAM_OUTCAM".

L'axe doit être référencé.

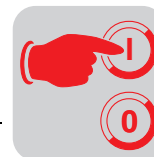


Entrées :

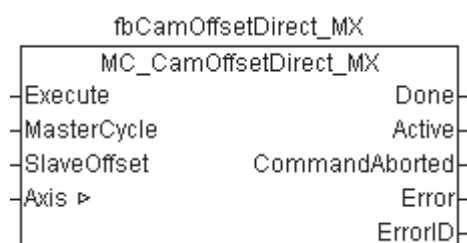
Nom	Type	Signification
Execute	BOOL	Cette entrée lance la synchronisation directe
Velocity	DINT	Velocity [rpm] or User unit scaling
Acceleration	DINT	Ramp up time in ms relating to 3000 rpm /or User unit scaling
Deceleration	DINT	Ramp down time in ms relating to 3000 rpm /or User unit scaling
Adjustmode	MC_CAM_ADJUSTMODE_MX	MX_CAMINDIRECT (0) La position maître de la came électronique est initialisée à 0 (XMASTERPOSZERO) MX_ADJUST(1) Position maître actuelle de la came électronique (XMASTERPOSAKT)
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
InGear	BOOL	Synchronisation correcte, l'axe est synchronisé avec le maître sur la came électronique
Active	BOOL	L'axe est en cours de synchronisation
CommandAborted	BOOL	La tâche du bloc fonction a été interrompue par l'appel d'un autre bloc fonction Move
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 126



3.5.7 MC_CamOffsetDirect_MX



64401axx

Description

Ce bloc fonction ajoute un offset g r  en vitesse/temps   la courbe actuelle.

L' tat *CamState* est commut .

MX_Cam_InCam -> MX_Cam_InCam_Offset

MX_Cam_InCamAdjust -> MX_Cam_InCamAdjust_Offset

Apr s ex cution de l'offset, l'axe repasse au mode came  lectronique normal.

Le *CamState* repasse   l' tat pr c dent.

Condition :

"MC_LinkTecCam_MX" a configur  la fonction technologique "Cam".

LinkState ne doit pas passer   "GEN_TEC_NOTLINKED".

L'axe est en mode came  lectronique.

CamState = MX_Cam_InCam ou *CamState* = MX_Cam_InCamAdjust.

Entr es :

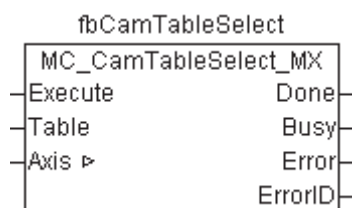
Nom	Type	Signification
Execute	BOOL	Cette entr�e active le d�placement d'offset.
MasterCycle	DINT	Distance ma�tre pour le d�placement d'offset (delta X)
SlaveOffset	DINT	D�placement d'offset que l'esclave doit parcourir en plus de la courbe (deltaY)
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	L'offset a �t� trait�, l'axe est � nouveau synchronis�.
Active	BOOL	Gestion d'offset en cours
CommandAborted	BOOL	La t�che du bloc fonction a �t� interrompue par l'appel d'un autre bloc fonction Move.
Error	BOOL	D�faut dans le bloc fonction durant l'ex�cution
ErrorID	UINT	Voir page 126



3.5.8 MC_CamTableSelect_MX



64415axx

Description

Ce bloc fonction permet de sélectionner une courbe. Si une autre courbe est activée, la commutation s'effectue lorsque la fin de la courbe active est atteinte.

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

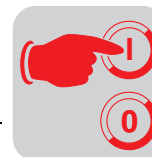
LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

Nom	Type	Signification
Execute	BOOL	Cette entrée active le déplacement d'offset.
Table	MC_CAM_TABLENUMBER_MX	Sélection de la courbe Cam_Wait_MX (0), sélection de la courbe d'attente CAM_1_MX (1), sélection courbe 1 CAM_2_MX (2), sélection courbe 2 CAM_3_MX (3), sélection courbe 3 CAM_4_MX (4), sélection courbe 4 CAM_5_MX (5), sélection courbe 5 CAM_6_MX (6), sélection courbe 6 CAM_7_MX (7), sélection courbe 7 CAM_8_MX (8), sélection courbe 8 CAM_9_MX (9), sélection courbe 9 CAM_10_MX (10), sélection courbe 10
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	La courbe a été sélectionnée.
Busy	BOOL	Traitement du bloc fonction en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	DWORD	Voir page 126



3.5.9 MC_CamTableSelectTrans_MX



64416axx

Description

Ce bloc fonction permet de sélectionner une courbe par une transition raccordée.

Si une autre courbe est activée, la commutation s'effectue lorsque la fin de la courbe activée est atteinte. L'entrée *Transitionmaster* permet d'indiquer la trajectoire du maître (Delta X) et l'entrée *YSlaveoffsetadd* la distance à parcourir par l'esclave. Le déplacement est effectué via LPG1. Une commutation est possible via la configuration avancée, voir *entrée ExtendedCamConfig*.

Type de donnée TecFunktionTransition "MC_CAM_TRANSTYP_MX".

Les paramètres de dynamisme peuvent être modifiés via le bloc fonction "MC_SetCamTransition Dynamics_MX".

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

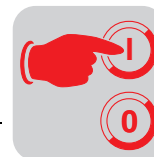
LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

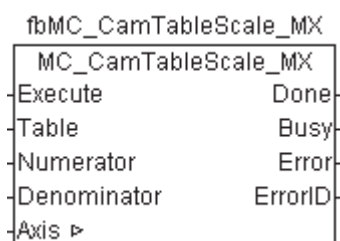
Nom	Type	Signification
Execute	BOOL	Cette entrée active le déplacement d'offset.
Table	MC_CAM_TABLENUMBER_MX	Sélection de la courbe Cam_Wait_MX (0), sélection de la courbe d'attente CAM_1_MX (1), sélection courbe 1 CAM_2_MX (2), sélection courbe 2 CAM_3_MX (3), sélection courbe 3 CAM_4_MX (4), sélection courbe 4 CAM_5_MX (5), sélection courbe 5 CAM_6_MX (6), sélection courbe 6 CAM_7_MX (7), sélection courbe 7 CAM_8_MX (8), sélection courbe 8 CAM_9_MX (9), sélection courbe 9 CAM_10_MX (10), sélection courbe 10
Transitionmaster	DINT	Distance maître de la fonction de transition
YSlaveoffsetadd	DINT	Trajectoire esclave de la fonction de transition
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

**Sorties :**

Nom	Type	Signification
Done	BOOL	La courbe a été sélectionnée.
Busy	BOOL	Traitement du bloc fonction en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	DWORD	Voir page 126



3.5.10 MC_CamTableScale_MX



64417axx

Description

Ce bloc fonction permet de comprimer une courbe, c.-à-d. dénominateur > numérateur. Si une courbe doit être dilatée, il est possible de forcer le bit *ExtendedCamConfig.TecFunctionCamExpant* sur TRUE dans la configuration avancée. Il est ensuite autorisé de dilater la courbe.

En cas d'utilisation de cette fonction, limiter en conséquence le facteur *SlaveShiftFaktor* ou configurer les courbes en conséquence.

Recommandation : **en principe, les courbes doivent uniquement être comprimées.**

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

La courbe sélectionnée ne doit pas être activée.

Entrées :

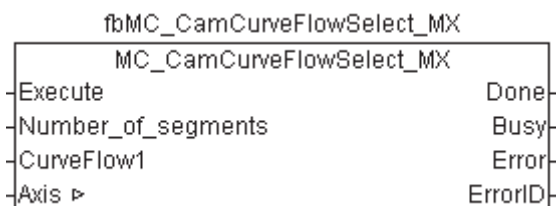
Nom	Type	Signification
Execute	BOOL	Cette entrée active le déplacement d'offset.
Table	MC_CAM_TABLENUMBER_MX	Sélection de la courbe Cam_Wait_MX (0), sélection de la courbe d'attente CAM_1_MX (1), sélection courbe 1 CAM_2_MX (2), sélection courbe 2 CAM_3_MX (3), sélection courbe 3 CAM_4_MX (4), sélection courbe 4 CAM_5_MX (5), sélection courbe 5 CAM_6_MX (6), sélection courbe 6 CAM_7_MX (7), sélection courbe 7 CAM_8_MX (8), sélection courbe 8 CAM_9_MX (9), sélection courbe 9 CAM_10_MX (10), sélection courbe 10
Numerator	DINT	Valeur du compteur pour la mise à l'échelle de la courbe
Denominator	UDINT	Valeur dénominateur pour la mise à l'échelle de la courbe
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	Paramètre modifié
Busy	BOOL	Traitement du bloc fonction en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	DWORD	Voir page 126



3.5.11 MC_CamCurveFlowSelect_MX



64423axx

Description

Ce bloc fonction permet de paramétrer un profil de courbe avec dix courbes maximum. Chaque courbe peut être affectée d'une transition.

Le profil est défini via l'entrée *CurveFlow1*.

Les paramètres sont transmis sous forme de tableau. Ce tableau peut être transmis via un bloc fonction d'initialisation.

Exemple :

```
FUNCTION_BLOCK InitCamFlow
```

```
VAR_INPUT
```

```
    Execute:BOOL; (* select the curve flo`*)
```

```
    par défaut : BOOL ;(*select all curve standalone*)
```

```
END_VAR
```

```
VAR_OUTPUT
```

```
    numberofsegments:UINT;
```

```
    CurveFlow1:MC_CAM_CURVE_FLOW_MX;
```

```
END_VAR
```

```
EXE:=Execute AND NOT bREdgeexecute;
```

```
bREdgeexecute:=Execute;
```

```
IF EXE THEN
```

```
    numberofsegments:=10;
```

```
    (*curve segment 1 first curve*)
```

```
    CurveFlow1.Camsegment[1].Curvenumber :=1;
```

```
    CurveFlow1.Camsegment[1].Nextcurve :=2;
```

```
    CurveFlow1.Camsegment[1].Transitionmaster :=5000;CurveFlow1.Camsegment[1].YSlaveoffsetadd :=5000;
```

```
    (*curve segment 2 *)
```

```
    CurveFlow1.Camsegment[2].Curvenumber :=2;
```

```
    CurveFlow1.Camsegment[2].Nextcurve :=3;
```

```
    CurveFlow1.Camsegment[2].Transitionmaster :=0;
```

```
    CurveFlow1.Camsegment[2].YSlaveoffsetadd :=0;
```

```
    (*curve segment 3 *)
```

```
    CurveFlow1.Camsegment[3].Curvenumber :=3;
```




```

CurveFlow1.Camsegment[3].Nextcurve :=4;
CurveFlow1.Camsegment[3].Transitionmaster :=10000;
CurveFlow1.Camsegment[3].YSlaveoffsetadd :=5000;
(*curve segment 4 *)
CurveFlow1.Camsegment[4].Curvenumber :=4;
CurveFlow1.Camsegment[4].Nextcurve :=5;
CurveFlow1.Camsegment[4].Transitionmaster :=10000;
CurveFlow1.Camsegment[4].YSlaveoffsetadd :=5000;
(*curve segment 5 *)
CurveFlow1.Camsegment[5].Curvenumber :=5;
CurveFlow1.Camsegment[5].Nextcurve :=6;CurveFlow1.Camsegment[5].Transi-
tionmaster :=10000;
CurveFlow1.Camsegment[5].YSlaveoffsetadd :=5000;
(*curve segment 6 *)
CurveFlow1.Camsegment[6].Curvenumber :=6;
CurveFlow1.Camsegment[6].Nextcurve :=7;
CurveFlow1.Camsegment[6].Transitionmaster :=10000;
CurveFlow1.Camsegment[6].YSlaveoffsetadd :=5000;
(*curve segment 7 *)
CurveFlow1.Camsegment[7].Curvenumber :=7;
CurveFlow1.Camsegment[7].Nextcurve :=8;
CurveFlow1.Camsegment[7].Transitionmaster :=10000;
CurveFlow1.Camsegment[7].YSlaveoffsetadd :=5000;
(*curve segment 8 *)
CurveFlow1.Camsegment[8].Curvenumber :=8;
CurveFlow1.Camsegment[8].Nextcurve :=9;
CurveFlow1.Camsegment[8].Transitionmaster :=10000;
CurveFlow1.Camsegment[8].YSlaveoffsetadd :=5000;
(*curve segment 9 *)
CurveFlow1.Camsegment[9].Curvenumber :=9;
CurveFlow1.Camsegment[9].Nextcurve :=10;
CurveFlow1.Camsegment[9].Transitionmaster :=10000;
CurveFlow1.Camsegment[9].YSlaveoffsetadd :=5000;
(*curve segment 10 last curve *)
CurveFlow1.Camsegment[10].Curvenumber :=10;
CurveFlow1.Camsegment[10].Nextcurve :=1;
CurveFlow1.Camsegment[10].Transitionmaster :=10000;
CurveFlow1.Camsegment[10].YSlaveoffsetadd :=5000;
END_IF

```



Voir type de donnée "MC_CAM_CURVE_FLOW_MX" et
"MC_CAM_CFLOW_MX"

Pour chaque segment, les grandeurs suivantes sont indiquées :

- le numéro de courbe
- le prochain numéro de courbe
- la trajectoire maître pour la transition
- la trajectoire esclave à parcourir pour la transition

Si la valeur 0 est indiquée pour *Transitionmaster* et *Yslaveoffsetadd*, automatiquement aucune transition n'est raccordée.

L'entrée *Transitionmaster* permet d'indiquer la trajectoire du maître (Delta X) et l'entrée *Yslaveoffsetadd* la distance à parcourir par l'esclave. Le déplacement s'effectue via LPG1.

Une commutation est possible avec la configuration avancée, voir entrée *ExtendedCamConfig.TecFunktionTransition*, type de donnée "MC_CAM_TRANSTYP_MX".

Les paramètres de dynamisme peuvent être modifiés via le bloc fonction "MC_SetCamTransition Dynamics_MX".

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

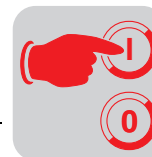
LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées binaires :

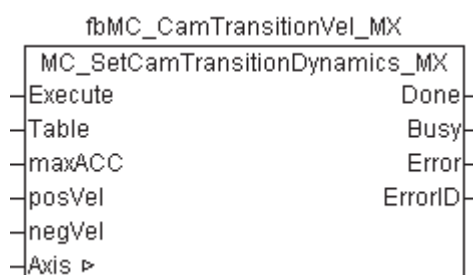
Nom	Type	Signification
Execute	BOOL	Cette entrée active le déplacement d'offset.
Number_of_segments	UINT	Nombre de courbes qui doivent être raccordées les unes aux autres
CurveFlow1	MC_CAM_CURVE_FLOW_MX (Camsegment: ARRAY[1..10] OF MC_CAM_CFLOW_MX;)	Sélection du profil de courbe Camsegment[1].Curvenumber, numéro de courbe Camsegment[1].Nextcurve, prochain numéro de courbe Camsegment[1].Transitionmaster, trajectoire maître de Tr. Camsegment[1].YSlaveoffsetadd, distance esclave de Tr Camsegment[10].Curvenumber Camsegment[10].Nextcurve Camsegment[10].Transitionmaster Camsegment[10].YSlaveoffsetadd
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	Paramètre modifié
Busy	BOOL	Traitement du bloc fonction en cours.
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	DWORD	Voir page 126



3.5.12 MC_SetCamTransitionDynamics_MX



64427axx

Description

Ce bloc fonction permet de modifier les paramètres de dynamisme d'une transition. Les transitions sont affectées aux courbes 1 à 10, c'est pourquoi elles peuvent également être sélectionnées via cette référence.

Voir *ExtDiag* de "MC_LinkTecCam_MX".

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

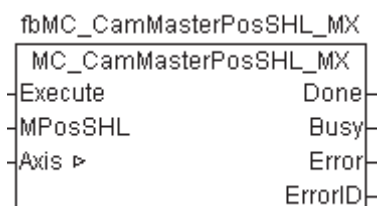
Nom	Type	Signification
Execute	BOOL	Cette entrée active le déplacement d'offset.
Table	MC_CAM_TABLENUMBER_MX	Sélection de la courbe Cam_Wait_MX (0), sélection des courbes d'attente CAM_1_MX (1), sélection courbe 1 CAM_2_MX (2), sélection courbe 2 CAM_3_MX (3), sélection courbe 3 CAM_4_MX (4), sélection courbe 4 CAM_5_MX (5), sélection courbe 5 CAM_6_MX (6), sélection courbe 6 CAM_7_MX (7), sélection courbe 7 CAM_8_MX (8), sélection courbe 8 CAM_9_MX (9), sélection courbe 9 CAM_10_MX (10), sélection courbe 10
maxACC	REAL	Accélération / décélération maximale pour LPG1/LPG2 par défaut = 0.000 000 001
posVel	REAL	Vitesse positive maximale pour LPG1 par défaut = 1.0
negVel	REAL	Vitesse positive maximale pour LPG1 par défaut = 1.0
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	Paramètre modifié
Busy	BOOL	Traitement du bloc fonction en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	DWORD	Voir page 126



3.5.13 MC_CamMasterPosSHL_MX



64429axx

Description

Ce bloc fonction permet d'augmenter la mise à l'échelle de la position maître pour obtenir une meilleure résolution. Ceci permet une meilleure interpolation de l'axe, par conséquent le fonctionnement en mode came électronique est plus silencieux.

Les longueurs de cycles maître des courbes sont également mises à une échelle supérieure par le facteur $2^{MPosSHL}$. Le facteur est automatiquement limité par référence au plus grand cycle maître des courbes. Affichage au niveau des variables internes *SEWTecCamMotion* [numéro d'axe].*ExtDiag.MasterCycle*.

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

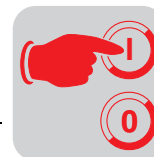
LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

Nom	Type	Signification
Execute	BOOL	Cette entrée active les modifications de paramètres.
MPosSHL	UINT	La position maître est décalée vers la gauche de cette valeur, c.-à-d $MPosSHL = 10$ $position\ maître \times 2^{10} =$ $position\ maître \times 1024$
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	Paramètre modifié
Busy	BOOL	Traitement du bloc fonction en cours.
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	DWORD	Voir page 126



3.5.14 MC_SetMasterCycle_MX

Description

Ce bloc fonction permet de modifier le cycle du maître.

Si la courbe sélectionnée est activée, elle est parcourue jusqu'à la fin avec l'ancien cycle maître ; le nouveau cycle maître est actif au prochain passage. Le cycle maître est automatiquement mis à une échelle supérieure par le facteur *MasterShiftfaktor*.

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

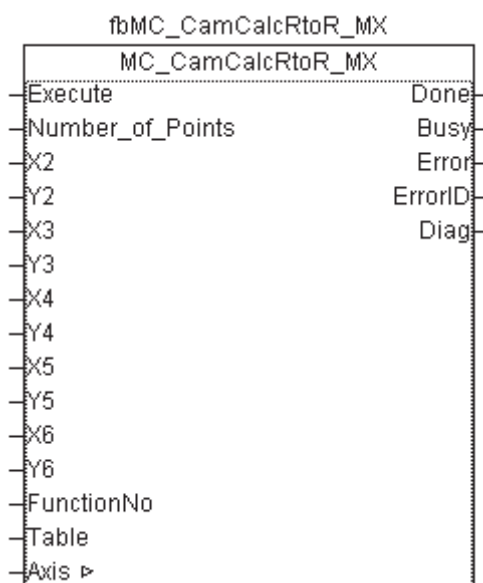
Nom	Type	Signification
Execute	BOOL	Cette entrée active les modifications de paramètres.
Table	MC_CAM_TABLENUMBER_MX	Sélection de la courbe Cam_Wait_MX (0), sélection de la courbe d'attente CAM_1_MX (1), sélection courbe 1 CAM_2_MX (2), sélection courbe 2 CAM_3_MX (3), sélection courbe 3 CAM_4_MX (4), sélection courbe 4 CAM_5_MX (5), sélection courbe 5 CAM_6_MX (6), sélection courbe 6 CAM_7_MX (7), sélection courbe 7 CAM_8_MX (8), sélection courbe 8 CAM_9_MX (9), sélection courbe 9 CAM_10_MX (10), sélection courbe 10
Mastercycle	DINT	Nouveau cycle maître de la courbe sélectionnée
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
Done	BOOL	Paramètre modifié
Busy	BOOL	Traitement du bloc fonction en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	DWORD	Voir page 126



3.5.15 MC_CamCalcRtoR_MX

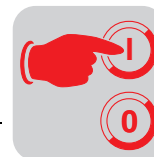


64430axx

Description

Calcul de courbes à partir de deux à six points de coordonnée. Le début de la courbe est toujours à 0 / 0 (X1 / Y1).

- Le calcul s'effectue toujours de manière imbriquée.
- Le cycle maître correspond à la plus grande position maître, p. ex. points = 5 → cycle maître = X5.
- Pour obtenir une meilleure mise à l'échelle, le cycle maître est mis à une échelle supérieure par le facteur décalage maître, p. ex. cycle maître = 10000 → facteur décalage maître = 6. Dans le MOVIAXIS® est donc écrit $10000 \times 2^6 = 64000$. Le facteur décalage maître peut être adapté à l'aide du bloc fonction "MC_CamMasterPosSHL_MX".
- La mise à l'échelle de la position est ajustée en fonction de la valeur Y la plus importante. Les coordonnées esclave (Y2-Y6) sont multipliées par le facteur $2^{\text{Shift-Factor}}$. Exemple : Y4 = 4096, facteur décalage = 5 → Y4_interne = $4096 \times 2^5 = 131072$. Le facteur de décalage est inséré dans le CamFlow1 de la courbe correspondante. Le facteur de décalage de l'esclave peut être limité via la variable *ExtConfig.Calc_SlaveShift_max*, (la valeur initiale est 10).
- La vitesse minimale et maximale sur la courbe est indiquée.
- L'accélération minimale et maximale est indiquée.
- La courbe est chargée dans le MOVIAXIS® en fonction du tableau choisi. Le bloc fonction intègre un tableau de 512 points de came pour la courbe correspondant à la configuration du MOVIAXIS®.



Entrées :

Nom	Type	Signification
Execute	BOOL	Cette entrée lance l'exécution de la commande du bloc fonction
Number_of_Points	UINT	Nombre de points sur la courbe
X2	DINT	2. Point de repère - position maître
Y2	DINT	2. Point de repère - position esclave
X3	DINT	3. Point de repère - position maître
Y3	DINT	3. Point de repère - position esclave
X4	DINT	4. Point de repère - position maître
Y4	DINT	4. Point de repère - position esclave
X5	DINT	5. Point de repère - position maître
Y5	DINT	5. Point de repère - position esclave
X6	DINT	6. Point de repère - position maître
Y6	DINT	6. Point de repère - position esclave
FunctionNo	MC_CAM_CALC_FUNCTIONNO_MX	0 = Straight line 1 = 5th order polynomial 2 = Sine 3 = 3rd order polynomial 4 = Sinoids from Bestehorn 5 = Modified acceleration trapezoid
Table	MC_CAM_TABLENUMBER_MX	Sélection de la courbe 1-10CAM_1_MX CAM_2_MX CAM_3_MX CAM_4_MX CAM_5_MX CAM_6_MX CAM_7_MX CAM_8_MX CAM_9_MX CAM_10_MX
Axis	AXIS_REF	Cette entrée permet de définir sur quel axe moteur les actions du bloc fonction doivent être exécutées.

Sorties :

Nom	Type	Signification
Done	BOOL	Calcul, chargement de la courbe et des paramètres achevés
Busy	BOOL	Traitement des blocs fonction pas encore achevé
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UDINT	Voir page 126
Suite du tableau sur la page suivante		



Nom		Type	Signification
DIAG	Vmax	DINT	"Vitesse" maximale $V(i)=Y(i+1)-Y(i)$ à l'intérieur de la came électronique
	Vmin	DINT	"Vitesse" minimale $V(i)=Y(i+1)-Y(i)$ à l'intérieur de la came électronique
	AbsVmax	DINT	Valeur maximale pour la "vitesse" $V(i)=Y(i+1)-Y(i)$ à l'intérieur de la came électronique
	Amax	DINT	"Accélération" maximale $A(i)=V(i)-V(i-1)$ à l'intérieur de la came électronique
	Amin	DINT	"Accélération" minimale $A(i)=V(i)-V(i-1)$ à l'intérieur de la came électronique
	AbsAmax	DINT	Valeur maximale pour l'"accélération" $A(i)=V(i)-V(i-1)$ à l'intérieur de la came électronique
	Mastercycle	DINT	Cycle du maître
	Shifffactor	DINT	Mise à l'échelle de l'esclave $\times 2^n$
	State	BYTE	Etat de traitement
	MasterShiftTab	DINT	Non affecté

Mode de fonctionnement du bloc fonction :

Le bloc fonction calcule une courbe à partir des coordonnées (X = maître / Y = esclave).

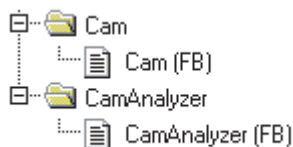
La courbe est composée de plusieurs tronçons : chaque tronçon se raccorde au précédent en un point où le moteur chargé d'exécuter la came est à l'arrêt. Les tronçons sont calculés à l'aide du bloc fonction "Cam". Le programme a besoin de plusieurs cycles de programme pour le calcul et le chargement (bibliothèque 11_System\MPLCTechFct_Cam.lib).

Les points du tableau sont sauvegardés sous forme d'array (tableau). Puis la vitesse et l'accélération minimales et maximales sont indiquées par le bloc fonction "CamAnalyser".

Le bloc fonction "MC_WriteParameter_MX" permet d'écrire les 512 points de came dans le MOVIAXIS®. Avec un temps de cycle de 3 ms, le bloc fonction a besoin d'env. 3,5 s. Si tout s'est déroulé correctement, le bit *Done* est mis à "True".

Blocs fonction utilisés en interne :

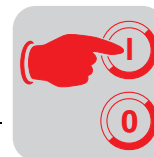
11_System\MPLCTechFct_Cam.lib



64431axx

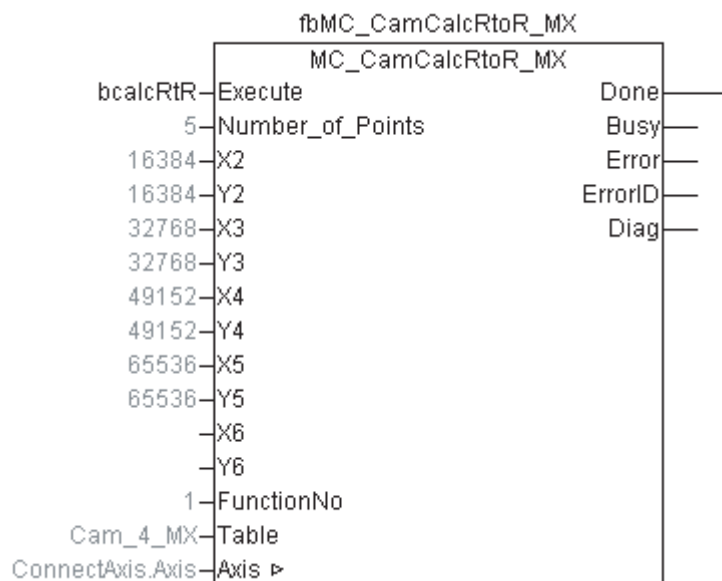
Attention :

- L'utilisateur devra s'assurer que la courbe calculée n'est pas sélectionnée ou que l'entraînement est à l'arrêt.
- Pour activer le tableau de courbes correspondant dans le MX, sélectionner un des blocs fonction suivants : "MC_CamTableSelect_MX", "MC_CamTableSelectTrans_MX", "MC_CamCurveFlowSelect_MX".
- L'éditeur de came électronique permet de charger le tableau depuis le MOVIAXIS®.



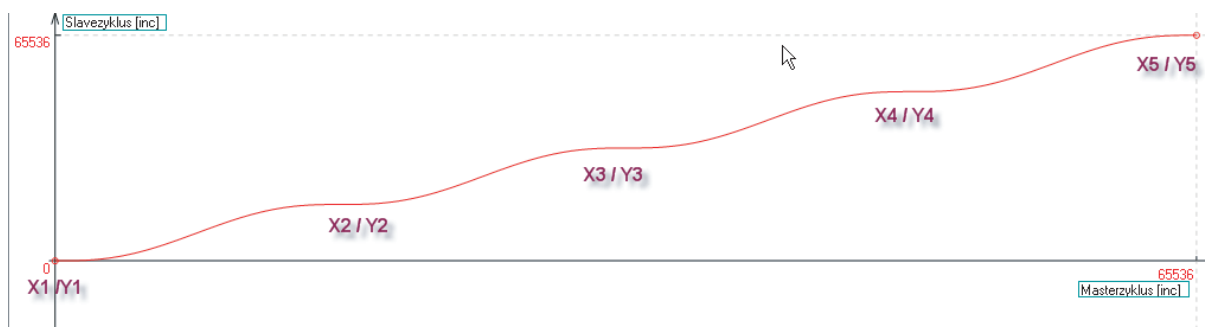
Exemple de courbe *Number_of_Points* = 5.

Appel du bloc fonction :



64432axx

Courbe résultant du tableau des points de came :



64433axx

Déroulement du déplacement :

L'entraînement se déplace d'une rotation jusqu'au point d'arrêt Y2 (Vm se déplace de X1 à X2).

L'entraînement se déplace d'une rotation jusqu'au point d'arrêt Y3 (Vm se déplace de X2 à X3). L'entraînement se déplace d'une rotation jusqu'au point d'arrêt Y4 (Vm se déplace de X3 à X4).

L'entraînement se déplace d'une rotation jusqu'au point d'arrêt Y5 (Vm se déplace de X4 à X5).

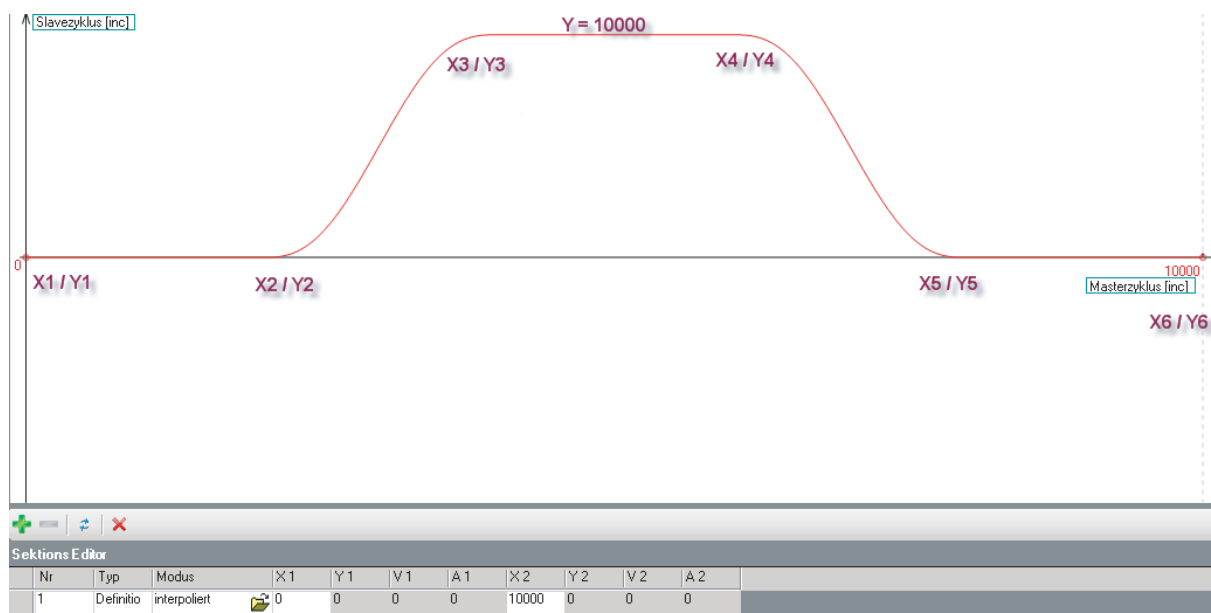
La courbe recommence au point X1.



Exemple de courbe *Points* = 6.

Appel du bloc fonction :

Courbe résultant du tableau des points de came :



64434axx

Déroulement du déplacement :

L'entraînement est à l'arrêt dans le segment 1 (le maître virtuel se déplace de X1 vers X2).

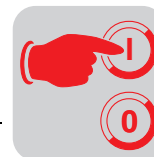
L'entraînement se déplace d'un tour (Vm se déplace de X2 vers X3).

L'entraînement est à l'arrêt dans le segment 3 (position Y3 = Y4/ Vm se déplace de X3 vers X4)

L'entraînement recule d'un tour (Vm se déplace de X4 vers X5).

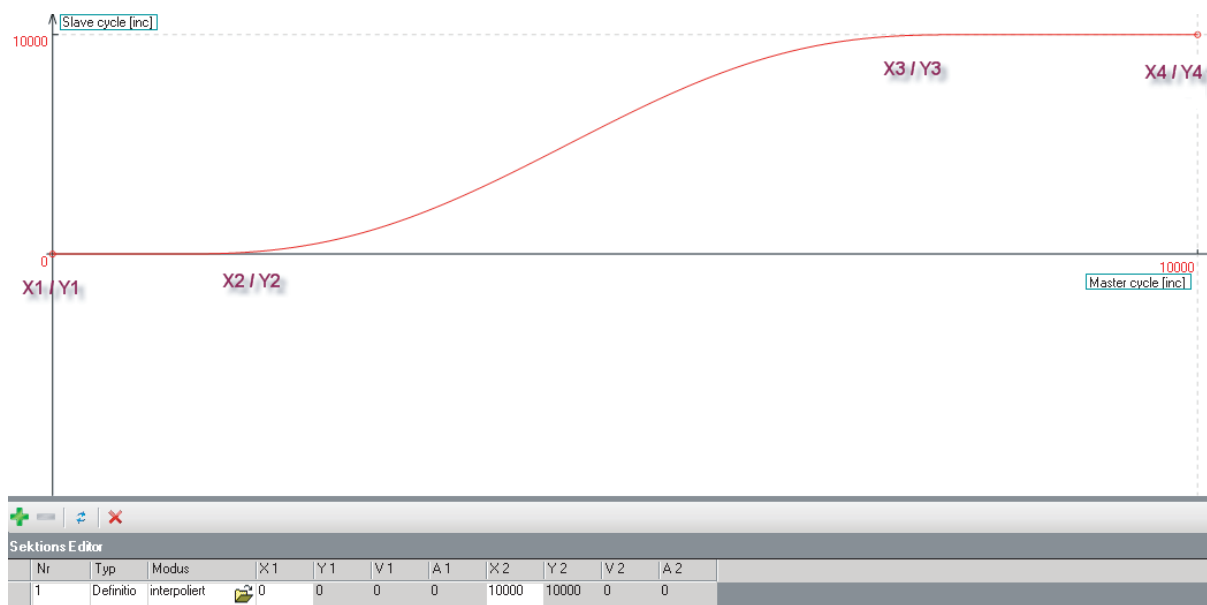
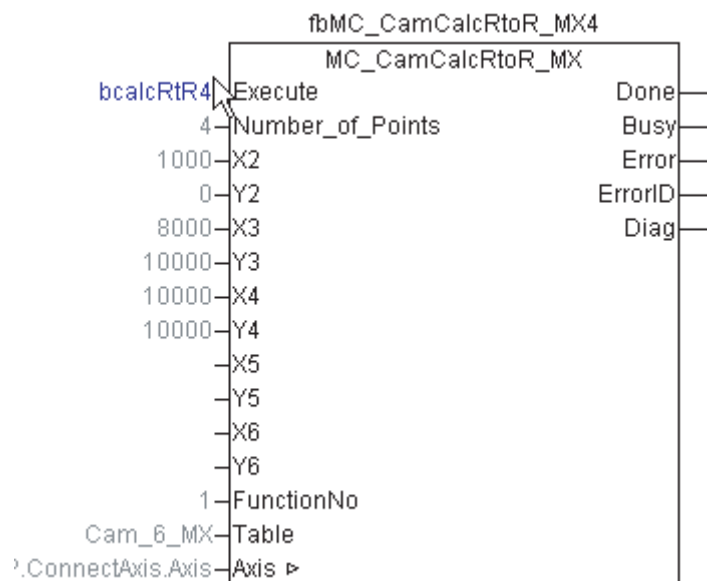
L'entraînement est à l'arrêt dans le segment 5. (Y6 = Y1/ Vm se déplace de X5 vers X6)

La courbe recommence aux points X1 / Y1.



Exemple de courbe *Points* = 4

Appel du bloc fonction :



64435axx

L'entraînement est à l'arrêt dans le segment 1 (le maître virtuel se déplace de X1 vers X2).

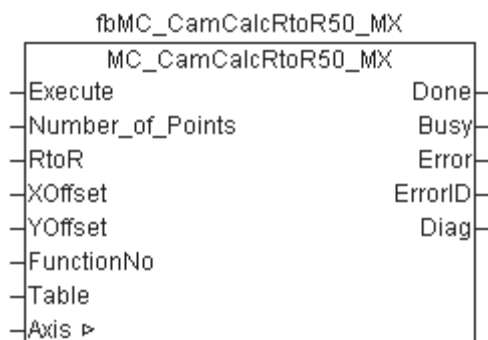
L'entraînement se déplace d'un tour en avant (Vm se déplace de X2 vers X3).

L'entraînement est à l'arrêt dans le segment 3 (position Y3 = Y4 / Vm se déplace de X3 vers X4).

La courbe recommence au point X1.



3.5.16 MC_CamCalcRtoR50_MX



64436axx

Description

Ce bloc fonction permet de paramétrer une courbe avec 50 points de came maximum. A chaque courbe peut être raccordée une transition.

L'entrée *RtoR* permet de transférer les coordonnées (X/Y) dans un Array (tableau) ; ce tableau peut être transféré avec un bloc d'initialisation.

Les valeurs X sont les valeurs maître, les valeurs Y les valeurs esclave.

Exemple :

```
FUNCTION_BLOCK InitRtR50
```

```
VAR_INPUT
```

```
END_VAR
```

```
VAR_OUTPUT
```

```
END_VAR
```

```
VAR
```

```
    bDone: BOOL ;
```

```
END_VAR
```

```
IF NOT bDone THEN
```

```
    gRtoR.Point[1].X:= 2048;
```

```
    gRtoR.Point[1].Y := 0 × 4096;
```

```
    gRtoR.Point[2].X:=2 × 2048;
```

```
    gRtoR.Point[2].Y := 1 × 4096;
```

```
    gRtoR.Point[3].X := 3 × 2048;
```

```
    gRtoR.Point[3].Y := 1 × 4096;
```

```
    gRtoR.Point[4].X := 4 × 2048;
```

```
    gRtoR.Point[4].Y := 2 × 4096;
```

```
    gRtoR.Point[5].X := 5 × 2048;
```

```
    gRtoR.Point[5].Y := 2 × 4096;
```

```
    gRtoR.Point[6].X := 6 × 2048;
```

```
    gRtoR.Point[6].Y := 3 × 4096;
```

```
    gRtoR.Point[7].X := 7 × 2048;
```

```
    gRtoR.Point[7].Y := 3 × 4096;
```



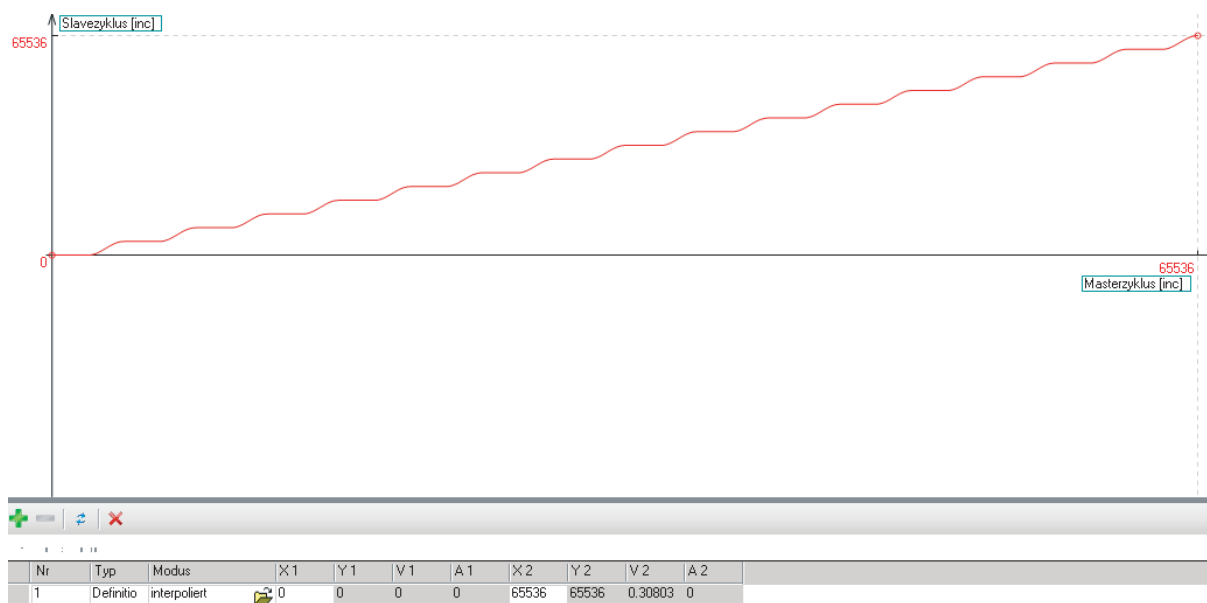
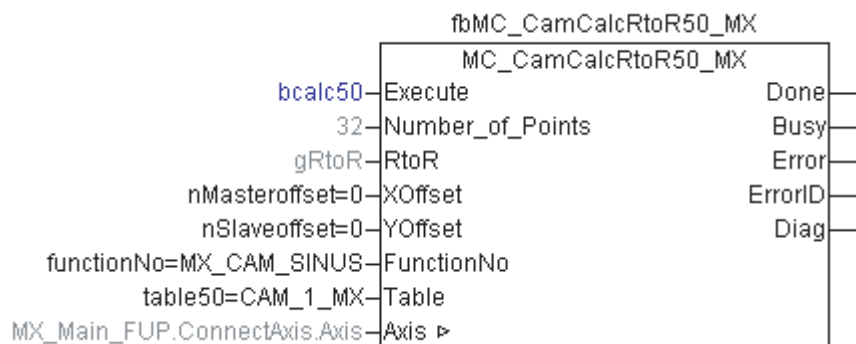
```
gRtoR.Point[8].X:= 8 × 2048;  
gRtoR.Point[8].Y := 4 × 4096;  
gRtoR.Point[9].X := 9 × 2048;  
gRtoR.Point[9].Y := 4 × 4096;  
gRtoR.Point[10].X := 10 × 2048;  
gRtoR.Point[10].Y := 5 × 4096;  
gRtoR.Point[11].X := 11 × 2048;  
gRtoR.Point[11].Y := 5 × 4096;  
gRtoR.Point[12].X := 12 × 2048;  
gRtoR.Point[12].Y := 6 × 4096;  
gRtoR.Point[13].X := 13 × 2048;  
gRtoR.Point[13].Y := 6 × 4096;  
gRtoR.Point[14].X := 14 × 2048;  
gRtoR.Point[14].Y := 7 × 4096;  
gRtoR.Point[15].X := 15 × 2048;  
gRtoR.Point[15].Y := 7 × 4096;  
gRtoR.Point[16].X := 16 × 2048;  
gRtoR.Point[16].Y := 8 × 4096;  
gRtoR.Point[17].X := 17 × 2048;  
gRtoR.Point[17].Y := 8 × 4096;  
gRtoR.Point[18].X := 18 × 2048;  
gRtoR.Point[18].Y := 9 × 4096;  
gRtoR.Point[19].X := 19 × 2048;  
gRtoR.Point[19].Y :=9 × 4096;  
gRtoR.Point[20].X := 20 × 2048;  
gRtoR.Point[20].Y := 10 × 4096;  
gRtoR.Point[21].X := 21 × 2048;  
gRtoR.Point[21].Y := 10 × 4096;  
gRtoR.Point[22].X := 22 × 2048;  
gRtoR.Point[22].Y := 11 × 4096;  
gRtoR.Point[23].X := 23 × 2048;  
gRtoR.Point[23].Y := 11 × 4096;  
gRtoR.Point[24].X:= 24 × 2048;  
gRtoR.Point[24].Y := 12 × 4096;  
gRtoR.Point[25].X := 25 × 2048;  
gRtoR.Point[25].Y := 12 × 4096;  
gRtoR.Point[26].X := 26 × 2048;  
gRtoR.Point[26].Y := 13 × 4096;  
gRtoR.Point[27].X := 27 × 2048;  
gRtoR.Point[27].Y := 13 × 4096;  
gRtoR.Point[28].X := 28 × 2048;
```



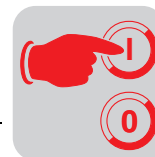
```

gRtoR.Point[28].Y := 14 × 4096;
gRtoR.Point[29].X := 29 × 2048;
gRtoR.Point[29].Y := 14 × 4096;
gRtoR.Point[30].X := 30 × 2048;
gRtoR.Point[30].Y := 15 × 4096;
gRtoR.Point[31].X := 31 × 2048;
gRtoR.Point[31].Y := 15 × 4096;
gRtoR.Point[32].X := 32 × 2048;
gRtoR.Point[32].Y := 16 × 4096;
bDone := TRUE;
END_IF

```



64437axx



Entrées :

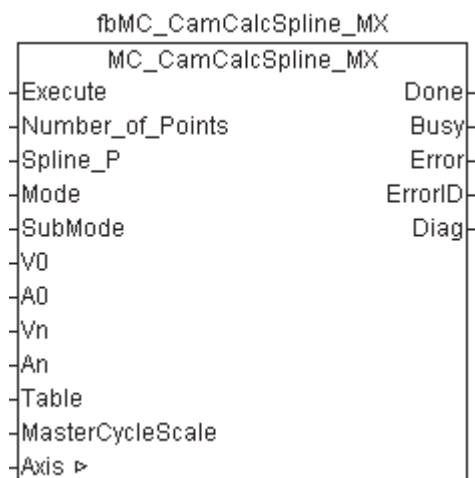
Nom	Type	Signification
Execute	BOOL	Cette entrée lance l'exécution de la commande du bloc fonction
Number_of_Points	UINT	Nombre de points sur la courbe
RtoR	MC_Cam_RtoR_MX (Point: ARRAY[1..50] OF MC_CAM_Point_MX) (X: DINT masterpoints Y: DINT slavepoints)	Les coordonnées (X/Y) sont transmises sous forme de tableau. La valeur X correspond au maître, la valeur Y à l'esclave.
XOffset	DINT	Décalage de la came électronique dans la direction du maître
YOffset	DINT	Décalage de la came électronique dans la direction de l'esclave
FunctionNo	MC_CAM_CALC_FUNC TIONNO_MX	0 = Straight line 1 = 5th order polynomial 2 = Sine 3 = 3rd order polynomial 4 = Sinoids from Bestehorn 5 = Modified acceleration trapezoid
Table	MC_CAM_TABLENUMBER_MX	Sélection de la courbe 1-10 CAM_1_MX CAM_2_MX CAM_3_MX CAM_4_MX CAM_5_MX CAM_6_MX CAM_7_MX CAM_8_MX CAM_9_MX CAM_10_MX
Axis	AXIS_REF	Cette entrée permet de définir sur quel axe moteur les actions du bloc fonction doivent être exécutées.

Sorties :

Nom	Type	Signification
Done	BOOL	Calcul, chargement de la courbe et des paramètres achevés
Busy	BOOL	Traitement des blocs fonction pas encore achevé
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UDINT	Voir page 126
DIAG	Vmax	"Vitesse" maximale $V(i)=Y(i+1)-Y(i)$ à l'intérieur de la came électronique
	Vmin	"Vitesse" minimale $V(i)=Y(i+1)-Y(i)$ à l'intérieur de la came électronique
	AbsVmax	Valeur maximale pour la "vitesse" $V(i)=Y(i+1)-Y(i)$ à l'intérieur de la came électronique
	Amax	"Accélération" maximale $A(i)=V(i)-V(i-1)$ à l'intérieur de la came électronique
	Amin	"Accélération" minimale $A(i)=V(i)-V(i-1)$ à l'intérieur de la came électronique
	AbsAmax	Valeur maximale pour l'"accélération" $A(i)=V(i)-V(i-1)$ à l'intérieur de la came électronique
	Mastercycle	Cycle du maître
	Shiftfactor	Mise à l'échelle de l'esclave $\times 2^n$
	State	Etat de traitement
	MasterShiftTab	Point de came du tableau à partir duquel la courbe a été décalée



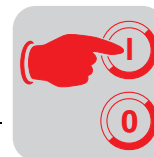
3.5.17 MC_CamCalcSpline_MX



64445axx

Entrées :

Nom	Type	Signification
Execute	BOOL	Cette entrée lance l'exécution de la commande du bloc fonction
Number_of_Points	UINT	Nombre de points sur la courbe
Spline_P	ARRAY[0..29] OF Spline_Point (X: INT masterpoints Y: DINT slavepoints)	Les coordonnées (X/Y) sont transmises sous forme de tableau. La valeur X correspond au maître, le numéro de point de coordonnée (1-512) est indiqué. La valeur Y correspond à la position de l'esclave.
Mode	MC_CAM_CALC_SPLI NEMODE_MX	MX_CAM_Spline_0 = Spline-Interpolation MX_CAM_Spline_B = Approximation MX_CAM_Spline_1 = fast, piecewise Interpolation
SubMode	UINT	Mode Control n'est utilisé que si Mode MX_CAM_Spline_B Submode 0 - Polynome 3 Submode 1- not used Submode 2 - Polynome 2 Submode 3 - Polynome 3 Submode 4 - Polynome 4 Submode 5 - Polynome 5 Submode 6 - Polynome 6 Submode 7 - Polynome 7
V0	REAL	Vitesse au point de départ
A0	REAL	Accélération au point de départ
Vn	REAL	Vitesse au point final
An	REAL	Accélération au point final
Table	MC_CAM_TABLENUM BER_MX	Sélection de la courbe 1-10 CAM_1_MX CAM_2_MX CAM_3_MX CAM_4_MX CAM_5_MX CAM_6_MX CAM_7_MX CAM_8_MX CAM_9_MX CAM_10_MX
MasterCycleScale	DINT	Cycle maître de la courbe
Axis	AXIS_REF	Cette entrée permet de définir sur quel axe moteur les actions du bloc fonction doivent être exécutées



Sorties :

Nom		Type	Signification
Done		BOOL	Calcul, chargement de la courbe et des paramètres achevés
Busy		BOOL	Traitement des blocs fonction pas encore achevé
Error		BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID		UDINT	Voir page 126
DIAG	Vmax	DINT	"Vitesse" maximale $V(i)=Y(i+1)-Y(i)$ à l'intérieur de la came électronique
	Vmin	DINT	"Vitesse" minimale $V(i)=Y(i+1)-Y(i)$ à l'intérieur de la came électronique
	AbsVmax	DINT	Valeur maximale pour la "vitesse" $V(i)=Y(i+1)-Y(i)$ à l'intérieur de la came électronique
	Amax	DINT	"Accélération" maximale $A(i)=V(i)-V(i-1)$ à l'intérieur de la came électronique
	Amin	DINT	"Accélération" minimale $A(i)=V(i)-V(i-1)$ à l'intérieur de la came électronique
	AbsAmax	DINT	Valeur maximale pour l'"accélération" $A(i)=V(i)-V(i-1)$ à l'intérieur de la came électronique
	Mastercycle	DINT	Cycle du maître
	Shiftfactor	DINT	Mise à l'échelle de l'esclave $\times 2^n$
	State	BYTE	Etat de traitement
	MasterShiftTab	DINT	Non affecté

Exemple :

SplinePoints: ARRAY [0..29] OF SPLINE_POINT :=

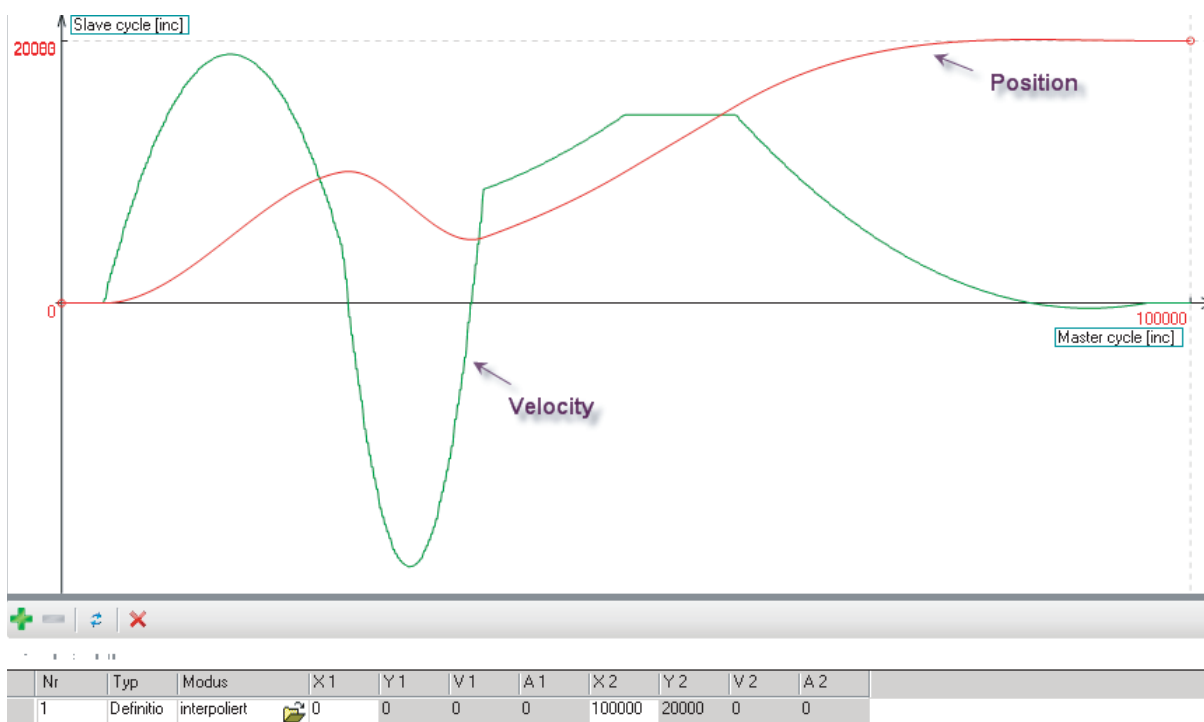
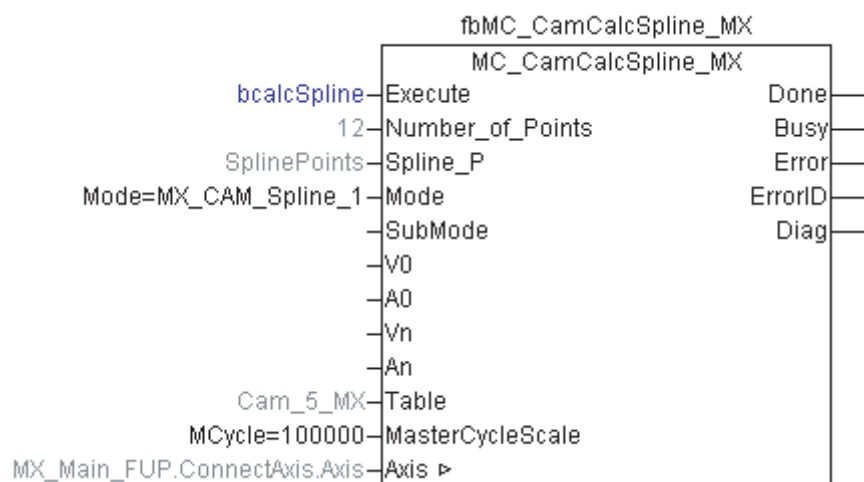
(x:=1, y:=0), (x:=10, y:=0), (x:=20, y:=0), (x:=128, y:=10000), (x:=192, y:=5000), (x:=256, y:=10000), (x:=266, y:=11000), (x:=286, y:=13000), (x:=306, y:=15000), (x:=492, y:=20000), (x:=502, y:=20000), (x:=512, y:=20000);

X correspond au numéro de point de came du tableau maître (1 - 512)

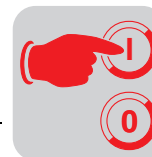
Y correspond à la position de l'esclave

Mode : MC_CAM_CALC_SPLINEMODE_MX:=2;

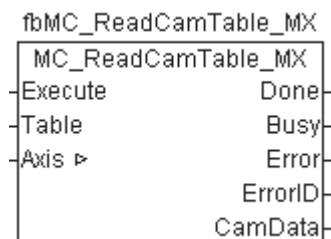
MCycle: DINT:=100000;



64446axx



3.5.18 MC_ReadCamTable_MX



64447axx

Description

Ce bloc fonction permet de lire un tableau de points de came à partir du MOVIAXIS®. Les points de came sont sauvegardés sous forme de tableau.

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".
LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

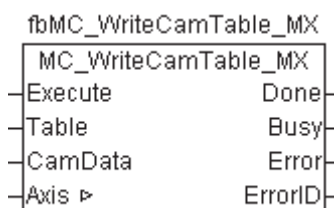
Nom	Type	Signification
Execute	BOOL	Cette entrée active la lecture des points de came.
Table	MC_CAM_TABLENUMBER_MX	Sélection de la courbe limitée aux valeurs 1-10 CAM_1_MX (1), sélection courbe 1 CAM_2_MX (2), sélection courbe 2 CAM_3_MX (3), sélection courbe 3 CAM_4_MX (4), sélection courbe 4 CAM_5_MX (5), sélection courbe 5 CAM_6_MX (6), sélection courbe 6 CAM_7_MX (7), sélection courbe 7 CAM_8_MX (8), sélection courbe 8 CAM_9_MX (9), sélection courbe 9 CAM_10_MX (10), sélection courbe 10
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
Done	BOOL	Paramètre lu
Busy	BOOL	Traitement du bloc fonction en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	DWORD	Voir page 126
CamData	ARRAY[0..511]	Sauvegarde des données pour les 512 points de came



3.5.19 MC_WriteCamTable_MX



64448axx

Description

Ce bloc fonction permet d'écrire un tableau de points de came dans le MOVIAXIS®.
Les points de came sont transmis sous forme de tableau.

Condition :

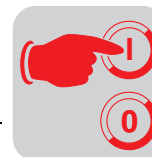
"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".
LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

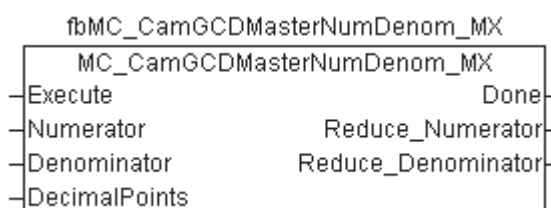
Nom	Type	Signification
Execute	BOOL	Cette entrée active l'écriture des points de came.
Table	MC_CAM_TABLENUMBER_MX	Sélection de la courbe limitée aux valeurs 1-10 CAM_1_MX (1), sélection courbe 1 CAM_2_MX (2), sélection courbe 2 CAM_3_MX (3), sélection courbe 3 CAM_4_MX (4), sélection courbe 4 CAM_5_MX (5), sélection courbe 5 CAM_6_MX (6), sélection courbe 6 CAM_7_MX (7), sélection courbe 7 CAM_8_MX (8), sélection courbe 8 CAM_9_MX (9), sélection courbe 9 CAM_10_MX(10), sélection courbe 10
CamData	ARRAY[0..511]	Mémoire des 512 points de came
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer.

Sorties :

Nom	Type	Signification
Done	BOOL	Paramètre modifié
Busy	BOOL	Traitement du bloc fonction en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	DWORD	Voir page 126



3.5.20 MC_CamGCDMasterNumDenom_MX



64449axx

Description

Ce bloc fonction permet de calculer les numérateur et dénominateur réduits pour la position maître. Les paramètres ne sont pas modifiés.

Les valeurs de numérateur et de dénominateur peuvent être utilisées pour le bloc fonction "MC_SetCamConfig_MX".

GCD -> **g**reatest **c**ommon **d**ivisor (plus grand diviseur commun)

	Exemple 1	Exemple 2
Numerator	65536	65536
Denominator	10000	20
DecimalPoints	0	3
Result: Reduce_Numerator	4096	2048
Result: Reduce_Denominator	625	625

Condition : aucune

Entrées :

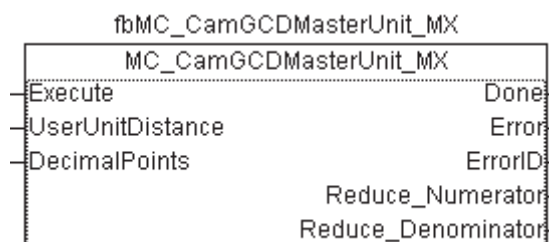
Nom	Type	Signification
Execute	BOOL	Active le calcul
Numerator	UDINT	Valeur numérateur du maître
Denominator	UDINT	Valeur dénominateur du maître
DecimalPoints	BYTE	Nombre de chiffres après la virgule

Sorties :

Nom	Type	Signification
Done	BOOL	Paramètre écrit
Reduce_Numerator	UDINT	Valeur numérateur réduite
Reduce_Denominator	UDINT	Valeur dénominateur réduite



3.5.21 MC_CamGCDMasterUnit_MX



64457axx

Description

Ce bloc fonction permet de calculer le numérateur et le dénominateur pour la position maître. Aucun paramètre n'est modifié.

Les valeurs de numérateur et de dénominateur peuvent être utilisées pour le bloc fonction "MC_SetCamConfig_MX".

GCD: **g**reatest **c**ommon **d**ivisor (plus grand diviseur commun).

Calcul :

MasterPosition [inc](65536inc) = MasterPosition [user unit] $\times 10^{\text{DecimalPoints}}$ $\times \text{Numerator} / \text{Denominator}$

	Exemple 1	Exemple 2
UserUnitDistance	12.345	1.000.000 [Unit]
DecimalPoints	3	0
Result: Reduce_Numerator	65536	1024
Result: Reduce_Denominator	12345	15625

Condition :

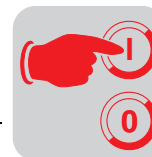
UserDistance supérieur à zéro.

Entrées :

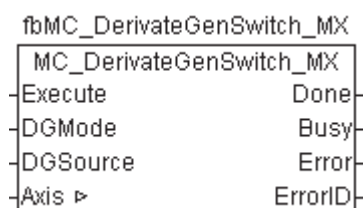
Nom	Type	Signification
Execute	BOOL	Active le calcul
UserUnitDistance	REAL	Longueur du cycle maître en unités utilisateur
DecimalPoints	BYTE	Nombre de chiffres après la virgule

Sorties :

Nom	Type	Signification
Done	BOOL	Paramètre écrit
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	DWORD	Voir page 126
Reduce_Numerator	UDINT	Valeur numérateur réduite
Reduce_Denominator	UDINT	Valeur dénominateur réduite



3.5.22 MC_DerivateGenSwitch_MX



64458axx

Description

Ce bloc fonction permet de commuter les paramètres "Adresse source" et "Mode d'exploitation" du générateur de dérive.

Le bloc fonction "MC_LinkTeCam_MX" règle la source sur la sortie de la came électronique 1 et le mode d'exploitation sur Position.

La bibliothèque utilise le CamFlow 1 pour les dix courbes et le CamFlow 2 pour le traitement d'offset. Le CamFlow 3 est libéré pour les applications spéciales.

Condition :

"MC_LinkTecCam_MX" a configuré la fonction technologique "Cam".

LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

Nom	Type	Signification
Execute	BOOL	Cette entrée active l'écriture des paramètres.
DGMode	DINT	Commutation mode d'exploitation 0 = None 1 = Position 2 = Vitesse 3 = Moment
DGSource	UINT	Source pour le générateur de dérivation (Derivategenerator) 1 = profil de courbe 1 (CamFlow 1)-> par défaut 2 = profil de courbe 2 (CamFlow 2) 3 = profil de courbe 3 (CamFlow 3)
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	Paramètre écrit
Busy	BOOL	Traitement du bloc fonction en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	DWORD	Voir page 126

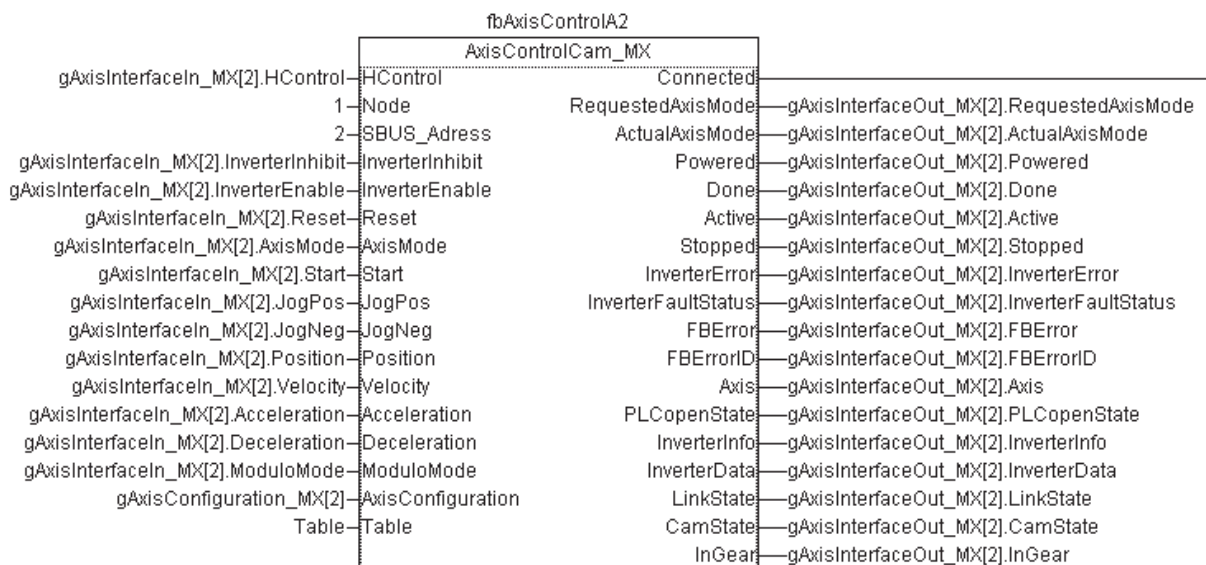


3.6 Exemple

3.6.1 Came électronique avec codeur virtuel maître

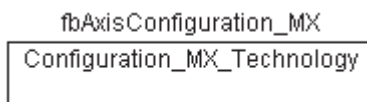
Création d'un nouveau projet PLC avec l'exemple "AxisControl_MX_Technologie".

Dans le bloc fonction "PRG_Technology_MX", supprimer les blocs "AxisControl_MX" et "AxisControlGear_MX".



64459axx

Régler les paramètres d'initialisation dans le bloc fonction "Configuration_MX_Technologie".



64460axx

Réglages de configuration dans le bloc fonction Configuration_MX_Technology

Codeur virtuel

IF NOT bInit THEN

```
(
*****
  Initialisation of AxisControlVirtual
*****
)

(* Configuration virtual encoder *)
gVirtualEncoderConfiguration.StartRisingEdge := FALSE;
gVirtualEncoderConfiguration.LinkTecAxisVirtual.CanNode := SBUS_NODE_1;
(* SBUS_NODE_1 SBUS_NODE_2 *)
gVirtualEncoderConfiguration.LinkTecAxisVirtual.SendObjectID:=  MX_VIRTUAL_
ENCODER_ID1;
gVirtualEncoderConfiguration.EncoderModeVirtual := VIRTUAL_MX_MODE;
(
*****
  Initialisation of AxisControl axis 2 (single axis)
*****
)
```




```
(* general axis control parameters *)
gAxisConfiguration_MX[2].StartRisingEdge := FALSE;
(* configuration modulo parameter --> only necessary if modulo mode is used *)
gAxisConfiguration_MX[2].SetModuloParameters.Overflow:= 1;
gAxisConfiguration_MX[2].SetModuloParameters.Underflow:= 1;
gAxisConfiguration_MX[2]. Adjustmode:= MX_MOD_ABS_SHORT_ XMASTER-
POSZERO;

(* Configuration cam parameters *)
(* the mastersignal is 65536 Inc/one revolution
if you want to have 1.000.000 Unit/revolution use the follow parameters
1.000.000/65536 = 62500/4096
1.000.000 UserUnit = 65536 Inc * 62500/4096*)
gAxisConfiguration_MX[2].SetCamConfig.MasterNumerator:= 62500;
(* Numerator for scaling of the master increments *)
gAxisConfiguration_MX[2].SetCamConfig.MasterDenominator:= 4096;
(* Denominator for scaling of the master increments *)
gAxisConfiguration_MX[2].SetCamConfig.MasterModuloMax:= 1.000.000;
(* Modulo maximum Value of the master position*)
gAxisConfiguration_MX[2].SetCamConfig.MasterModuloMin:= 0;
(* Modulo minimum Value of the master position*)
gAxisConfiguration_MX[2].SetCamConfig.MasterSource := MX_CAM_RECEIVE_
PDO_1;
(* select physical master or receive message box master is the SBus receive
objekt PDO1 *)
gAxisConfiguration_MX[2].SetCamConfig.LagErrorWindow := 1000;
(* lag error window in [inc] *)
gAxisConfiguration_MX[2].SetCamConfig.RotationDirectionLock :=0;
(* 0 = both directions, 1 = only CW enabled, 2 = only CCW enabled *)
gAxisConfiguration_MX[2].SetCamConfig.ExtConfig.ReactionLagEr-
ror:=MX_CAM_INHIBIT_WAITING;
gAxisConfiguration_MX[2].SetCamConfig.ExtConfig.MSignal_InterpolationTime :=
10;
(* Interpolation time: range 1...60 means 0.5....30ms filter time*)
gAxisConfiguration_MX[2].SetCamConfig.ExtConfig.AverageFilterTime := 10;
(* AverageFilter: range 1...60 means 0.5....30ms filter time*)
gAxisConfiguration_MX[2].SelectUserFcb.FCB_Number:= 26;
(* Designated FCB number *)
gAxisConfiguration_MX[2].SelectUserFcb.FCB_Instance:= 0;
(* Designated FCB instance number *)
bInit:=TRUE;
END_IF;
```

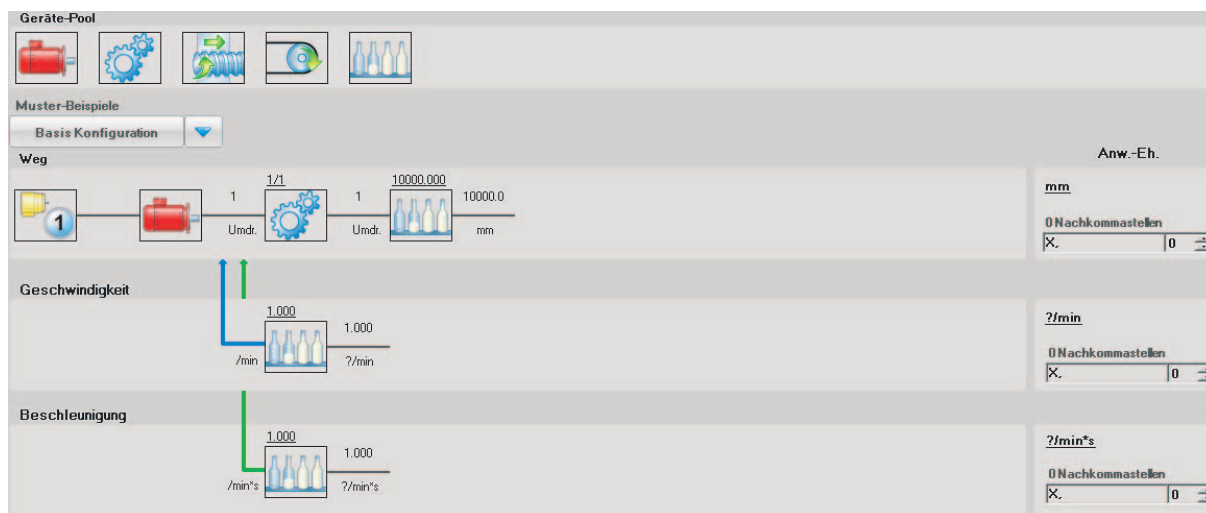


Description

Régler l'adresse SBus 2 sur le module de puissance.

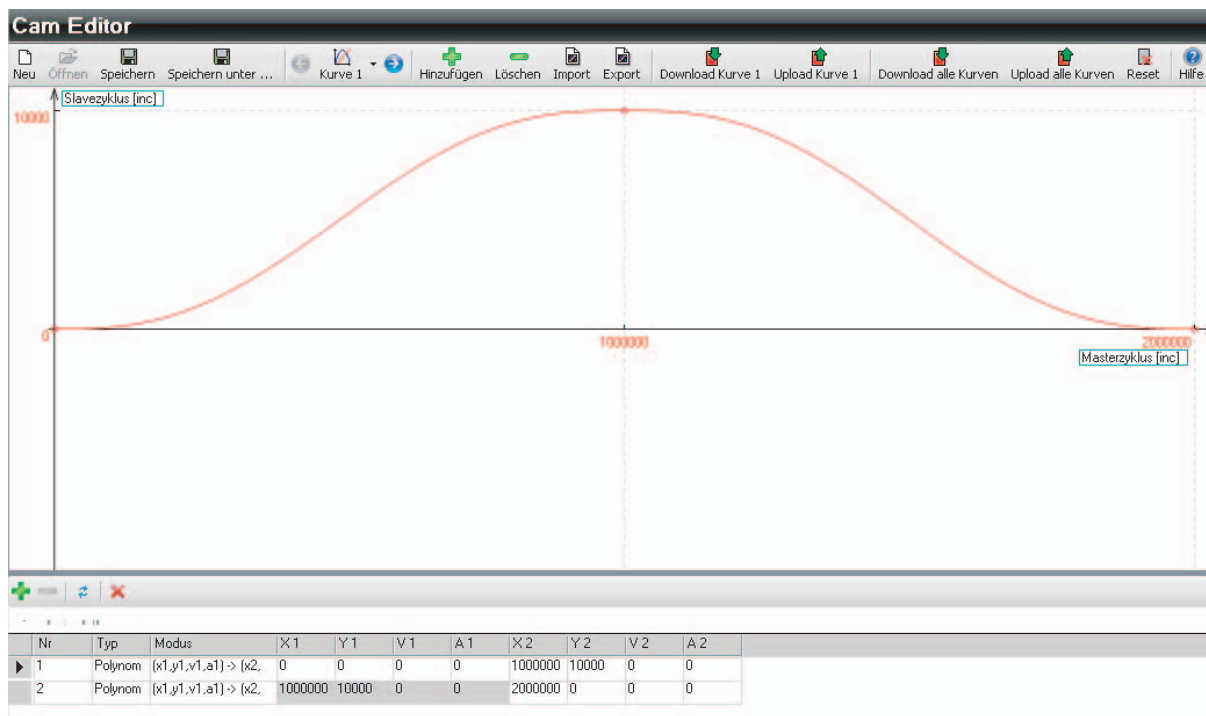
Effectuer un Drive Startup pour MOVI-PLC®. Procéder éventuellement aux réglages de base au préalable.

Un tour moteur doit correspondre à 10000 mm.



64471axx

A l'aide de l'éditeur de came électronique, créer la courbe 1 et la charger dans le variateur.

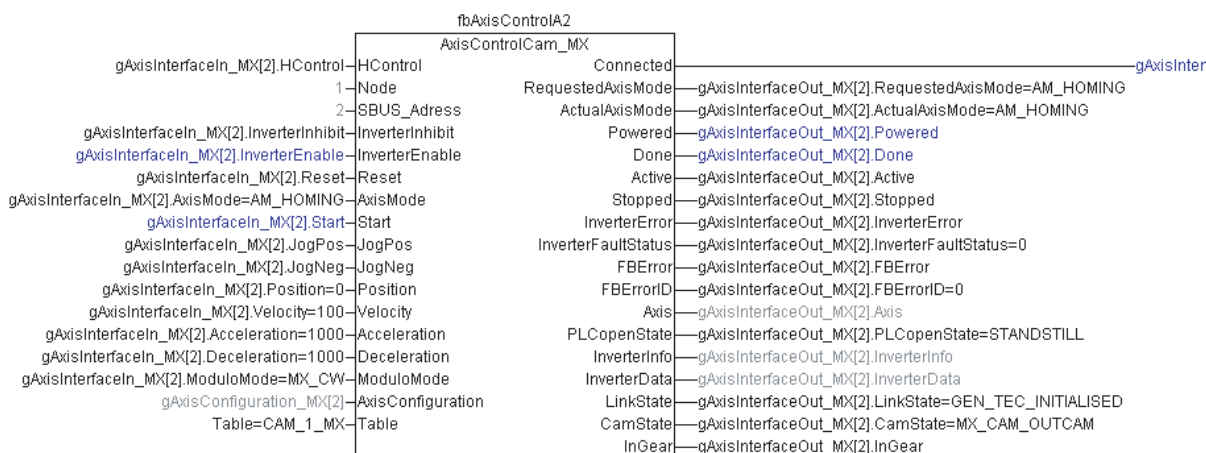
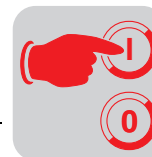


64472axx

Démarrer le programme PLC.

Libérer l'axe via l'entrée DI 0 et mettre à "1" le bit *InverterEnable*.

Sélectionner "AxisMode AM_Homing", mettre à "1" le bit *Start* et référencer l'axe.

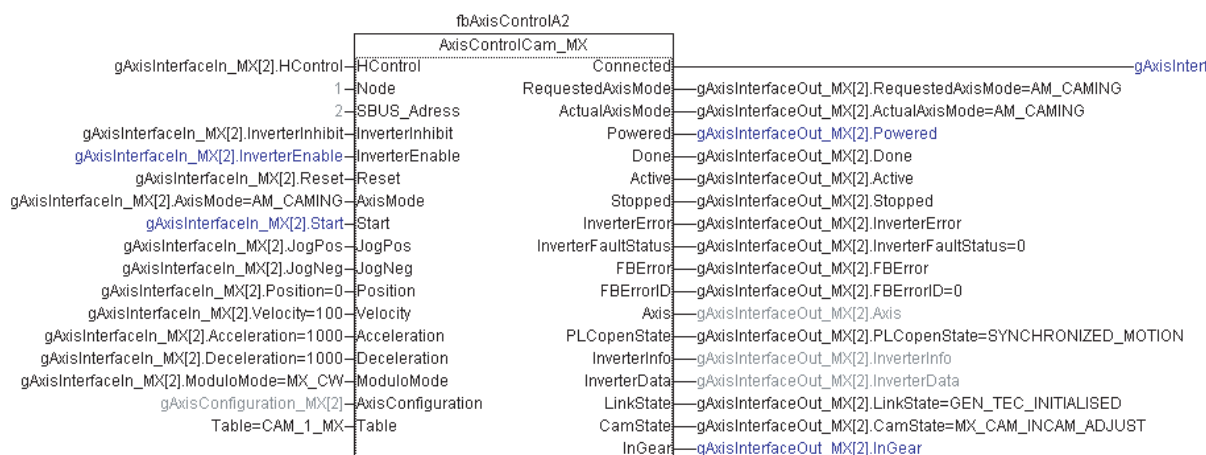


64473axx

Indiquer la vitesse et l'accélération.

Sélectionner "AxisMode AM_Camimg" et commuter l'axe en mode came électronique.

L'axe se cale via le bloc fonction "MC_CamInAdjust_MX" et se synchronise.

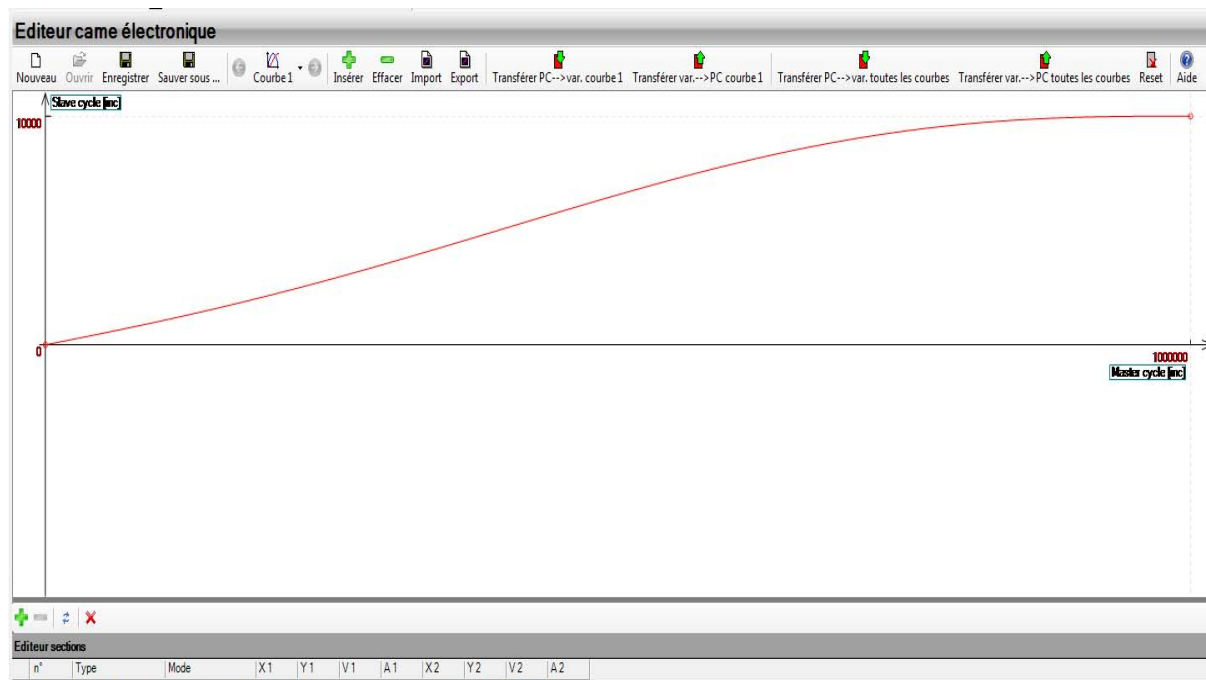


64474axx

Dans le bloc fonction "PRG_VirtualEncoder", démarrer le maître virtuel via Axismode "Velocity".

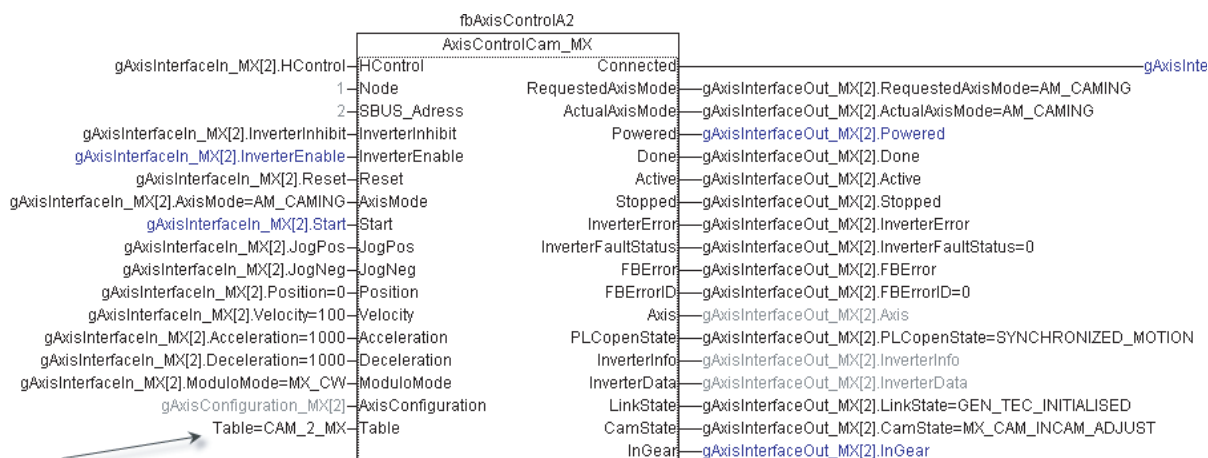
L'axe effectue une rotation à droite puis une rotation à gauche.

A l'aide de l'éditeur de came électronique, créer la "Courbe 2" puis la charger dans le variateur.



64475axx

Régler l'entrée *Table* du bloc fonction "AxisControlCam_MX sur Cam_2_MX".



64476axx

L'axe passe de la courbe 1 à la courbe 2. L'axe se déplace d'une rotation vers la droite à chaque rotation du maître.

Désactiver l'entrée *DIO* ou *InverterEnable*.

- L'axe s'arrête
- Le maître virtuel poursuit sa course.



Mettre l'entrée *Start* du maître virtuel sur FALSE.

- Le maître virtuel s'arrête.

Activer l'entrée *DIO* ou *InverterEnable*.

- L'axe se déplace vers la position esclave, correspondant à la position maître, par le chemin le plus court.

Mettre l'entrée *Start* du maître virtuel sur TRUE.

- L'axe suit le maître selon la courbe sélectionnée.



3.7 Annexes

3.7.1 Détection des défauts (ErrorID)

Code défaut	Nom du défaut	Description du défaut
FA0010	E_IEC_GENERAL_INVALID_TECHNOLOGIE_OPTION	La fonction technologique demandée n'est pas supportée
FA020...	General Link Tec Error	
FA0200	E_TEC_GENERAL_MULTIPLE_TECLINKS	L'appel du bloc fonction LinkTec plusieurs fois n'est pas autorisé
FA0201	E_TEC_GENERAL_INVALID_LINKSTATE	LinkState non admissible pour appel de cette fonction
FA0202	E_TEC_GENERAL_NOT_LINKED	Le bloc fonction LinkTec n'a pas encore de liaison logique
FA0203	E_TEC_GENERAL_NOT_INITIALIZED	Fonction technologique pas encore activée
FA0204¹⁾	E_TEC_GENERAL_SERVICE_NOT_IMPLEMENTED	La fonction demandée n'est pas supportée
FB02002	E_TEC_CAM_MX_INVALID_CAM_TEC_EDITOR_CONFIGURATION	Pas de réglages valides de l'éditeur de technologie
FA023	Cam Technologyfunction Error	
FA0233	E_TEC_CAM_MX_CURVE_AKTIV	Non autorisé lorsqu'une courbe est activée
FA0234	E_TEC_CAM_MX_SLAVESHIFT_TO_LOW	Le facteur de décalage calculé est inférieur au facteur de décalage maximal.
	Messages de défaut CEI généraux	
FA0070	E_IEC_PARAMETER_VALUE_OUT_OF_RANGE	Valeur de paramètre saisie non autorisée
FB0071	E_MDX_MOTIONBLOCK_LOG_ADR_NOT_INITIALIZED	MC_ConnectAxis_MDX n'a pas encore attribué d'adresse log.
FB0072	E_MDX_MOTIONBLOCK_INVALID_LOG_ADR	Adresse log non valide
FB0073	E_MDX_MOTIONBLOCK_INVALID_STATE	Appel de fonction depuis PLCopenState actuel non autorisé
FB0074¹⁾	E_MDX_MOTIONBLOCK_INVALID_OPERATING_MODE	La fonction technologique demandée n'est pas supportée par le mode d'exploitation MOVIDRIVE® réglé.

1) Ce message de défaut n'est pas pertinent avec un MOVIAXIS® (uniquement avec un MOVIDRIVE®)

3.7.2 Interruption des blocs fonction (CommandAborted)

La sortie "CommandAborted" est utilisée pour signaler les appels multiples d'un bloc fonction de déplacement ou l'appel d'un autre bloc fonction de déplacement. La tâche du bloc fonction interrompu ne sera pas menée à terme (pour des informations détaillées à ce sujet, voir le manuel "Bibliothèque MPLCMotion_MDX / MX pour MOVI-PLC®").

Le tableau suivant montre l'interruption côté opposé des blocs fonction "Motion".

Bloc fonction activé	provoque CommandAborted sur :
MC_CamInDirect_MX	- Instances de MC_CamInDirect_MX- 'Move-FB'
MC_CamInAdjust_MX	- Instances de MC_CamInAdjust_MX -"Move-FB"
MC_CamOffsetDirect_MX	- Instances de MC_CamOffsetDirect_MX -"Move-FB"
'Move-FB'	- MC_CamInDirect_MX - MC_CamInAdjust_MX - MC_CamOffsetDirect_MX

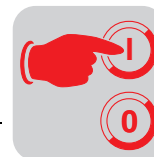
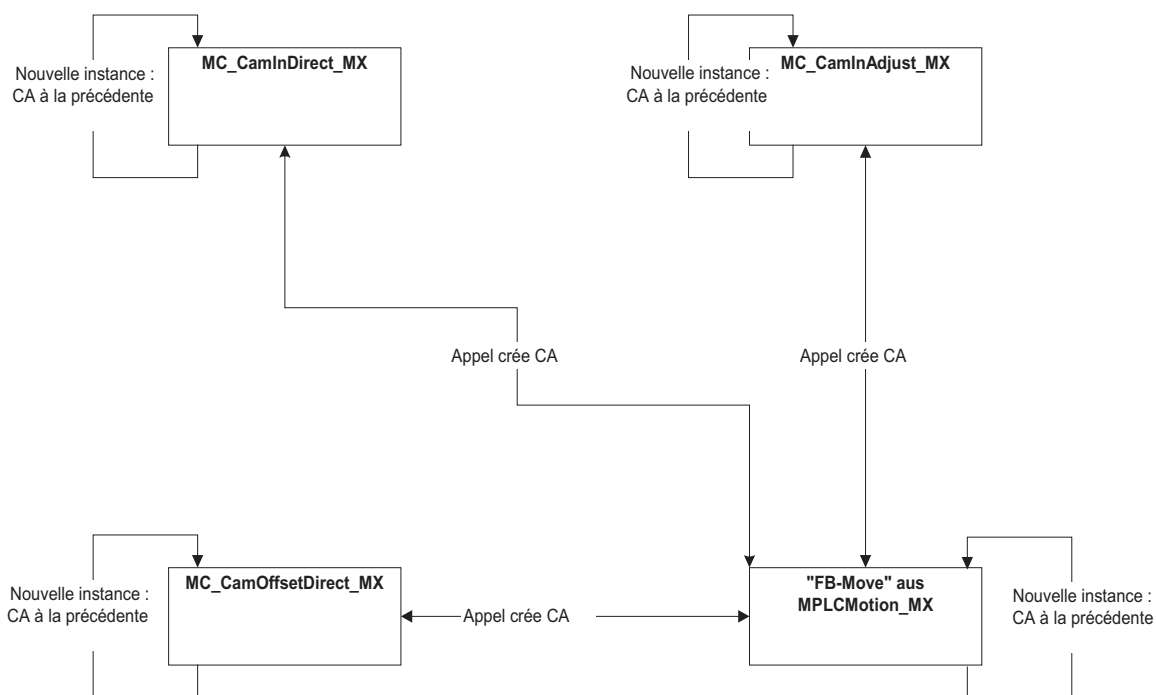


Schéma de "CommandAborted" :



64477axx

Remarque :

Les blocs fonction "Move-FB" de la bibliothèque MPLCMotion_MX sont les suivants :

- MC_MoveAbsolute_MX
- MC_MoveRelative_MX
- MC_MoveTargetPosition_MX
- MC_MoveTargetSpeed_MX
- etc.



4 MPLCTecCAMSwitch_MX

4.1 Introduction

Les servovariateurs multi-axes MOVIAXIS® offrent des fonctions technologiques très complètes, telles par exemple :

- Réducteur électronique,
- Cames électroniques,
- Capteurs d'événements,
- Gestion d'événements,
- Boîte à cames.

L'assistant de mise en route "Motion Technologie-Editor" de MotionStudio permet de configurer toutes les fonctions technologiques. Il est alors possible de procéder à un réglage de base. Mais il est fréquent que certaines valeurs ou réglages nécessitent une modification en cours de process, ou diverses fonctions devront être combinées et se dérouler de manière coordonnée dans le temps.

Les blocs fonction de la bibliothèque "MPLCTecCamSwitch_MX" règlent les paramètres généraux de la boîte à cames et configurent les voies et les fenêtre de cames.

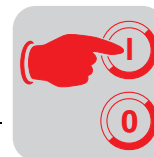
Une connaissance approfondie des paramètres respectifs des différentes fonctions technologiques n'est ainsi pas nécessaire. La configuration avec l'éditeur technologique n'est pas nécessaire, tous les réglages pertinents sont effectués par les blocs fonction de la bibliothèque "MPLCTecCamSwitch_MX".

Les avantages de la bibliothèque "MPLCTecCamSwitch_MX" sont :

- Paramétrage selon CEI 61131, permettant l'implémentation aisée des applications spécifiques.
- Bibliothèque pour le pilotage de la fonction technologique Boîte à cames. Celle-ci permet la configuration et le pilotage des fonctionnalités Boîte à cames sans nécessiter de connaissances approfondies des différentes fonctions technologiques.
- Pilotage centralisé de plusieurs boîtes à cames de différents axes.
- Fonctionnalités complémentaires, comme p. ex. la définition de la valeur maître par un codeur maître virtuel, commandées de manière centralisée via la commande MOVI-PLC®.
- Sources de données différentes des voies respectives facilement paramétrables.
- Modification simplifiée des fenêtres de cames en cours de fonctionnement ou en cas de changement de format.

D'autres informations figurent dans les documentations suivantes :

- Les remarques générales concernant le fonctionnement de la bibliothèque figurent dans le document "Bibliothèques MPLCMotion_MDX et MPLCMotion_MX pour MOVI-PLC®".
- Manuel "Fonction technologique réducteur électronique pour MOVIAXIS® MX".



4.2 Domaines d'application

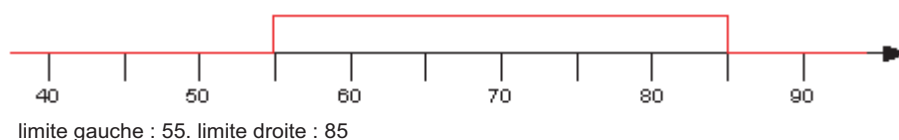
4.2.1 Fenêtre de cames et voies

La boîte à cames MOVIAxis® comprend 8 voies de cames ; chaque voie peut être affectée de 16 fenêtres avec un maximum de 32 pour les 8 voies. Une limite gauche et droite sont écrites dans les fenêtres de cames.

Chaque voie de came dispose d'un mode d'exploitation. Celle-ci permet d'une part de désactiver une voie, d'autre part de décider si les cames affectées doivent être traitées en mode d'exploitation standard ou modulo.

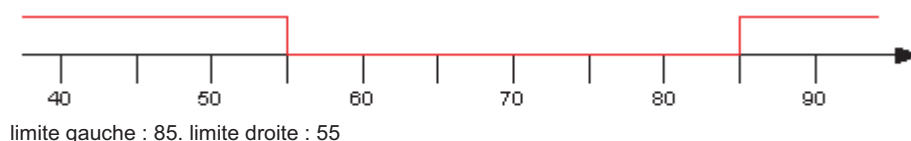
En mode d'exploitation "Standard", la valeur limite gauche de chaque came affectée doit en permanence être plus petite ou égale à la valeur limite droite. Si le mode d'exploitation est réglé sur "Modulo", les valeurs limite d'une came peuvent avoir les valeurs souhaitées. Les limites de cames font partie intégrante des cames.

L'illustration suivante présente une voie de came modulo / standard : limite gauche < limite droite.



64568afr

L'illustration suivante montre une voie de came modulo : limite gauche > limite droite



64569afr

Une fenêtre de came dispose également d'un sens de traitement. L'indication du sens de rotation permet de définir si la fenêtre doit être activée pour le dépassement par la gauche, par la droite, des deux côtés ou d'aucun côté.

Les conditions suivantes s'appliquent pour l'activation de la came :

- Les variations des valeurs doivent correspondre au sens d'action de la came.
- La valeur actuelle doit faire partie intégrante de la came :
 - En cas de came standard avec valeur limite gauche inférieure ou égale à la valeur limite droite : valeur \geq limite gauche **et** valeur \leq limite droite.
 - En cas de came modulo avec valeur limite gauche supérieure à la valeur limite droite : valeur \geq limite gauche **ou** valeur \leq limite droite.

Cette définition permet à la fois une action de la came en cas de dépassement liée à la direction et une réaction correcte en cas de réveil sur une came suivi d'un déplacement à l'intérieur de la fenêtre de came.

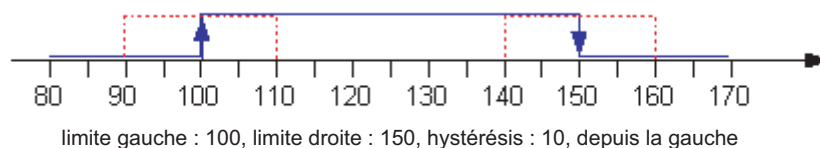
La règle suivante s'applique pour la désactivation de la came :

- La valeur actuelle doit se situer hors de la plage de came.

En cas de désactivation, la direction n'est pas prise en compte. Elle n'a d'importance que pour l'activation. Ceci permet d'obtenir le comportement souhaité, à savoir qu'en cas d'inversion de sens sur une came dépendant du sens de rotation le résultat de la voie reste à "un" jusqu'à ce que l'entraînement ait quitté la came et ne soit pas immédiatement égal à "zéro" dès l'inversion du sens.



Exemple :



64570afr

Valeur	Résultat de la voie	Commentaires
80	0	Valeur à l'extérieur de la came
90	0	Valeur à l'extérieur de la came
100	1	La limite de came fait partie intégrante de la came, fenêtre d'hystérésis : 90/110.
120	1	Valeur sur la came
119	1	Changement de sens, mais uniquement sur la came
100	1	Valeur sur la came
99	0	Quitter la came, fenêtre d'hystérésis : 90/110

Ces règles permettent cependant également d'obtenir le comportement suivant, éventuellement non souhaité :

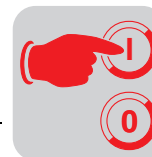
Valeur	Résultat de la voie	Commentaires
99	0	Valeur à l'extérieur de la came
100	1	La limite de came fait partie intégrante de la came, direction correcte, fenêtre d'hystérésis : 90/110.
130	1	Valeur dans la plage de came
150	1	La limite de came fait partie intégrante de la came.
151	0	Quitter la came, fenêtre d'hystérésis : 140/160.
130	0	Quitter la fenêtre d'hystérésis, hystérésis désactivé.
135	1	Valeur dans la plage de came, direction correcte, pas d'hystérésis
100	1	Valeur dans la plage de came
99		Quitter la came, fenêtre d'hystérésis : 90/110.

Chaque voie dispose de consignes et de mesures propres. Toutes les grandeurs présentes dans le DDB sont possibles au choix. Pour cette grandeur, il est possible d'indiquer une compensation de temps mort de -500 μ s à +500 μ s par pas de 0.1- μ s. Pour cela la comparaison s'effectue non pas avec la valeur de comparaison actuelle, mais avec la valeur calculée par avance ou après coup pour le temps mort.

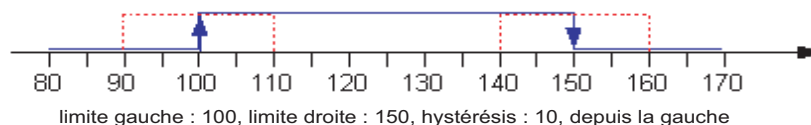
Par cette méthode, on risque de calculer par avance des valeurs fausses en raison d'une valeur mesurée faussée. Pour obtenir un lissage, il est possible d'indiquer une base de temps pour le calcul. Cette durée indique alors l'ancienneté de la valeur précédente pour l'extrapolation linéaire. La direction de ces droites est identique au sens de déplacement auquel est comparée la direction de la came, c.-à-d. que la base de temps effectue également un lissage des sens pour les valeurs comparées.

Si la valeur mesurée varie autour du front de déclenchement d'une came, la réaction de déclenchement varie également. Afin de prévenir ce comportement non souhaité, chaque voie de came possède une valeur d'hystérésis. Cette valeur indique une fenêtre autour du front de déclenchement actuel, p. ex. valeur limite de came gauche - hystérésis jusqu'à valeur limite de came + hystérésis.

Après modification du résultat de déclenchement d'une voie, cette fenêtre est définie et activée. Tant que la valeur de la source de données se trouve dans cette fenêtre, la voie n'est pas traitée une nouvelle fois. Les limites font partie intégrante de la fenêtre d'hystérésis.

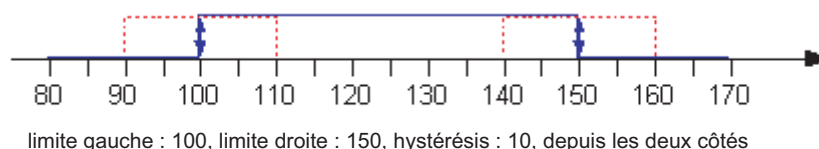


Exemples de résultats de déclenchement :



64571afr

Valeur	Résultat de la voie	Commentaires
80	0	Valeur à l'extérieur de la came
90	0	Valeur à l'extérieur de la came
100	1	La limite de came fait partie intégrante de la came, fenêtre d'hystérésis : 90/110.
90	1	La limite d'hystérésis fait encore partie de l'hystérésis → pas de nouveau traitement !
150	1	La limite de came fait partie intégrante de la came.
151	0	Valeur à l'extérieur de la came, fenêtre d'hystérésis : 140/160
140	0	La limite d'hystérésis fait encore partie de l'hystérésis → pas de nouveau traitement !
139	0	Valeur sur came à l'extérieur de l'hystérésis, fenêtre d'hystérésis désactivée, fausse direction !
151	0	Valeur à l'extérieur de la came
170	0	Valeur à l'extérieur de la came
150	0	Valeur sur came, fausse direction !
160	0	Valeur à l'extérieur de la came
161	0	Valeur à l'extérieur de la came



64572afr

Valeur	Résultat de la voie	Commentaires
80	0	Valeur à l'extérieur de la came
90	0	Valeur à l'extérieur de la came
100	1	La limite de came fait partie intégrante de la came, fenêtre d'hystérésis : 90/110.
90	1	La limite d'hystérésis fait encore partie de l'hystérésis → pas de nouveau traitement !
150	1	La limite de came fait partie intégrante de la came.
151	0	Valeur à l'extérieur de la came, fenêtre d'hystérésis : 140/160.
140	0	La limite d'hystérésis fait encore partie de l'hystérésis → pas de nouveau traitement !
139	1	Valeur sur came à l'extérieur de l'hystérésis, fenêtre d'hystérésis désactivée
151	0	Valeur à l'extérieur de la came
170	0	Valeur à l'extérieur de la came
150	1	La limite de came fait partie intégrante de la came, fenêtre d'hystérésis : 140/160.
160	1	La limite d'hystérésis fait encore partie de l'hystérésis → pas de nouveau traitement !
161	0	Valeur à l'extérieur de l'hystérésis



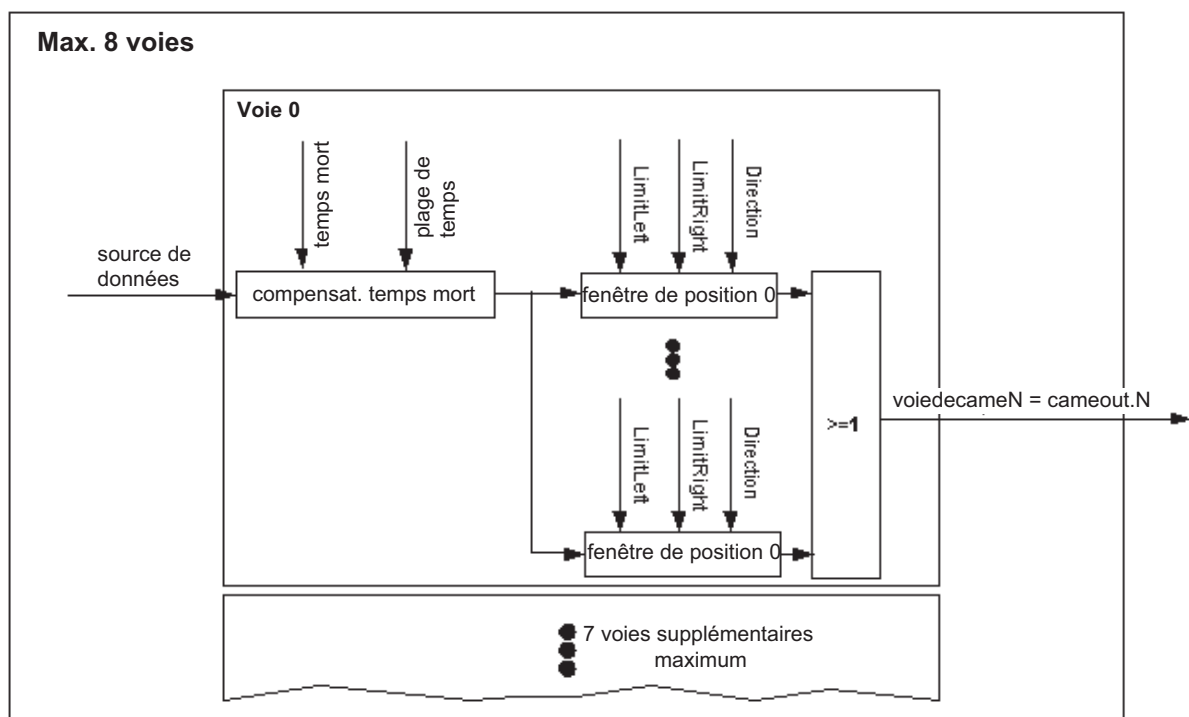
4.2.2 Vitesse de traitement

La fonction boîte à cames permet de traiter jusqu'à huit voies avec une vitesse de traitement de 1 ms.

Les voies utilisent le mode dynamique linéaire. Avec ce mode, toutes les fenêtres de came attribuées pour toutes les voies de came sont parcourues lors du traitement jusqu'à la fin ou jusqu'à ce qu'une fenêtre détermine la sortie.

Afin d'obtenir un déclenchement précis en sortie, il est possible d'utiliser les résultats de quatre de ces voies comme consignes pour les entrées/sorties *Rapides*. Pour cela, les résultats des voies sont reproduits sur l'index 10488.2. Dans l'éditeur PDO, l'index est inséré dans la sortie-process 9 ; de là, le bit de résultat peut être commuté depuis l'éditeur PDO sur un bit dans un mot de commande puis commuté à nouveau sur les sorties binaires de l'appareil. Avec ce branchement, l'entrée/sortie est traitée automatiquement par la boîte à cames comme *entrée/sortie rapide*. Il est également possible de choisir un branchement sur les cartes option.

L'illustration suivante montre le schéma synoptique de la boîte à cames.



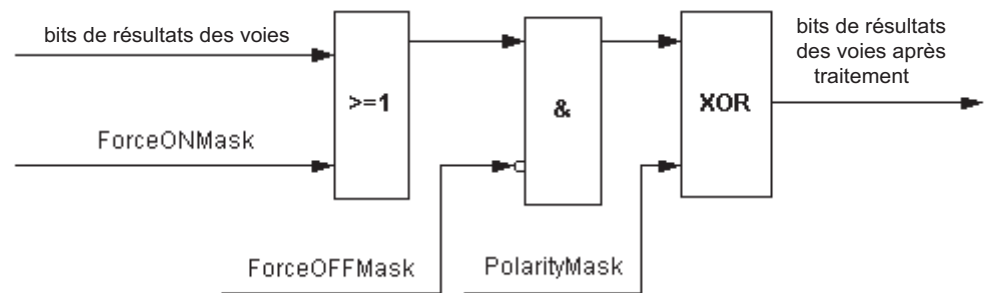
64573afr



4.2.3 Résultats

Les sorties des voies de came respectives dépendent de plusieurs paramètres. "ForceOnMaske" permet d'activer en continu certaines sorties. "ForceOffMaske" permet de désactiver les sorties en continu. Elle est prioritaire sur "ForceOnMaske". "PolarityMaske" permet en outre de définir si une sortie doit être *activée à l'état high* ou *activée à l'état low*.

L'illustration suivante montre le traitement des sorties des voies.



64574afr

La fonction boîte à came écrit toujours les sorties dans sa plage de données, dans le buffer Data Distribution, à partir duquel le résultat peut être utilisé sous forme de donnée-process. Ainsi les sorties des voies peuvent être mises à disposition sous différentes formes, p. ex. via les sorties physiques ou un bus de terrain. Le résultat se trouve également dans le paramètre fixe 10488.2.

	Opération	Exemples de valeurs	Commentaires
Résultats des voies		0b10101010	
Masque ForceOn	OR	0b00010101	Les voies 0 et 2 sont toujours mises à 1.
Masque ForceOff	AND, valeur inv.	0b00010010	Les voies 1 et 4 sont toujours mises à 0. Saute la ligne précédente !
Masque de polarité	XOR	0b00000100	La voie 2 est modifiée dans la polarité.
Résultat final dans 10488.2		0b10101001	



4.3 Configuration

Pour pouvoir utiliser la bibliothèque "MPLCTecCamSwitch_MX", une commande MOVI-PLC® en exécution technologique T1 ou ultérieure est nécessaire.

La bibliothèque "MPLCMotion_MX" doit également être intégrée. Les conditions et restrictions indiquées dans le manuel "MPLCMotion_MX" s'appliquent par ailleurs.

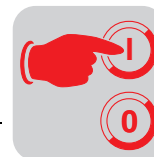
4.3.1 Principe de base et remarques

- Réglage des fonctions technologiques adéquates de l'axe via blocs fonction CEI.
- Le bloc fonction "MC_LinkTecCamSwitch_MX" est absolument nécessaire comme interface de données centrale pour la fonction technologique "Boîte à cames".
- L'éditeur technologique de MotionStudio n'est pas nécessaire ; pour le diagnostic, il est possible, grâce au projet exemple "MOVI-PLC projetpardéfaut.....fonctionTec", de démarrer un moniteur en ligne. Ne pas effectuer cependant de chargement de l'exemple de projet, sans quoi les réglages effectués par la bibliothèque seraient à nouveau modifiés.

4.3.2 Logique de détermination

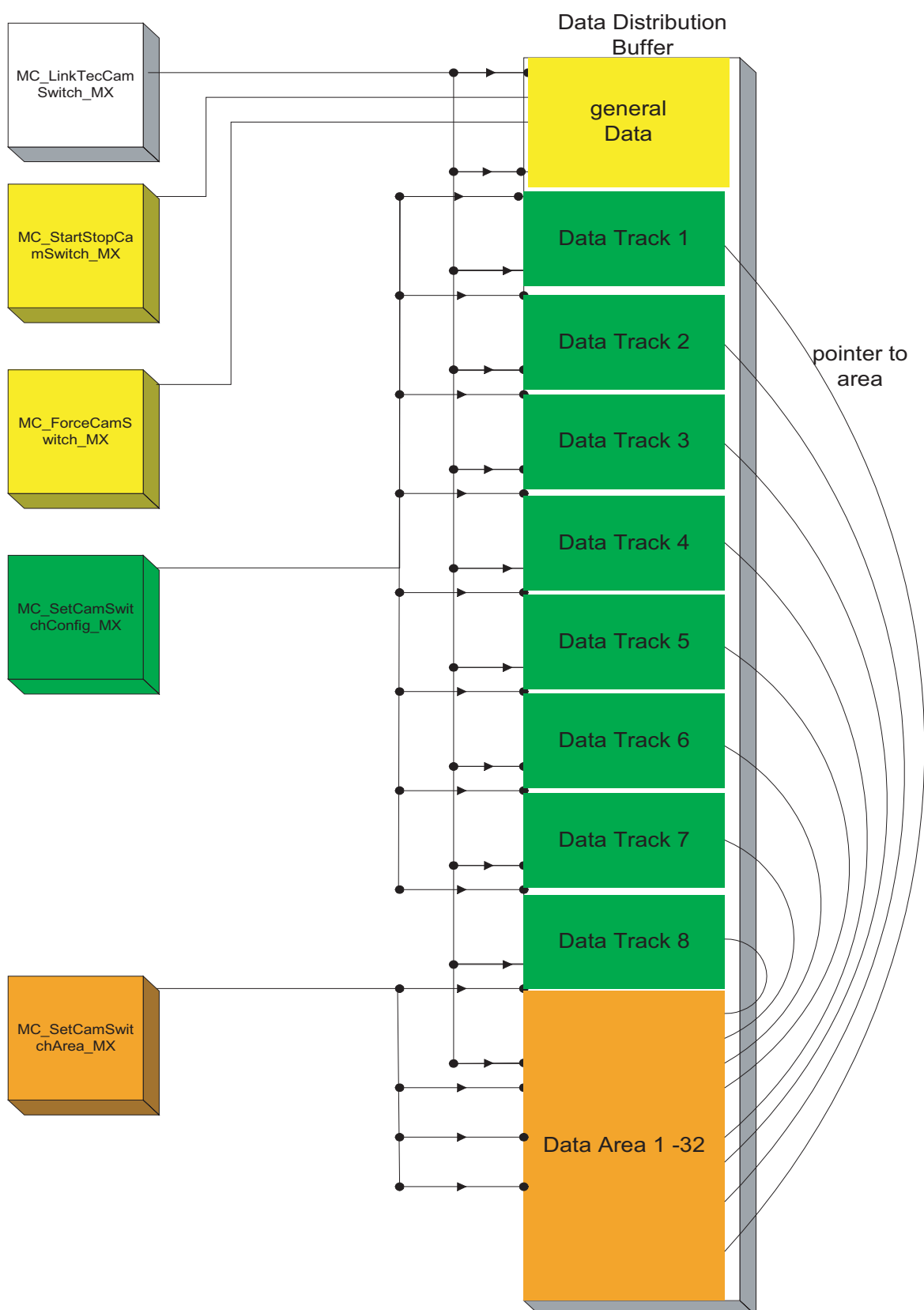
Procédure

- Insérer dans le projet la bibliothèque "MPLCTecCamSwitch_MX" avec le gestionnaire des bibliothèques, en complément à la bibliothèque "MPLCMotion_MX".
- Le bloc fonction "MC_LinkTecCamSwitch_MX" est l'interface de la fonction technologique vers l'axe et doit donc être appelé de manière cyclique dans le programme. Lorsque le bloc fonction "MC_ConnectAxis_MX" a établi la liaison avec l'axe (*Done* = True), le bloc fonction "LinkTecCamSwitch" active les fonctions technologiques du MOVIAXIS® nécessaires pour la boîte à cames. Les réglages nécessaires sont également effectués dans l'éditeur PDO.
- Le bloc fonction "MC_StartStopCamConfig_MX" permet d'activer la boîte à cames.
- Le bloc fonction "MC_SetCamSwitchAreas_MX" permet d'écrire les limites de came dans le MOVIAXIS®.
- Le bloc fonction "MC_SetCamSwitchConfig_MX" permet de régler les paramètres des voies et les affectations de cames.
- Le bloc fonction "M_ForceCamSwitch_MX" permet d'activer bit à bit en permanence ou de désactiver les sorties (voies) de la boîte à cames.



4.3.3 Stockage des données dans le MOVIAXIS®

Vue d'ensemble du stockage des données



64591axx



Toutes les valeurs et réglages de la boîte à cames sont sauvegardés dans le buffer Data Distribution. Le début de la structure DDB avec la configuration de boîte à cames est sauvegardé dans le paramètre 10488.1.

Les blocs de données DDB sont détaillés ci-après. Pour les blocs de données, l'offset et la longueur sont calculés dans des variables DDB, c.-à-d. en tant que valeurs 32 bits.

4.3.4 Structure DDB complète

Offset	Nom	GUI-Nom	Plage de valeurs
0	Longueur	-	582
1	Type et version	-	Type : 3 (mot high) Version principale : 1 (octet de poids le plus fort du mot low) Sous-version : 1 (octet de poids le plus faible du mot low)
2	Bloc fonction général		
22	8 blocs de données de voies de cames		
262	32 blocs de données de fenêtre de came		

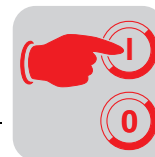
Réglages généraux

Les données générales sont écrites dans le MOVIAXIS® via le bloc fonction "MC_LinkTecCamSwitch_MX". Les voies de la boîte à cames peuvent être branchées sur les sorties standard et sur les sorties de cartes option.

Les sorties de la boîte à came peuvent être mises à "1" ou à "0" une à une. La logique peut par ailleurs être modifiée. La définition manuelle des sorties-process (forçage) peut s'effectuer via le bloc fonction "MC_ForceCamSwitch_MX". Le contrôleur Cam peut être activé ou désactivé via le bloc fonction "MC_StartStopCamSwitch_MX".

Si l'adresse de démarrage (10488.1) du contrôleur Cam est réglée sur 815, il en résulte les adresses suivantes :

Offset	Index / Sous-index	Description
819	20206.52	Mode de fonctionnement (0 = désactivé / 1 = activé)
820	20206.53	Sous-mode de fonctionnement
821	20206.54	Affichage d'état
822	20206.55	Masque "Force On"
823	20206.56	Masque "Force OFF"
824	20206.57	Masque "polarité"
825	20206.58	Résultat du contrôleur CAN



Stockage des voies CAM 1 - 8

Il est possible de régler une à huit voies de cames. Chaque voie a des possibilités de réglage propres, voir type de donnée : "MC_CAMSWITCH_CONFIG_MX". Les données sont écrites dans le MOVIAXIS® via le bloc fonction "MC_SetCamSwitchConfig_MX". Si l'adresse de démarrage (10488.1) du contrôleur Cam est réglée sur la valeur 815, il en résulte les adresses suivantes :

Offset	Index / Sous-index	Description	Voie n°
839	20206.72	Mode de fonctionnement (0 = désactivé / 1 = activé)	1
840	20206.73	Sous-mode de fonctionnement (0 = standard / 1 = modulo)	1
841	20206.74	Etat	1
842	20206.75	Sources de données	1
843	20206.76	Temps mort (-500000ms-500000ms)	1
844	20206.77	Plage de temps (0 - 16 ms)	1
845	20206.78	Valeur modulo min.	1
846	20206.79	Valeur modulo max.	1
847	20206.80	Hystérésis	1
851	20206.84	Affectation de came 1	1
852	20206.85	Affectation de came 2	1
853	20206.86	Affectation de came 3	1
854	20206.87	Affectation de came 4	1
855	20206.88	Affectation de came 5	1
856	20206.89	Affectation de came 6	1
857	20206.90	Affectation de came 7	1
858	20206.91	Affectation de came 8	1
859	20206.92	Affectation de came 9	1
860	20206.93	Affectation de came 10	1
861	20206.94	Affectation de came 11	1
862	20206.95	Affectation de came 12	1
863	20206.96	Affectation de came 13	1
864	20206.97	Affectation de came 14	1
865	20206.98	Affectation de came 15	1
866	20206.99	Affectation de came 16	1
...			
869	20206.72	Mode de fonctionnement	2
...			
896	20207.01	Affectation de came 16	2
...			
899	20207.04	Mode de fonctionnement	3
...			
926	20207.31	Affectation de came 16	3
...			
929	20207.34	Mode de fonctionnement	4
...			
956	20207.61	Affectation de came 16	4
...			



Offset	Index / Sous-index	Description	Voie n°
959	20207.64	Mode de fonctionnement	5
...			
986	20207.91	Affectation de came 16	5
...			
989	20207.94	Mode de fonctionnement	6
...			
1016	20207.121	Affectation de came 16	6
...			
1019	20207.124	Mode de fonctionnement	7
...			
1046	20208.23	Affectation de came 16	7
...			
1059	20208.24	Mode de fonctionnement	8
...			
1076	20208.53	Affectation de came 16	8

Pour l'affichage des affectations de came, ouvrir l'éditeur technologique avec un projet par défaut.

Voies	Came
Voie 1	1 2
Voie 2	3 4
Voie 3	5 6
Voie 4	7 8
Voie 5	9 10
Voie 6	11 12
Voie 7	13 14
Voie 8	1 2 3 4 5 6 7 8 9 10 11 12

64592axx

Attention :

La valeur modulo maximale ne peut jamais être atteinte et ne peut donc pas servir de limite de came.



REMARQUES

Pour être assuré que les valeurs des cames affectées dans le contexte de la voie sont correctes, la voie doit impérativement être activée. Les valeurs de came peuvent cependant également être modifiées en conséquence. En raison de la durée de fonctionnement, les valeurs des cames affectées ne font cependant pas l'objet d'une vérification lorsqu'une came est activée.



Stockage des Cam Areas 1 - 32 (fenêtre de came)

Le nombre maximal de cames est de 32. Chaque came a une limite gauche et une limite droite ainsi qu'un sens d'activation. Voir à ce sujet type de données : "MC_CAMSWITCH_AREAS_MX".

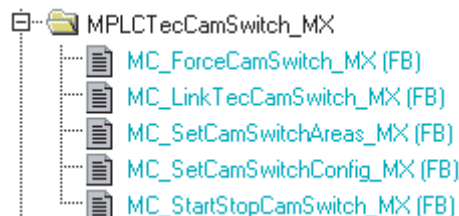
Les données sont écrites dans le MOVIAXIS® via le bloc fonction "MC_SetCamSwitchAreas_MX". Lorsque l'adresse de démarrage (10488.1) du contrôleur Cam est réglée sur la valeur 815, il en résulte les adresses suivantes :

Offset	Index / Sous-index	Description	N°
1079	20208.56	Limite Cam gauche	1
1080	20208.57	Limite Cam droite	1
1081	20208.58	Sens Cam	1
..			
1089	20208.66	Limite Cam gauche	2
1090	20208.67	Limite Cam droite	2
1091	20208.68	Sens Cam	2
...			
1099	20208.76	Limite Cam gauche	3
1100	20208.77	Limite Cam droite	3
1101	20208.78	Sens Cam	3
...			...
...			...
1179	20209.28	Limite Cam gauche	10
1180	20209.29	Limite Cam droite	10
1181	20209.30	Sens Cam	10
...			
1189	20209.38	Limite Cam gauche	11
1190	20209.39	Limite Cam droite	11
1191	20209.40	Sens Cam	11
...			...
...			...
1279	20209.128	Limite Cam gauche	20
1280	20210.1	Limite Cam droite	20
1281	20210.2	Sens Cam	20
...			
1289	20210.10	Limite Cam gauche	21
1290	20210.11	Limite Cam droite	21
1291	20210.12	Sens Cam	21
...			...
...			...
1379	20210.100	Limite Cam gauche	31
1380	20210.101	Limite Cam droite	31
1381	20210.102	Sens Cam	31
...			
1389	20210.100	Limite Cam gauche	32
1390	20210.101	Limite Cam droite	32
1391	20210.102	Sens Cam	32



4.4 La bibliothèque **MPLCTecCamSwitch_MX**

4.4.1 Bref aperçu de la bibliothèque **MPLCTecCamSwitch_MX**



64593axx

MC_LinkTecCamSwitch_MX

Description

Le bloc fonction "MC_LinkTecCamSwitch_MX" a diverses fonctions :

- Configuration des fonctions technologiques du MOVIAXIS® et activation des fonctions et paramètres nécessaires pour Cam Switch
- Configuration de l'interface de données-process MOVIAXIS®
- Ecriture des voies du contrôleur Cam sur les sorties binaires du MOVIAXIS®
- Indication de l'état d'initialisation

MC_SetCamSwitchAreas_MX

Description

Ce bloc fonction écrit les fenêtres de came dans le MOVIAXIS®. Il est possible d'écrire sur certaines fenêtres ou sur l'ensemble des 32 fenêtres.

Condition :

"MC_LinkTecCamSwitch_MX" signale Done.

MC_SetCamSwitchConfig_MX

Description

Ce bloc fonction active chaque voie sélectionnée (1 - 8) et écrit les paramètres généraux ainsi que les affectations de cames. Il est possible d'affecter jusqu'à 16 cames (fenêtre ou Area) à une fenêtre.

Condition :

"MC_LinkTecCamSwitch_MX" signale Done.

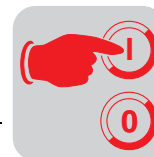
MC_StartStopCamSwitch_MX

Description

Ce bloc fonction active ou désactive le contrôleur Cam.

Condition :

"MC_LinkTecCamSwitch_MX" signale Done.



MC_ForceCamSwitch_MX

Description

Ce bloc fonction permet de mettre à "1" ou à "0" individuellement les blocs fonction. La logique risque d'être modifiée.

Condition :

"MC_LinkTecCamSwitch_MX" signale Done.

4.4.2 Les types de données de la bibliothèque MPLCTecCamSwitch_MX



64594axx

MC_CAMSWITCH_AREAS_MX

```

TYPE MC_CAMSWITCH_AREAS_MX :
STRUCT
  LeftLimit:    DINT; (*Left limit cam window*)
  RightLimit:   DINT; (*Right limit cam window*)
  Direction:    DINT; (* 0 = off , 1 = from left , 2 = from right , 3 = both direction*)
END_STRUCT
END_TYPE
  
```

64595axx

**MC_CAMSWITCH_CONFIG_MX**

TYPE MC_CAMSWITCH_CONFIG_MX :

STRUCT

```

Mode          :BOOL;          (* 0 = off , 1 = on*)
ModeControl   :BOOL;          (* 0 = Standard cam track , 1 = Modulo cam track*)
DataSource:    MC_CAMSWITCH_SOURCE_MX; (*datasource of the cam track*)
DeadTime      :DINT;          (*-500.000 - + 500.000 microsecond steps 0.1ms*)
TimeFrame     :DINT;          (* 1-16 in millisecond steps*)
MinModuloValue :DINT;          (* min modulo value*)
MaxModuloValue :DINT;          (* max modulo value*)
Hysteresis    :DINT;          (*Hysteresis of the cam window*)
SelectArea:    ARRAY[1..16] OF DINT; (* number of the Cam areas 1 - 32 possible*)
State         :DINT;          (*
                                0 = off
                                1 = false lenght error 38 suberror 184
                                2 = false typ error 38 suberror 185
                                3 = false version error 38 suberror 186
                                4 = false datasource error 38 suberror 187
                                5 = false cam number error 38 suberror 188
                                6 = false cam data error 38 suberror 189
                                7 = false cam track error 38 suberror 190
                                8 = not enough comparison value
                                9 = initialisation
                                10= activ*)

```

END_STRUCT

END_TYPE

64596axx

MC_CAMSWITCH_DESTINATION_MX

TYPE MC_CAMSWITCH_DESTINATION_MX :

(

```

MX_CAMSWITCH_DEST_NONE      := 0,
MX_CAMSWITCH_DEST_X11       := 1,
MX_CAMSWITCH_DEST_OPT1      := 2,
MX_CAMSWITCH_DEST_OPT2      := 4,
MX_CAMSWITCH_DEST_X11_OPT1  := 3,
MX_CAMSWITCH_DEST_X11_OPT2  := 5,
MX_CAMSWITCH_DEST_OPT1_OPT2 := 6,
MX_CAMSWITCH_DEST_X11_OPT1_OPT2:= 7

```

);

END_TYPE

```

(* MX_CAMSWITCH_DEST_NONE      := 0,      none
   MX_CAMSWITCH_DEST_X11       := 1,      Standard output / X11 DO 0 - 3
   MX_CAMSWITCH_DEST_OPT1      := 2,      PDO Output option 1 / DO 0-7
   MX_CAMSWITCH_DEST_OPT2      := 4,      PDO Output option 2 / DO 0-7
   MX_CAMSWITCH_DEST_X11_OPT1  := 3,      Standard output / X11 DO 0 - 3 and PDO Output option 1 / DO 0-7
   MX_CAMSWITCH_DEST_X11_OPT2  := 5,      Standard output / X11 DO 0 - 3 and PDO Output option 2 / DO 0-7
   MX_CAMSWITCH_DEST_OPT1_OPT2 := 6,      PDO Output option 1 / DO 0-7 and PDO Output option 2 / DO 0-7
   MX_CAMSWITCH_DEST_X11_OPT1_OPT2:= 7,    Standard output and PDO Output option 1 and PDO Output option 2
*)

```

64597axx



MC_CAMSWITCH_SOURCE_MX

```

TYPE MC_CAMSWITCH_SOURCE_MX :
(
    MX_CAMSWITCH_SYSTEMPOS_ENCODER1:=109,      (* Master position is the linear position of encoder 1*)
    MX_CAMSWITCH_SYSTEMPOS_ENCODER2:=110,      (* Master position is the linear position of encoder 2*)
    MX_CAMSWITCH_SYSTEMPOS_ENCODER3:=111,      (* Master position is the linear position of encoder 3*)
    MX_CAMSWITCH_MODULOPOS_ENCODER1:=112,      (* Master position is the modulo position of encoder 1*)
    MX_CAMSWITCH_MODULOPOS_ENCODER2:=113,      (* Master position is the modulo position of encoder 2*)
    MX_CAMSWITCH_MODULOPOS_ENCODER3:=114,      (* Master position is the modulo position of encoder 3*)
    MX_CAMSWITCH_USERPOS_ENCODER1:=115,        (* Master position is the modulo position of encoder 1*)
    MX_CAMSWITCH_USERPOS_ENCODER2:=116,        (* Master position is the modulo position of encoder 2*)
    MX_CAMSWITCH_USERPOS_ENCODER3:=117,        (* Master position is the modulo position of encoder 3*)
    MX_CAMSWITCH_MODULO_VIRTUAL_ENCODER_MX:=156, (* Master position is the modulo position of the MX virtual encoder *)
    MX_CAMSWITCH_LINEAR_VIRTUAL_ENCODER_MX:=157, (* Master position is the linear position of the MX virtual encoder *)
    MX_CAMSWITCH_RECEIVE_PDO_1:=6,             (* Master position is the SBus receive objetk PDO1 *)
    MX_CAMSWITCH_RECEIVE_PDO_2:=8,             (* Master position is the SBus receive objetk PDO2 *)
    MX_CAMSWITCH_VAR_INPUT_2:=10,              (* Master position is the Variablen Input 2 *)
    MX_CAMSWITCH_VAR_INPUT_3:=12,              (* Master position is the Variablen Input 3 *)
    MX_CAMSWITCH_VAR_INPUT_4:=14,              (* Master position is the Variablen Input 4 *)
    MX_CAMSWITCH_VAR_INPUT_5:=16,              (* Master position is the Variablen Input 5 *)
    MX_CAMSWITCH_VAR_INPUT_6:=18,              (* Master position is the Variablen Input 6 *)
    MX_CAMSWITCH_VAR_INPUT_7:=20,              (* Master position is the Variablen Input 7 *)
    MX_CAMSWITCH_PSG_GEAR:=49,                 (* Master position is the Output Gear off the Position setpoint generator*)
    MX_CAMSWITCH_PSG_CAM:=50,                  (* Master position is the Output Cam off the Position setpoint generator*)
);
END_TYPE

```

64598axx

MC_CAMSWITCH_STATE_MX

```

TYPE MC_CAMSWITCH_STATE_MX :
(
    MX_CAMSWITCH_INACTIVE,
    MX_CAMSWITCH_ERROR LENGHT,
    MX_CAMSWITCH_ERROR_TYP,
    MX_CAMSWITCH_ERROR_VERSION,
    MX_CAMSWITCH_ERROR_DATASOURCE,
    MX_CAMSWITCH_ERROR_CAMNUMBER,
    MX_CAMSWITCH_ERROR_CAMAREA,
    MX_CAMSWITCH_ERROR_CAMTRACK,
    MX_CAMSWITCH_NOCOMPAREVALUE,
    MX_CAMSWITCH_INIT,
    MX_CAMSWITCH_ACTIVE
);
END_TYPE
(* MX_CAMSWITCH_INACTIVE          0 = off
   MX_CAMSWITCH_ERROR LENGHT      1 = false lenght error 38 suberror 184
   MX_CAMSWITCH_ERROR_TYP         2 = false typ error 38 suberror 185
   MX_CAMSWITCH_ERROR_VERSION     3 = false version error 38 suberror 186
   MX_CAMSWITCH_ERROR_DATASOURCE  4 = false datasource error 38 suberror 187
   MX_CAMSWITCH_ERROR_CAMNUMBER   5 = false cam number error 38 suberror 188
   MX_CAMSWITCH_ERROR_CAMAREA     6 = false cam area error 38 suberror 189
   MX_CAMSWITCH_ERROR_CAMTRACK    7 = false cam track error 38 suberror 190
   MX_CAMSWITCH_NOCOMPAREVALUE    8 = not enough comparison value
   MX_CAMSWITCH_INIT              9 = initialisation
   MX_CAMSWITCH_ACTIVE            10= activ*)

```

64599axx

**MC_CAMSWITCH_EXTENDED_CONFIG_MX**

```
TYPE MC_CAMSWITCH_EXTENDED_CONFIG_MX :  
STRUCT  
    Dummy : BOOL;  (*prepared for further use*)  
END_STRUCT  
END_TYPE
```

64600axx

MC_CAMSWITCH_EXTENDED_DIAG_MX

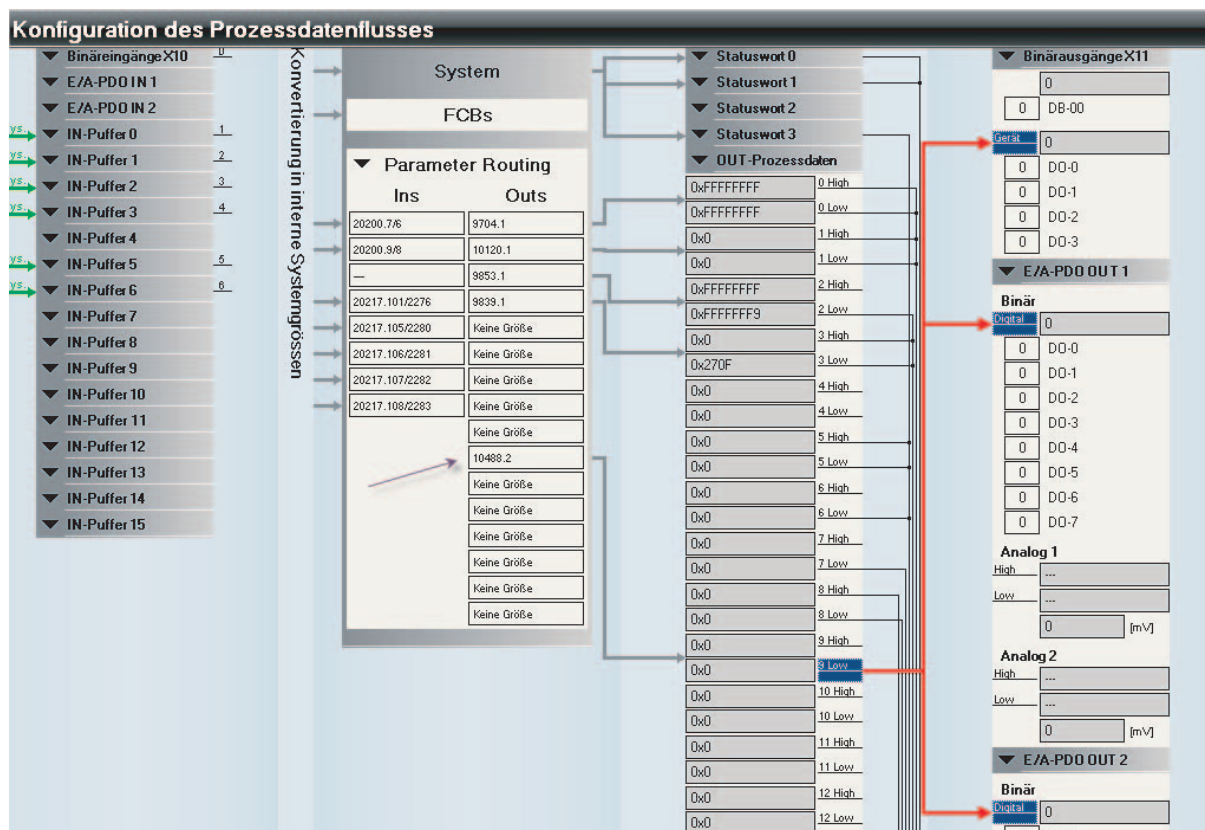
```
TYPE MC_CAMSWITCH_EXTENDED_DIAG_MX :  
STRUCT  
    Dummy : BOOL;  (*prepared for further use*)  
END_STRUCT  
END_TYPE
```

64601axx



4.4.3 Configuration PDO

En plus de la configuration effectuée via "MC_ConnectAxis_MX", le bloc fonction "MC_LinkTecCamSwitch_MX" effectue également certains réglages sur l'interface de données-process. Le résultat des voies est écrit sur *Out Process Data Low Word 9*. Selon la destination choisie, il est commuté sur les sorties Hardware, voir type de données "MC_CAMSWITCH_DESTINATION_MX".

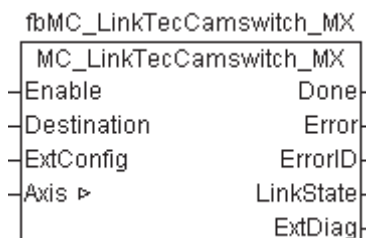


64602axx



4.5 Les blocs fonction de la bibliothèque MPLCTecCamSwitch_MX

4.5.1 MC_LinkTecCamSwitch_MX



64603axx

Description

Le bloc fonction "MC_LinkTecCamSwitch_MX" constitue l'interface logique entre l'axe et la fonction technologique "Boîte à cames". Lorsque "ConnectAxis" a configuré la communication cyclique vers l'axe et après libération par l'entrée "Enable", les fonctions technologiques respectives du MOVIAXIS® et celles nécessaires pour la fonction Boîte à cames sont d'abord configurées et activées.

L'interface de données-process est également réglée (OUT-Low Word 9), ainsi que le câblage vers les sorties physiques.

Attention :

Le bloc fonction "MC_LinkTecCamSwitch_MX" est indispensable pour l'utilisation des fonctionnalités de boîte à cames et doit être ouvert cycliquement dans le programme.

Si *LinkState* = NotLinked, aucun bloc fonction de la bibliothèque "MPLCTecCamSwitch_MX" ne peut être exécuté ; le message de défaut correspondant est généré.

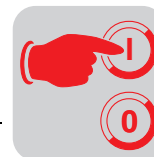
Condition :

MOVI-PLC® *advanced* en version technologique T1 ou supérieure.

ConnectAxis pour l'axe correspondant est intégré dans le programme cyclique.

Entrées :

Nom	Type	Signification
Enable	BOOL	Le bloc fonction est traité tant que Enable = True. L'initialisation est effectuée sur un front montant.
Destination	MC_CAMSWITCH_DESTINATION_MX	MX_CAMSWITCH_DEST_NONE = 0 MX_CAMSWITCH_DEST_X11 = 1 MX_CAMSWITCH_DEST_OPT1 = 2 MX_CAMSWITCH_DEST_OPT2 = 4 MX_CAMSWITCH_DEST_X11_OPT1 = 3 MX_CAMSWITCH_DEST_X11_OPT2 = 5 MX_CAMSWITCH_DEST_OPT1_OPT2 = 6 MX_CAMSWITCH_DEST_X11_OPT1_OPT2 = 7 Signification des abréviations : X11 - Sorties standard X11 D0 -D3 OPT1 - Carte option 1 XIO11A / XIA11A OPT2 - Carte option 2 XIO11A / XIA11A
ExtConfig	MC_CAMSWITCH_EXTENDED_CONFIG_MX	Configuration avancée
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

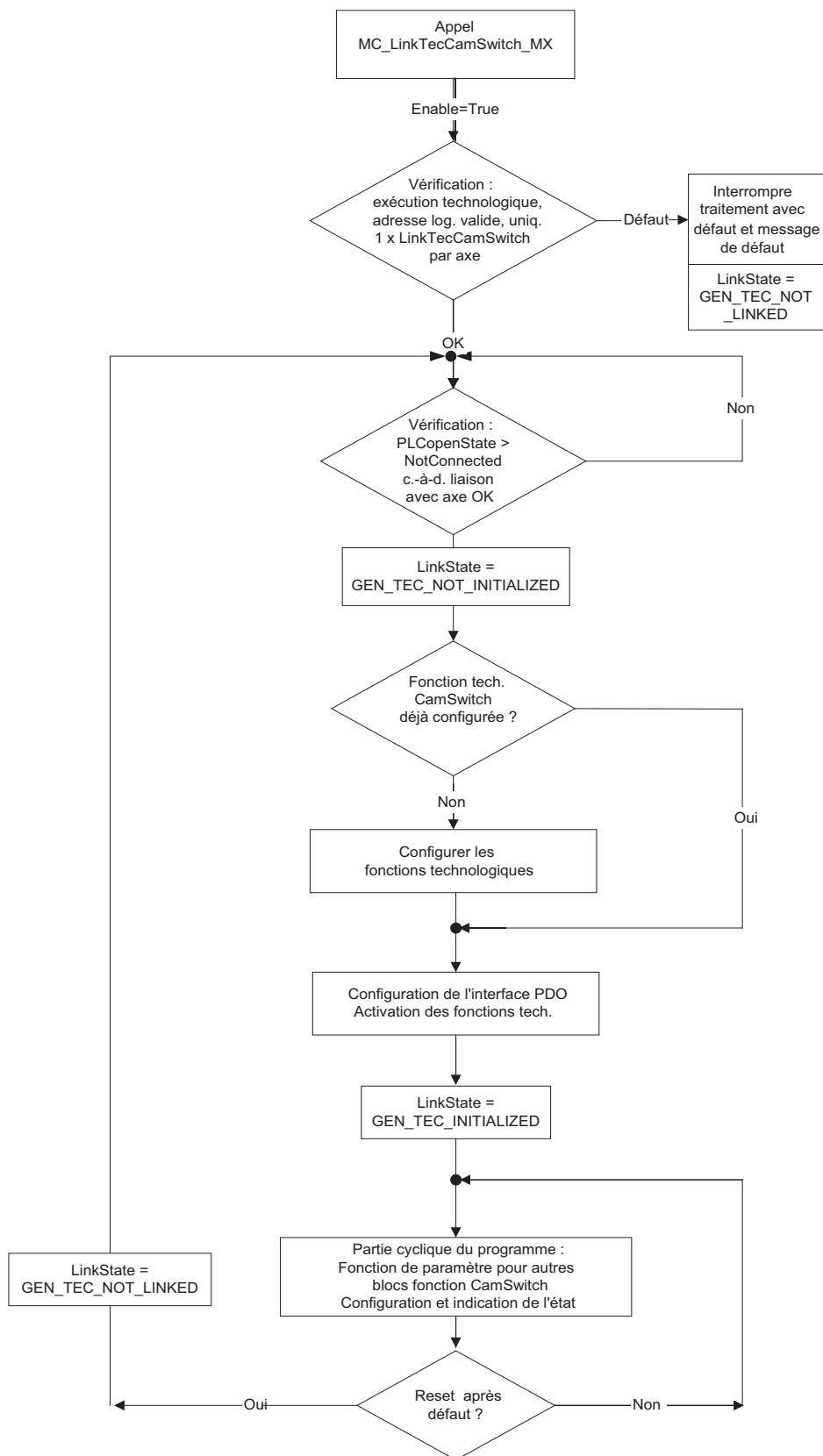


Sorties :

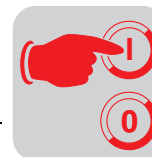
Nom	Type	Signification
Done	BOOL	Initialisation effectuée avec succès, liaison avec fonction technologique Cam établie.
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 163
LinkState	MC_LINKTECSTATE	Etat actuel : GEN_TEC_NOTLINKED (pas de liaison) GEN_TEC_NOTINITIALISED (initialisation non réalisée) GEN_TEC_INITIALISED (fonction technologique Cam Switch initialisée)
ExtDiag	MC_CAMSWITCH_EXTENDED_DIAG_MX	Diagnostic avancé



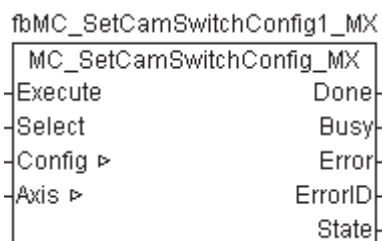
MC_LinkTecCam_MX : initialisation



64604afr



4.5.2 MC_SetCamSwitchConfig_MX



64605axx

Description

Ce bloc fonction règle les paramètres de base pour chaque voie Cam.

L'entrée *Select* sert à transmettre le numéro de la voie et l'ARRAY *Config* les paramètres de la voie. Une instance spécifique du bloc fonction est nécessaire pour chaque voie. Les paramètres sont pris en compte et écrits à l'emplacement correspondant dans le MOVIAXIS® lorsqu'un front montant est appliqué sur l'entrée *Execute*.

Recommandation:

Créer un array pour l'initialisation des paramètres *Config*.

Exemple : CamTrack : ARRAY[1..8] OF MC_CAMSWITCH_CONFIG_MX.

Les paramètres de l'array sont écrits avec un bloc d'initialisation.

Il est possible d'affecter jusqu'à 16 cames à chaque voie (Track). 32 cames (Areas) sont disponibles. Il est également possible de leur attribuer plusieurs voies.

Condition :

MC_LinkTecCamSwitch_MX a configuré la fonction technologique "CamSwitch".

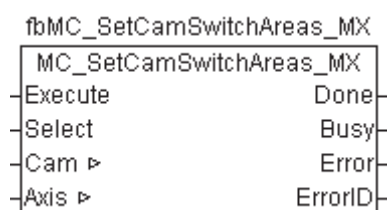
LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

Nom	Type	Signification
Execute	BOOL	Cette entrée lance l'exécution de la commande du bloc fonction par un front montant.
Select	UINT	1 = chargement paramètre voie Cam 1 2 = chargement paramètre voie Cam 2 3 = chargement paramètre voie Cam 3 4 = chargement paramètre voie Cam 4 5 = chargement paramètre voie Cam 5 6 = chargement paramètre voie Cam 6 7 = chargement paramètre voie Cam 7 8 = chargement paramètre voie Cam 8
Config	MC_CAMSWITCH_CONFIG_MX	Mode : 0 = désactivé / 1 = activé ModeControl : 0 - voie de cames standard 1 - voie de cames modulo Source : source de données pour la voie, voir MC_CAMSWITCH_SOURCE_MX DeadTime : -500.000[μs] - +500.000[μs] par pas de 100μs TimeFrame : 0-16 ms MinModulo : valeur modulo minimale MaxModulo : valeur modulo maximale Hystérésis : Hystérésis des cames SelectArea[1] - Numéro de la came * SelectArea[2] - Numéro de la came * SelectArea[16] - Numéro de la came * * Le numéro (1 - 32) de la came est indiqué.
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

**Sorties :**

Nom	Type	Signification
Done	BOOL	Tous les paramètres ont été transmis correctement.
Busy	BOOL	Transfert de paramètres en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir chapitre "Définition des défauts".
State	MC_CAMSWITCH_STATE_MX	MX_CAMSWITCH_INACTIVE = 0 MX_CAMSWITCH_ERROR_LENGTH = 1 MX_CAMSWITCH_ERROR_TYP = 2 MX_CAMSWITCH_ERROR_VERSION = 3 MX_CAMSWITCH_ERROR_DATASOURCE = 4 MX_CAMSWITCH_ERROR_CAMNUMBER = 5 MX_CAMSWITCH_ERROR_CAMAREA = 6 MX_CAMSWITCH_ERROR_CAMTRACK = 7 MX_CAMSWITCH_NOCOMPAREVALUE = 8 MX_CAMSWITCH_INIT = 9 MX_CAMSWITCH_ACTIVE = 10

4.5.3 MC_SetCamSwitchAreas_MX

64607axx

Description

Ce bloc fonction permet de transmettre les données de 32 camés (Areas) maximum.

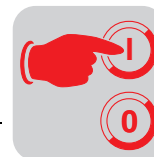
L'entrée *Select* permet de transmettre le numéro de la came et l'ARRAY Cam les données de la came. Si l'entrée *Select* n'est pas raccordée ou si la valeur 0 est transmise, tous les paramètres des 32 camés sont transmis.

Les paramètres sont pris en compte et écrits sur les emplacements correspondants dans le MOVIAXIS® lorsqu'un front montant est appliqué sur l'entrée *Execute*.

Condition :

MC_LinkTecCamSwitch_MX a configuré la fonction technologique "CamSwitch".

LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".



Entrées :

Nom	Type	Signification
Execute	BOOL	Cette entrée lance l'exécution de la commande du bloc fonction.
Select	UINT	0 - Chargement de toutes les Areas (cames) 1 - Chargement du paramètre Area 1 2 - Chargement du paramètre Area 2 ... 32 - Chargement du paramètre Area 32
Cam	Array [1..32] OFMC_CAMSWITCH_AREAS_MX	Cam[1].LeftLimit - limite gauche de la came Cam[1].RightLimit - limite droite de la came Cam[1].Direction - Limite d'effet 0 - désactivé 1 - depuis la gauche 2 - depuis la droite 3 - les deux sens ... Cam[...].LeftLimit - limite gauche de la came Cam[...].RightLimit - limite droite de la came Cam[...].Direction - Limite d'effet ... Cam[32].LeftLimit - limite gauche de la came Cam[32].RightLimit - limite droite de la came Cam[32].Direction - limite d'effet
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	Tous les paramètres ont été transmis correctement.
Busy	BOOL	Transfert de paramètres en cours
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 163



4.5.4 MC_StarStopCamSwitch_MX



64609axx

Description

Le bloc fonction "MC_StartStopCamSwitch" active la boîte à cames par *Enable* = True et désactive la boîte à cames par *Enable* = False.

Attention :

L'état *CamSwitchState* n'est actualisé qu'en cas de front positif ou négatif sur l'entrée *Enable*.

Condition :

MC_LinkTecCamSwitch_MX a configuré la fonction technologique CamSwitch.

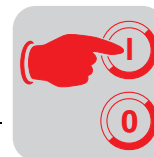
LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

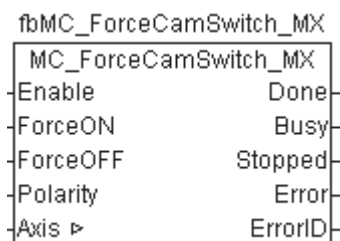
Nom	Type	Signification
Enable	BOOL	True = activation de la boîte à cames False = désactivation de la boîte à cames
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

Nom	Type	Signification
Done	BOOL	Boîte à cames activée
Busy	BOOL	Transfert de paramètres en cours
Stopped	BOOL	Boîte à cames désactivée
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 163
CamSwitchState	MC_CAMSWITCH_STATE_MX	MX_CAMSWITCH_INACTIVE = 0 MX_CAMSWITCH_ERROR_LENGTH = 1 MX_CAMSWITCH_ERROR_TYP = 2 MX_CAMSWITCH_ERROR_VERSION = 3 MX_CAMSWITCH_ERROR_DATASOURCE = 4 MX_CAMSWITCH_ERROR_CAMNUMBER = 5 MX_CAMSWITCH_ERROR_CAMAREA = 6 MX_CAMSWITCH_ERROR_CAMTRACK = 7 MX_CAMSWITCH_NOCOMPAREVALUE = 8 MX_CAMSWITCH_INIT = 9 MX_CAMSWITCH_ACTIVE = 10



4.5.5 MC_ForceCamSwitch_MX



64611axx

Description

Le bloc fonction "MC_ForceCamSwitch" permet de modifier individuellement les sorties de la boîte à cames. Il est également possible de modifier la polarité.

Si l'entrée *Enable* est un "1" logique, les paramètres sont modifiés. Si l'entrée est un "0" logique, l'activation, la désactivation ou la modification de la logique sont annulées.

Attention :

Si une sortie est simultanément activée et modifiée dans la logique, elle est remise à zéro en permanence. Ceci signifie que la modification de la logique s'effectue après définition manuelle des sorties-process (forçage).

Condition :

"MC_LinkTecCamSwitch_MX" a configuré la fonction technologique "CamSwitch".

LinkState ne doit pas passer à "GEN_TEC_NOTLINKED".

Entrées :

Nom	Type	Signification
Enable	BOOL	True = forçage activé False = pas de forçage
ForceON	Octet	Voies 1-8 <---> bits 0-7 Bit = 0 pas d'activation du bit de sortie Bit = 1 la voie est sur True en permanence
ForceOFF	Octet	Voies 1-8 <---> bits 0-7 Bit = 0 pas de désactivation de la sortie Bit = 1 la voie est sur False en permanence
Polarity	Octet	Voies 1-8 <---> bits 0-7 Bit = 0 pas de bit de rotation Bit = la sortie est modifiée (XOR)
Axis	AXIS_REF	Adresse logique de l'axe sur lequel le bloc fonction doit communiquer

Sorties :

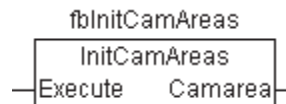
Nom	Type	Signification
Done	BOOL	Forçage activé
Busy	BOOL	Transfert de paramètres en cours
Stopped	BOOL	Forçage désactivé
Error	BOOL	Défaut dans le bloc fonction durant l'exécution
ErrorID	UINT	Voir page 163



4.6 Exemple

4.6.1 Huit voies et 32 cames

Les paramètres d'initialisation des cames sont à régler dans le bloc fonction "InitCamAreas".



64612axx

Déclaration

```
FUNCTION_BLOCK InitCamAreas
VAR_INPUT
    Execute:BOOL;
END_VAR
VAR_OUTPUT
    Camarea: ARRAY[1..32] OF MC_CAMSWITCH_AREAS_MX;
END_VAR
VAR
    EXE: BOOL ;
    bREdgeexecute: BOOL;
END_VAR
```

Programme

```
EXE:=Execute AND NOT bREdgeexecute;
bREdgeexecute:=Execute;
IF EXE THEN
(*area 1*)
Camarea[1].LeftLimit :=1000;
Camarea[1].RightLimit :=2000;
Camarea[1].Direction :=3;
(*area 2*)
Camarea[2].LeftLimit :=3000;
Camarea[2].RightLimit :=4000;
Camarea[2].Direction :=3;
(*area 3*)
Camarea[3].LeftLimit :=5000;
Camarea[3].RightLimit :=6000;
Camarea[3].Direction :=3;
(*area 4*)
Camarea[4].LeftLimit :=7000;
Camarea[4].RightLimit :=8000;
Camarea[4].Direction :=3;
(*area 5*)
Camarea[5].LeftLimit :=9000;
Camarea[5].RightLimit :=10000;
```



```
Camarea[5].Direction :=3;
(*area 6*)
Camarea[6].LeftLimit :=11000;
Camarea[6].RightLimit :=12000;
Camarea[6].Direction :=3;
(*area 7*)
Camarea[7].LeftLimit :=13000;
Camarea[7].RightLimit :=14000;
Camarea[7].Direction :=3;
(*area 8*)
Camarea[8].LeftLimit :=15000;
Camarea[8].RightLimit :=16000;
Camarea[8].Direction :=3;
(*area 9*)
Camarea[9].LeftLimit :=17000;
Camarea[9].RightLimit :=18000;
Camarea[9].Direction :=3;
(*area 10*)
Camarea[10].LeftLimit :=19000;
Camarea[10].RightLimit :=20000;
Camarea[10].Direction :=3;
(*area 11*)
Camarea[11].LeftLimit :=21000;
Camarea[11].RightLimit :=22000;
Camarea[11].Direction :=3;
(*area 12*)
Camarea[12].LeftLimit :=23000;
Camarea[12].RightLimit :=24000;
Camarea[12].Direction :=3;
(*area 13*)
Camarea[13].LeftLimit :=25000;
Camarea[13].RightLimit :=26000;
Camarea[13].Direction :=3;
(*area 14*)
Camarea[14].LeftLimit :=27000;
Camarea[14].RightLimit :=28000;
Camarea[14].Direction :=3;
(*area 15*)
Camarea[15].LeftLimit :=29000;
Camarea[15].RightLimit :=30000;
Camarea[15].Direction :=3;
(*area 16*)
Camarea[16].LeftLimit :=31000;
```



```

Camarea[16].RightLimit :=32000;
Camarea[16].Direction :=3;
(*area 17*)
Camarea[17].LeftLimit :=33000;
Camarea[17].RightLimit :=34000;
Camarea[17].Direction :=3;
(*area 18*)
Camarea[18].LeftLimit :=35000;
Camarea[18].RightLimit :=36000;
Camarea[18].Direction :=3;
(*area 19*)
Camarea[19].LeftLimit :=37000;
Camarea[19].RightLimit :=38000;
Camarea[19].Direction :=3;
(*area 20*)
Camarea[20].LeftLimit :=39000;
Camarea[20].RightLimit :=40000;
Camarea[20].Direction :=3;
(*area 21*)
Camarea[21].LeftLimit :=41000;
Camarea[21].RightLimit :=42000;
Camarea[21].Direction :=3;
(*area 22*)
Camarea[22].LeftLimit :=43000;
Camarea[22].RightLimit :=44000;
Camarea[22].Direction :=3;
(*area 23*)
Camarea[23].LeftLimit :=45000;
Camarea[23].RightLimit :=46000;
Camarea[23].Direction :=3;
(*area 24*)
Camarea[24].LeftLimit :=47000;
Camarea[24].RightLimit :=48000;
Camarea[24].Direction :=3;
(*area 25*)
Camarea[25].LeftLimit :=49000;
Camarea[25].RightLimit :=50000;
Camarea[25].Direction :=3;
(*area 26*)
Camarea[26].LeftLimit :=51000;
Camarea[26].RightLimit :=52000;
Camarea[26].Direction :=3;
(*area 27*)

```

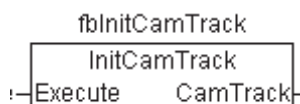


```

Camarea[27].LeftLimit :=53000;
Camarea[27].RightLimit :=54000;
Camarea[27].Direction :=3;
(*area 28*)
Camarea[28].LeftLimit :=55000;
Camarea[28].RightLimit :=56000;
Camarea[28].Direction :=3;
(*area 29*)
Camarea[29].LeftLimit :=57000;
Camarea[29].RightLimit :=58000;
Camarea[29].Direction :=3;
(*area 30*)
Camarea[30].LeftLimit :=59000;
Camarea[30].RightLimit :=60000;
Camarea[30].Direction :=3;
(*area 31*)
Camarea[31].LeftLimit :=61000;
Camarea[31].RightLimit :=62000;
Camarea[31].Direction :=3;
(*area 32*)
Camarea[32].LeftLimit :=63000;
Camarea[32].RightLimit :=64000;
Camarea[32].Direction :=3;
END_IF

```

Les paramètres d'initialisation des voies sont à régler dans le bloc fonction "InitCamTrack".



64613axx

Déclaration

```

FUNCTION_BLOCK InitCamTrack
VAR_INPUT
    Execute:BOOL;
END_VAR
VAR_OUTPUT
    CamTrack : ARRAY[1..8] OF MC_CAMSWITCH_CONFIG_MX;
END_VAR
VAR
    EXE: BOOL ;
    bREdgeexecute: BOOL;
END_VAR

```



Programme

```

EXE:=Execute AND NOT bREdgeexecute;
bREdgeexecute:=Execute;
IF EXE THEN
(*track 1*)
CamTrack[1].Mode :=TRUE;
CamTrack[1].ModeControl :=FALSE;
CamTrack[1].DataSource :=MX_CAMSWITCH_SYSTEMPOS_ENCODER1;
CamTrack[1].DeadTime :=0;
CamTrack[1].TimeFrame :=1;
CamTrack[1].MinModuloValue :=0;
CamTrack[1].MaxModuloValue := 131072;
CamTrack[1].Hysteresis :=0;
CamTrack[1].SelectArea[1]:=1;
CamTrack[1].SelectArea[2]:=2;
CamTrack[1].SelectArea[3]:=3;
CamTrack[1].SelectArea[4]:=4;
CamTrack[1].SelectArea[5]:=5;
CamTrack[1].SelectArea[6]:=6;
CamTrack[1].SelectArea[7]:=23;
CamTrack[1].SelectArea[8]:=24;
CamTrack[1].SelectArea[9]:=25;
CamTrack[1].SelectArea[10]:=26;
CamTrack[1].SelectArea[11]:=27;
CamTrack[1].SelectArea[12]:=28;
CamTrack[1].SelectArea[13]:=29;
CamTrack[1].SelectArea[14]:=30;
CamTrack[1].SelectArea[15]:=31;
CamTrack[1].SelectArea[16]:=32;
(*track 2*)
CamTrack[2].Mode :=TRUE;
CamTrack[2].ModeControl :=FALSE;
CamTrack[2].DataSource :=MX_CAMSWITCH_SYSTEMPOS_ENCODER1;
CamTrack[2].DeadTime :=0;
CamTrack[2].TimeFrame :=1;
CamTrack[2].MinModuloValue :=0;
CamTrack[2].MaxModuloValue := 131072;
CamTrack[2].Hysteresis :=0;
CamTrack[2].SelectArea[1]:=3;
CamTrack[2].SelectArea[2]:=4;
(*track 3*)
CamTrack[3].Mode :=TRUE;
CamTrack[3].ModeControl :=FALSE;
CamTrack[3].DataSource :=MX_CAMSWITCH_SYSTEMPOS_ENCODER1;

```



```
CamTrack[3].DeadTime :=0;
CamTrack[3].TimeFrame :=1;
CamTrack[3].MinModuloValue :=0;
CamTrack[3].MaxModuloValue := 131072;
CamTrack[3].Hysteresis :=0;
CamTrack[3].SelectArea[1]:=5;
CamTrack[3].SelectArea[2]:=6;
(*track 4*)
CamTrack[4].Mode :=TRUE;
CamTrack[4].ModeControl :=FALSE;
CamTrack[4].DataSource :=MX_CAMSWITCH_SYSTEMPOS_ENCODER1;
CamTrack[4].DeadTime :=0;
CamTrack[4].TimeFrame :=1;
CamTrack[4].MinModuloValue :=0;
CamTrack[4].MaxModuloValue := 131072;
CamTrack[4].Hysteresis :=0;
CamTrack[4].SelectArea[1]:=7;
CamTrack[4].SelectArea[2]:=8;
(*track 5*)
CamTrack[5].Mode :=TRUE;
CamTrack[5].ModeControl :=FALSE;
CamTrack[5].DataSource :=MX_CAMSWITCH_SYSTEMPOS_ENCODER1;
CamTrack[5].DeadTime :=0;
CamTrack[5].TimeFrame :=1;
CamTrack[5].MinModuloValue :=0;
CamTrack[5].MaxModuloValue := 131072;
CamTrack[5].Hysteresis :=0;
CamTrack[5].SelectArea[1]:=1;
CamTrack[5].SelectArea[2]:=2;
(*track 6*)
CamTrack[6].Mode :=TRUE;
CamTrack[6].ModeControl :=FALSE;
CamTrack[6].DataSource := MX_CAMSWITCH_MODULOPOS_ENCODER1;
CamTrack[6].DeadTime :=0;
CamTrack[6].TimeFrame :=1;
CamTrack[6].MinModuloValue :=0;
CamTrack[6].MaxModuloValue := 10000;
CamTrack[6].Hysteresis :=0;
CamTrack[6].SelectArea[1]:=3;
CamTrack[6].SelectArea[2]:=4;
(*track 7*)
CamTrack[7].Mode :=TRUE;
CamTrack[7].ModeControl :=FALSE;
```



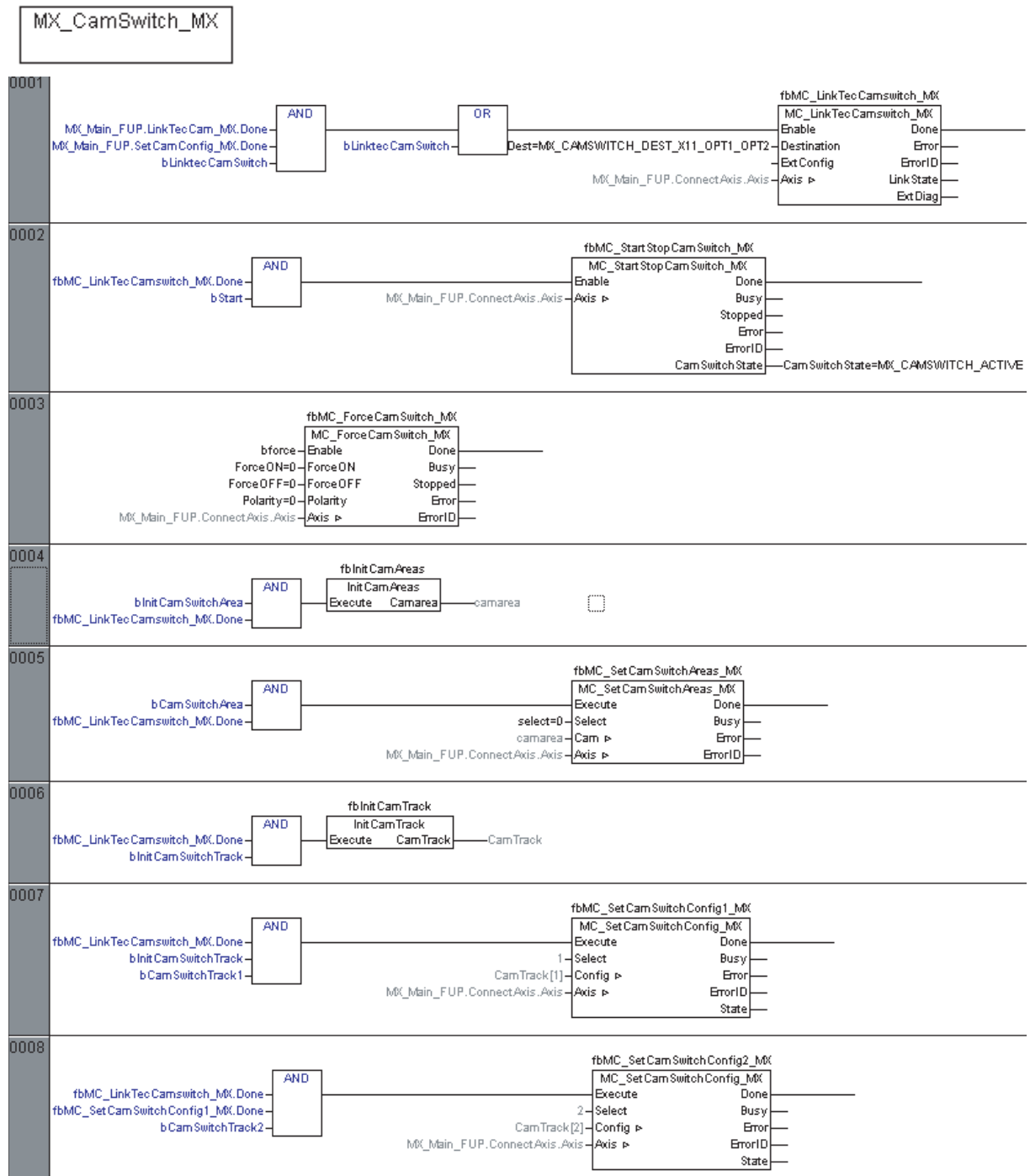
```

CamTrack[7].DataSource := MX_CAMSWITCH_MODULOPOS_ENCODER1;
CamTrack[7].DeadTime :=0;
CamTrack[7].TimeFrame :=1;
CamTrack[7].MinModuloValue :=0;
CamTrack[7].MaxModuloValue := 10000;
CamTrack[7].Hysteresis :=0;
CamTrack[7].SelectArea[1]:=5;
CamTrack[7].SelectArea[2]:=6;
(*track 8*)
CamTrack[8].Mode :=TRUE;
CamTrack[8].ModeControl :=FALSE;
CamTrack[8].DataSource :=MX_CAMSWITCH_USERPOS_ENCODER1;
CamTrack[8].DeadTime :=0;
CamTrack[8].TimeFrame :=1;
CamTrack[8].MinModuloValue :=0;
CamTrack[8].MaxModuloValue := 20000;
CamTrack[8].Hysteresis :=0;
CamTrack[8].SelectArea[1]:=1;
CamTrack[8].SelectArea[2]:=2;
CamTrack[8].SelectArea[3]:=3;
CamTrack[8].SelectArea[4]:=4;
CamTrack[8].SelectArea[5]:=5;
CamTrack[8].SelectArea[6]:=6;
CamTrack[8].SelectArea[7]:=7;
CamTrack[8].SelectArea[8]:=8;
CamTrack[8].SelectArea[9]:=9;
CamTrack[8].SelectArea[10]:=10;
CamTrack[8].SelectArea[11]:=11;
CamTrack[8].SelectArea[12]:=12;
END_IF

```



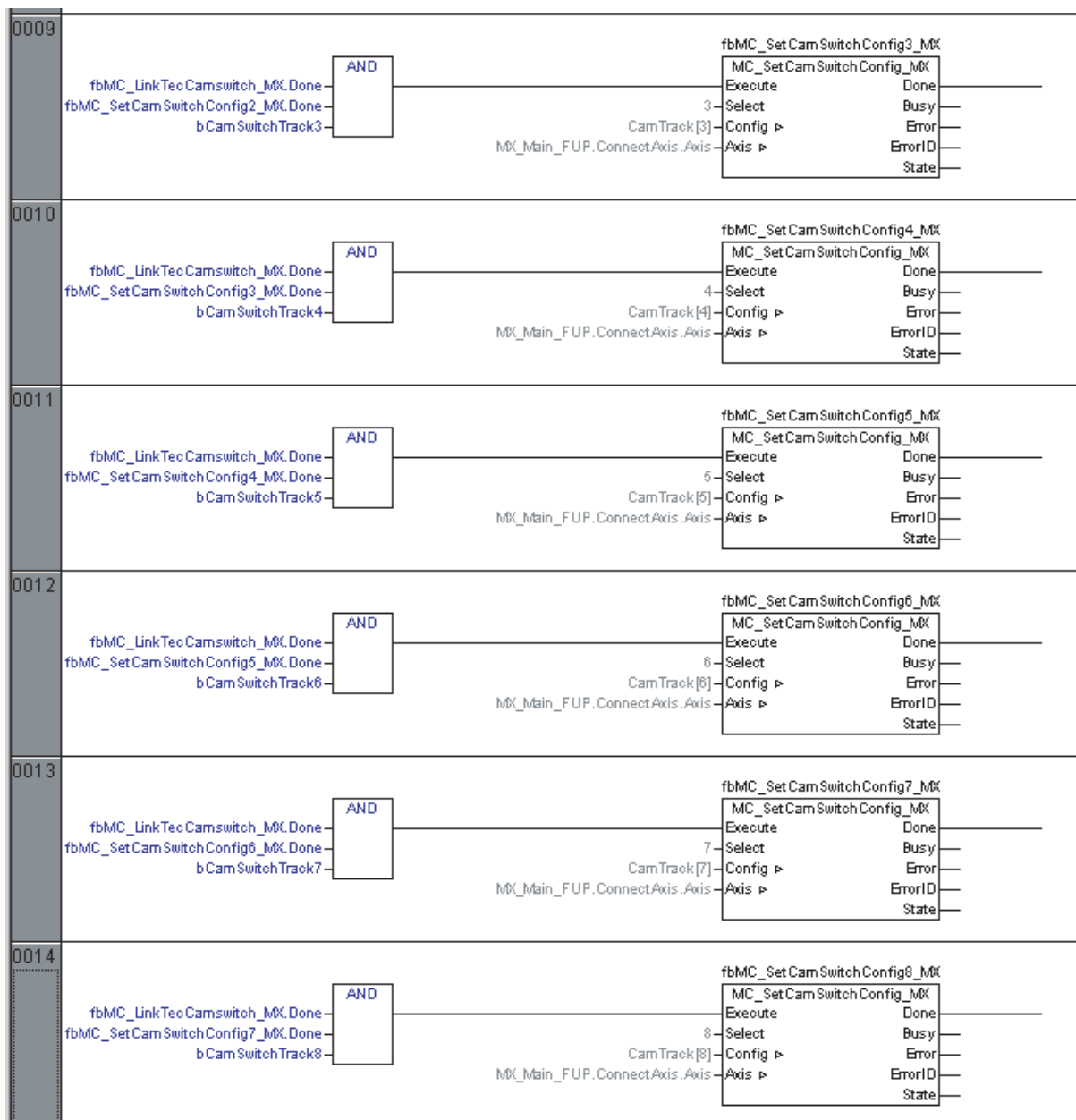

Créer et appeler cycliquement le bloc fonction "MX_CamSwitch_MX".



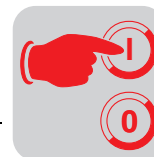
64614axx



MPLCTecCAMSwitch_MX Example



64615axx



4.7 Annexes

4.7.1 Définition des défauts (ErrorID)

Code défaut	Nom du défaut	Description du défaut
FA0010	E_IEC_GENERAL_INVALID_TECHNOLOGIE_OPTION	La fonction technologique demandée n'est pas supportée.
FA020...	General Link Tec Error	
FA0200	E_TEC_GENERAL_MULTIPLE_TECLINKS	Il n'est pas autorisé d'appeler plusieurs fois le bloc fonction LinkTec.
FA0201	E_TEC_GENERAL_INVALID_LINKSTATE	Etat LinkState non admissible pour appel de fonction
FA0202	E_TEC_GENERAL_NOT_LINKED	Le bloc fonction LinkTec-FB n'a pas encore de liaison logique.
FA0203	E_TEC_GENERAL_NOT_INITIALIZED	Fonction technologique pas encore activée
	IEC Messages de défaut généraux	
FA0070	E_IEC_PARAMETER_VALUE_OUT_OF_RANGE	Valeur de paramètre saisie non autorisée
FB0071	E_MDX_MOTIONBLOCK_LOG_ADR_NOT_INITIALIZED	MC_ConnectAxis_MX n'a pas encore attribué d'adresse log.
FB0072	E_MDX_MOTIONBLOCK_INVALID_LOG_ADR	Adresse log. non valide
FB0073	E_MDX_MOTIONBLOCK_INVALID_STATE	Appel de la fonction depuis PLCopenState actuel non autorisé



5 Index

B

Bibliothèque MPLCTecCamMotion_MX	66
<i>Blocs fonction</i>	79
<i>Configuration</i>	58
<i>Logique de détermination</i>	63
<i>Sauvegarde des caractéristiques des</i> <i>courbes</i>	65
<i>Stockage des courbes</i>	64
<i>Types de données</i>	69
Bibliothèque MPLCTecCAMSwitch_MX	128
<i>Blocs fonction</i>	146
<i>Configuration</i>	134
<i>Configuration PDO</i>	145
<i>Logique de détermination</i>	134
<i>Stockage des données</i>	135
<i>Structure DDB</i>	136
<i>Types de données</i>	141
Bibliothèque MPLCTecCamSwitch_MX	140
Bibliothèque MPLCTecGearMotion_MX	8, 11
<i>Aperçu via la bibliothèque</i> <i>MPLCTecGearMotion_MX</i>	18
<i>Blocs fonction</i>	28
<i>Configuration</i>	10
<i>Configuration PDO</i>	25
<i>Logique de détermination</i>	16
<i>Synchronisation par rattrapage en</i> <i>distance</i>	12
<i>Types de données</i>	20
Bibliothèque MPLCTecGearMotion_MX	18
Bref aperçu de la bibliothèque	
MPLCTecCamMotion_MX	66

C

Consignes de sécurité	
<i>Structure des consignes de sécurité</i>	6

D

Différents modes de synchronisation	11
Différents résultats de démarrages de synchronisation et d'offset	14
Domaines d'application	9, 57, 129

E

Exclusion de la responsabilité	6
Exemples MPLCTecGearMotion_MX	
1. <i>Le maître est un codeur virtuel,</i> <i>synchronisation directe par</i>	

<i>rattrapage en vitesse / temps</i>	45
2. <i>MOVIAXIS® maître, synchronisation par</i> <i>interruption DI02</i>	47
3. <i>Offset avec gestion en distance, activé</i> <i>selon une trajectoire maître définie,</i> <i>répétition automatique</i>	50

G

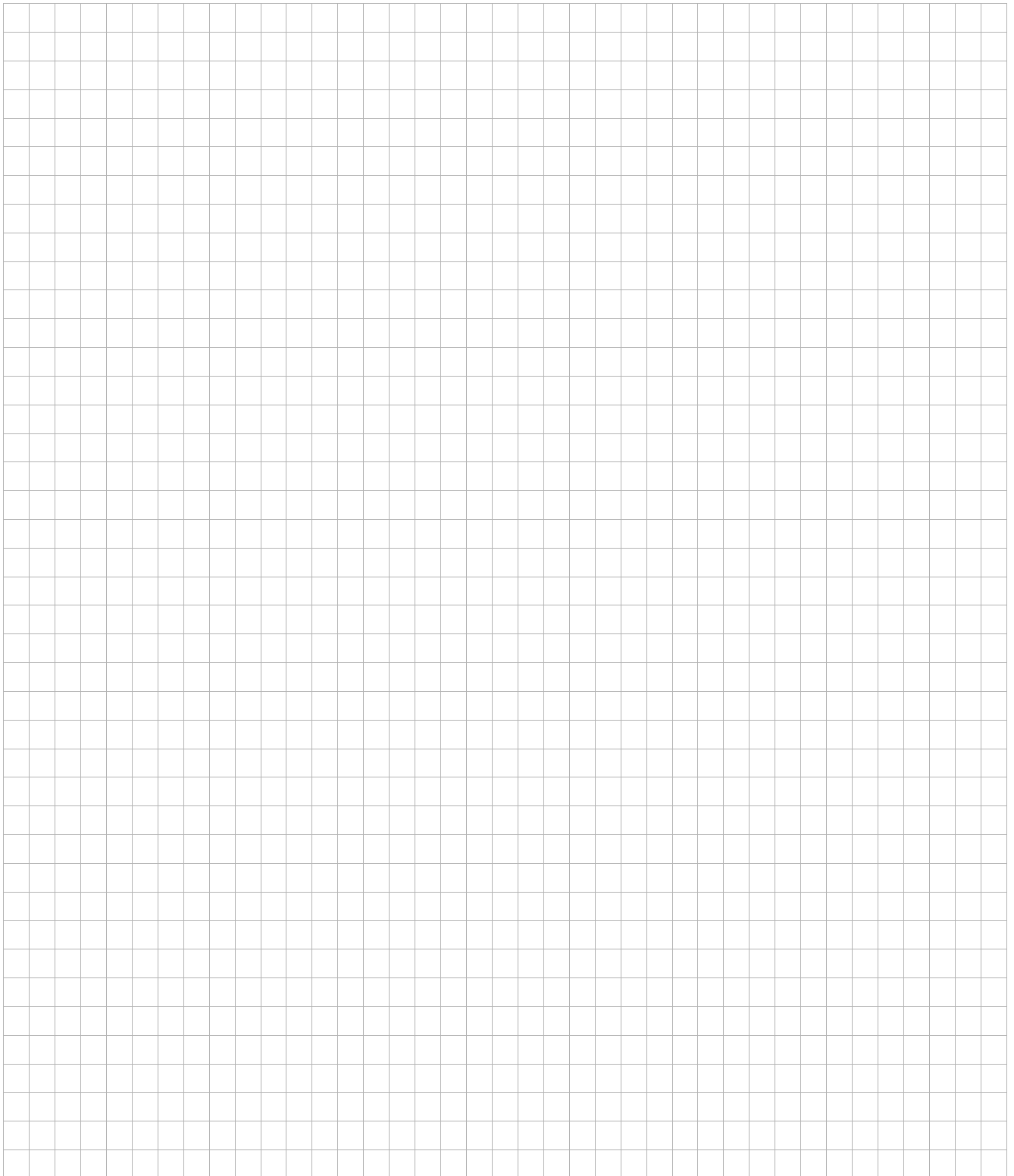
Gestion de l'offset	13
---------------------------	----

M

MPLCTecCamMotion_MX	
<i>Détection des défauts</i>	126
<i>Interruption des blocs fonction</i> <i>(CommandAborted)</i>	126
MPLCTecCamMotion_MX - Exemple	
<i>Came électronique avec codeur virtuel</i> <i>maître</i>	120
MPLCTecCAMSwitch_MX	
<i>Détection des défauts</i>	163
MPLCTecCAMSwitch_MX - Exemple	
<i>Huit voies et 32 cames</i>	154
MPLCTecGearMotion_MX	
<i>Détection des défauts</i>	54
<i>Interruption des tâches des blocs fonction</i> <i>(CommandAborted)</i>	54

R

Recours en cas de défectuosité	6
Remarques générales	6
<i>Exclusion de la responsabilité</i>	6
<i>Recours en cas de défectuosité</i>	6
<i>Structure des consignes de sécurité</i>	6
Remarques importantes	
<i>Applications de levage</i>	7
<i>Consignes de sécurité spécifiques pour les</i> <i>systèmes de bus</i>	7
<i>Fonctions de sécurité</i>	7



En mouvement perpétuel

Des interlocuteurs qui réfléchissent vite et juste, et qui vous accompagnent chaque jour vers l'avenir.

Une assistance après-vente disponible 24 h sur 24 et 365 jours par an.

Des systèmes d'entraînement et de commande qui surmultiplient automatiquement votre capacité d'action.

Un savoir-faire consistant et reconnu dans les secteurs primordiaux de l'industrie moderne.

Une exigence de qualité extrême et des standards élevés qui facilitent le travail au quotidien.



SEW-EURODRIVE
Driving the world

La proximité d'un réseau de bureaux techniques dans votre pays. Et ailleurs aussi.

Des idées innovantes pour pouvoir développer demain les solutions qui feront date après-demain.

Un accès permanent à l'information et aux données via Internet.

SEW
EURODRIVE