

SIEMENS

SINUMERIK

SINUMERIK ONE Actions synchrones

Description fonctionnelle

Introduction

1

Consignes de sécurité
élémentaires

2

Description succincte

3

Description détaillée

4

Exemples

5

Annexes

A

Valable pour

commande
SINUMERIK ONE

Logiciel CNC version 6.23

01/2024

A5E48053843D AH

Mentions légales

Signalétique d'avertissement

Ce manuel donne des consignes que vous devez respecter pour votre propre sécurité et pour éviter des dommages matériels. Les avertissements servant à votre sécurité personnelle sont accompagnés d'un triangle de danger, les avertissements concernant uniquement des dommages matériels sont dépourvus de ce triangle. Les avertissements sont représentés ci-après par ordre décroissant de niveau de risque.

DANGER

signifie que la non-application des mesures de sécurité appropriées entraîne la mort ou des blessures graves.
--

ATTENTION
--

signifie que la non-application des mesures de sécurité appropriées peut entraîner la mort ou des blessures graves.
--

PRUDENCE

signifie que la non-application des mesures de sécurité appropriées peut entraîner des blessures légères.

IMPORTANT

signifie que la non-application des mesures de sécurité appropriées peut entraîner un dommage matériel.

En présence de plusieurs niveaux de risque, c'est toujours l'avertissement correspondant au niveau le plus élevé qui est reproduit. Si un avertissement avec triangle de danger prévient des risques de dommages corporels, le même avertissement peut aussi contenir un avis de mise en garde contre des dommages matériels.

Personnes qualifiées

L'appareil/le système décrit dans cette documentation ne doit être manipulé que par du **personnel qualifié** pour chaque tâche spécifique. La documentation relative à cette tâche doit être observée, en particulier les consignes de sécurité et avertissements. Les personnes qualifiées sont, en raison de leur formation et de leur expérience, en mesure de reconnaître les risques liés au maniement de ce produit / système et de les éviter.

Utilisation des produits Siemens conforme à leur destination

Tenez compte des points suivants:

ATTENTION
--

Les produits Siemens ne doivent être utilisés que pour les cas d'application prévus dans le catalogue et dans la documentation technique correspondante. S'ils sont utilisés en liaison avec des produits et composants d'autres marques, ceux-ci doivent être recommandés ou agréés par Siemens. Le fonctionnement correct et sûr des produits suppose un transport, un entreposage, une mise en place, un montage, une mise en service, une utilisation et une maintenance dans les règles de l'art. Il faut respecter les conditions d'environnement admissibles ainsi que les indications dans les documentations afférentes.

Marques de fabrique

Toutes les désignations repérées par ® sont des marques déposées de Siemens Aktiengesellschaft. Les autres désignations dans ce document peuvent être des marques dont l'utilisation par des tiers à leurs propres fins peut enfreindre les droits de leurs propriétaires respectifs.

Exclusion de responsabilité

Nous avons vérifié la conformité du contenu du présent document avec le matériel et le logiciel qui y sont décrits. Ne pouvant toutefois exclure toute divergence, nous ne pouvons pas nous porter garants de la conformité intégrale. Si l'usage de ce manuel devait révéler des erreurs, nous en tiendrons compte et apporterons les corrections nécessaires dès la prochaine édition.

Sommaire

1	Introduction	7
1.1	À propos de SINUMERIK	7
1.2	À propos de cette documentation.....	7
1.3	Documentation sur Internet	10
1.3.1	Vue d'ensemble de la documentation SINUMERIK ONE	10
1.3.2	Vue d'ensemble de la documentation pour les éléments de conduite SINUMERIK	10
1.4	Remarques concernant la documentation technique.....	11
1.5	Documentation mySupport.....	11
1.6	S.A.V. et assistance	12
1.7	Utilisation de OpenSSL	13
1.8	Respect du règlement général sur la protection des données	14
2	Consignes de sécurité élémentaires.....	15
2.1	Consignes de sécurité générales.....	15
2.2	Garantie et responsabilité pour les exemples d'application	15
2.3	Note relative à la cybersécurité.....	15
3	Description succincte.....	17
4	Description détaillée.....	19
4.1	Définition d'une action synchrone	19
4.2	Composantes des actions synchrones.....	20
4.2.1	Validité, numéros d'identification (ID, IDS)	20
4.2.2	Fréquence (WHENEVER, FROM, WHEN, EVERY)	21
4.2.3	Instruction G (condition)	22
4.2.4	Condition	23
4.2.5	Instruction G (action).....	23
4.2.6	Actions lorsque la condition est remplie (DO)	24
4.2.7	Actions lorsque la condition n'est pas remplie (ELSE)	24
4.3	Variable système pour actions synchrones.....	25
4.3.1	Lecture et écriture.....	26
4.3.2	Opérateurs et fonctions de calcul	26
4.3.3	Conversions de type.....	29
4.3.4	Mémentos/compteurs (\$AC_MARKER)	30
4.3.5	Paramètre (\$AC_PARAM).....	31
4.3.6	Paramètres R (\$R)	32
4.3.7	Paramètres machine et données de réglage (\$\$M, \$\$S).....	33
4.3.8	Temporisation (\$AC_TIMER)	34
4.3.9	Variables FIFO (\$AC_FIFO).....	36
4.3.10	Angle de la tangente à la trajectoire (\$AC_TANEB)	41
4.3.11	Correction (\$A...OVR).....	41

4.3.12	Évaluation de l'utilisation (\$AN_IPO..., \$AN/AC_SYNC..., \$AN_SERVO).....	43
4.3.13	Limitation de la zone de travail (\$SA_WORKAREA....)	46
4.3.14	Positions et temps des cames logicielles (\$\$SN_SW_CAM....)	46
4.3.15	Évaluation de longueur de déplacement / maintenance de la machine (\$AA_TRAVEL..., \$AA_JERK....)	47
4.3.16	Coefficients/paramètres de polynôme (\$AC_FCT....)	49
4.3.17	Déplacements forcés (\$AA_OFF).....	51
4.3.18	Correction de longueur d'outil en ligne (\$AA_TOFF).....	54
4.3.19	Bloc actuel dans l'interpolateur (\$AC_BLOCKTYPE, \$AC_BLOCKTYPEINFO, \$AC_SPLITBLOCK).....	58
4.3.20	Initialisation de variables de tableau (SET, REP).....	60
4.3.21	Variables système spécifiques à la rectification (\$AC_IN_KEY_G....)	61
4.3.22	État du blocage d'action synchrone (\$AC_SYNA_STATE).....	65
4.4	Variables définies par l'utilisateur pour actions synchrones.....	66
4.5	Éléments de langage pour les actions synchrones et les cycles technologiques.....	68
4.6	Éléments de langage pour cycles technologiques uniquement.....	73
4.7	Actions synchrones.....	74
4.7.1	Sortie des fonctions auxiliaires M, S et H en direction de l'AP.....	74
4.7.2	Lecture et écriture de variables système.....	75
4.7.3	Évaluation du polynôme (SYNFCT).....	75
4.7.4	Correction d'outil en ligne (FTOC).....	81
4.7.5	Blocage de lecture programmé (RDISABLE).....	83
4.7.6	Annulation de l'arrêt du prétraitement (STOPREOF).....	84
4.7.7	Suppression de la distance restant à parcourir (DELDTG).....	85
4.7.8	Déplacement d'axes en position (POS).....	86
4.7.9	Réglage du système de mesure (G70, G71, G700, G710).....	90
4.7.10	Position dans la zone de référence prescrite (POSRANGE).....	92
4.7.11	Déplacement d'axes, sans fin (MOV).....	93
4.7.12	Avance axiale (FA).....	93
4.7.13	Permutation d'axe (GET, RELEASE, AXTOCHAN).....	94
4.7.14	Déplacement de broches (M, S, SPOS).....	100
4.7.15	Forçage de valeurs réelles avec perte de l'état de référencement (PRESETON).....	101
4.7.16	Forçage de valeurs réelles sans perte de l'état de référencement (PRESETONS).....	107
4.7.17	Couplages (CP..., LEAD..., TRAIL..., CTAB....)	113
4.7.18	Mesure (MEAWA, MEAC).....	117
4.7.19	Accostage d'une butée (FXS, FXST, FXSW, FOCON, FOCOF, FOC).....	120
4.7.20	Synchronisation de canal (SETM, CLEARM).....	122
4.7.21	Réaction aux erreurs spécifique à l'utilisateur (SETAL).....	123
4.7.22	Interruption du niveau de sous-programme actuel (CANCELSUB).....	124
4.8	Cycles technologiques.....	125
4.8.1	Généralités.....	125
4.8.2	Mode d'exécution (ICYCON, ICYCOF).....	126
4.8.3	Définitions (DEF, DEFINE).....	128
4.8.4	Transfert de paramètres.....	128
4.8.5	Variable de contexte (\$P_TECCYCLE).....	128
4.9	Actions synchrones protégées.....	129
4.10	Coordination par programme pièce et action synchrone (LOCK, UNLOCK, CANCEL).....	131
4.11	Coordination via AP.....	131

4.12	Configuration.....	132
4.13	Comportement de la commande dans des états de fonctionnement donnés	133
4.13.1	Power On.....	133
4.13.2	Reset CN	134
4.13.3	Arrêt CN	134
4.13.4	Changement de mode de fonctionnement	135
4.13.5	Fin du programme	136
4.13.6	Recherche de bloc.....	136
4.13.7	Interruption du programme par un ASUP.....	137
4.13.8	REPOS.....	138
4.13.9	Comportement en cas d'alarmes	138
4.14	Diagnostic	139
4.14.1	Afficher l'état des actions synchrones	139
5	Exemples.....	141
5.1	Exemples de conditions d'actions synchrones	141
5.2	Écriture et lecture de SD/MD d'actions synchrones	141
5.3	Exemples de régulation AC.....	143
5.3.1	Régulation de la distance avec limite supérieure variable	143
5.3.2	Régulation de l'avance	144
5.3.3	Réguler la vitesse en fonction de la trajectoire normalisée.....	145
5.4	Surveillance d'une distance de sécurité entre deux axes.....	146
5.5	Enregistrer les temps d'exécution dans le paramètre R	147
5.6	"Centrer" avec mesure continue.....	148
5.7	Couplages d'axes par actions synchrones.....	150
5.7.1	Couplage sur axe pilote	150
5.7.2	Rectification non circulaire par couplage par valeur pilote	152
5.7.3	Tronçonnage au vol	155
5.8	Cycles technologiques de positionnement de la broche	156
5.9	Actions synchrones dans le domaine WZW/BAZ.....	158
A	Annexes	163
A.1	Liste des abréviations.....	163
	Index.....	169

Introduction

1.1 À propos de SINUMERIK

Des machines-outils courantes CNC simples aux machines modulaires les plus sophistiquées, en passant par les machines standardisées – les commandes CNC SINUMERIK offrent la solution idéale pour chaque machine. Qu'il s'agisse de fabrication de pièces uniques ou de fabrication de masse, de pièces simples ou complexes – SINUMERIK est la solution d'automatisation homogène hautement productive pour tous les secteurs de production – de la fabrication de prototypes et d'outils jusqu'à la fabrication en grandes séries, en passant par l'usinage de moules.

Pour plus d'informations, consulter le site Internet relatif à SINUMERIK (<https://www.siemens.com/sinumerik>).

1.2 À propos de cette documentation

La présente documentation fait partie du groupe des descriptions fonctionnelles SINUMERIK.

Descriptions fonctionnelles SINUMERIK

Les descriptions fonctionnelles SINUMERIK décrivent les fonctions CN d'une commande SINUMERIK.

Les concepteurs, les technologues, les techniciens de mise en service et les programmeurs constituent le groupe cible.

Chaque description fonctionnelle traite un domaine précis et contient toutes les descriptions des fonctions qui font partie de ce domaine.

Les descriptions fonctionnelles disponibles pour votre commande SINUMERIK et les domaines traités par une description fonctionnelle sont indiqués dans le tableau suivant :

Description fonctionnelle	Domaine
Fonctions de base	Fonctions de base d'une commande CNC
Axes et broches	Fonctions d'axes et de broches, couplages d'axes
Transformations	Fonctions de transformation
Surveillance et compensation	Fonctions de surveillance d'axes et de compensation, prévention des collisions
Outils	Fonctions de sélection, de correction et de surveillance d'outils
Actions synchrones	Fonctionnalité des actions synchrones
AP	Structure et fonctions de l'AP
Gestion des outils	Fonctionnement, mise en service et programmation de la gestion des outils
Technologies	Fonctionnalités technologiques étendues

Sommaire

Une vue d'ensemble des principaux chapitres des descriptions des fonctions que contient une description fonctionnelle est disponible sur la page de titre.

Validité

La page de titre contient aussi toutes les informations sur la validité d'un document, c'est-à-dire les commandes SINUMERIK et les versions logicielles auxquelles cette édition de la description fonctionnelle s'applique.

Données système

Dans les descriptions des fonctions, les données système ayant une importance pour une fonction (paramètres machine, données de réglage, variables système, signaux d'interface et alarmes) ne sont décrites que dans la mesure qui est indispensable à la compréhension de la fonction. Vous trouverez des informations détaillées concernant ces données dans les tables de paramètres correspondantes et, pour les alarmes, dans le manuel de diagnostic.

Basic Program Plus et Basic Program

Deux types différents de programme de base sont disponibles dans TIA Portal pour la programmation de la fonctionnalité AP.

- Le programme Basic Program Plus travaille exclusivement avec un adressage symbolique. La fonctionnalité est réalisée à l'aide de types de données et de blocs, fonctions et instructions AP. Le Basic Program Plus est uniquement disponible pour la SINUMERIK ONE.
- Le Basic Program travaille aussi bien avec un adressage symbolique qu'avec un adressage absolu. Les programmes sont ainsi en grande partie compatibles avec des programmes existants du SIMATIC Manager. Le Basic Program est disponible pour les commandes ONE et MC.

Il n'est pas possible d'utiliser conjointement Basic Program Plus et Basic Program.

Blocs AP et types de données

Les descriptions des fonctions se réfèrent souvent à des signaux et à des fonctions AP. Dans le texte, il s'agit en premier lieu des blocs et des types de données du Basic Program Plus, qui n'est disponible que pour la SINUMERIK ONE.

Vous trouverez des informations détaillées dans la description fonctionnelle AP de la commande concernée.

Signaux d'interface

Pour l'adressage des signaux d'interface, c'est l'adressage symbolique du Basic Program Plus qui est utilisé en premier lieu. Pour en déduire les signaux pour le Basic Program, des tableaux sont fournis à la fin de chaque section avec les signaux PLC utilisés pour les différentes directions de signal, comme le montre l'exemple suivant :

CN → AP

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31,DBX60.4
...		

AP → CN

Basic Program Plus	Basic Program	
<Axis>.basic.out.refPointApproachCam	LBP_Axis*.A_DelayRef	DB31,DBX12.7
...		

Remarque**Capacités fonctionnelles**

Dans le Basic Program Plus, les axes, canaux, etc. sont représentés par des instances des objets correspondants.

En cas d'adressage absolu dans le Basic Program, on part du nombre maximal absolu des composants suivants : groupes de modes de fonctionnement (DB11), canaux (DB21, ...), axes/broches (DB31, ...)

Version standard

La présente documentation décrit la fonctionnalité de la version standard du système. Elle peut différer de l'étendue des fonctionnalités du système livré. Pour toute précision concernant les fonctionnalités du système livré, se reporter exclusivement aux documents de commande.

Le système peut comporter des fonctions opérationnelles supplémentaires qui ne sont pas mentionnées dans cette documentation. Le client ne peut toutefois pas faire valoir de droit en liaison avec ces fonctions, que ce soit dans le cas de matériels neufs ou dans le cadre d'interventions du service après-vente.

Pour des raisons de clarté, la présente documentation peut ne pas contenir toutes les informations détaillées concernant toutes les variantes du produit. En outre, elle ne peut pas tenir compte de tous les cas possibles d'installation, d'exploitation ou de maintenance.

Les options complémentaires ou les modifications apportées par le constructeur de machines au produit sont documentées par celui-ci.

Pages web non Siemens

Ce document peut contenir des liens vers des pages web non Siemens. Siemens décline toute responsabilité quant au contenu de ces pages web et ne considère pas comme siennes ces pages et leur contenu. Siemens ne contrôle pas les informations accessibles par ces pages web et n'est pas non plus responsable du contenu et des informations qui y sont mis à disposition, leur utilisation étant aux risques et périls de l'utilisateur.

1.3 Documentation sur Internet

1.3.1 Vue d'ensemble de la documentation SINUMERIK ONE

Une documentation détaillée relative aux fonctions de SINUMERIK ONE à partir de la version 6.13 est disponible sous Vue d'ensemble de la documentation SINUMERIK ONE (<https://support.industry.siemens.com/cs/ww/en/view/109768483>).



Il est possible d'afficher les documents ou de les télécharger aux formats PDF et HTML5.

La documentation est divisée en plusieurs catégories comme suit :

- Utilisateur : Commande
- Utilisateur : Programmation
- Constructeur/S.A.V. : Fonctions
- Constructeur/S.A.V. : Matériel
- Constructeur/S.A.V. : Configuration/mise en service
- Constructeur/S.A.V. : Safety Integrated
- Information et formation
- Constructeur/S.A.V. : SINAMICS

1.3.2 Vue d'ensemble de la documentation pour les éléments de conduite SINUMERIK

Une documentation détaillée relative aux éléments de conduite SINUMERIK est disponible sous Vue d'ensemble de la documentation pour les éléments de conduite SINUMERIK (<https://support.industry.siemens.com/cs/document/109783841/technische-dokumentation-zu-sinumerik-bedienkomponenten?dti=0&lc=en-WW>).

Il est possible d'afficher les documents ou de les télécharger aux formats PDF et HTML5.

La documentation est divisée en plusieurs catégories comme suit :

- Tableaux de commande
- Tableaux de commande de machine
- Machine Push Button Panel
- Unité portable/Mini-consoles
- Autres éléments de conduite

Une vue d'ensemble des documents, des contributions et des liens les plus importants relatifs à SINUMERIK se trouve sous Vue d'ensemble/page thématique SINUMERIK (<https://support.industry.siemens.com/cs/document/109766201/sinumerik-an-overview-of-the-most-important-documents-and-links?dti=0&lc=en-WW>).

1.4 Remarques concernant la documentation technique

En cas de questions, suggestions ou corrections relatives à la documentation technique publiée dans Siemens Industry Online Support, utiliser le lien "Donner un avis" à la fin d'une contribution.

1.5 Documentation mySupport

Le système "Documentation mySupport" sur Internet permet à un utilisateur de composer sa propre documentation à partir des contenus Siemens et de l'adapter à sa documentation machine.

Démarrer l'application via le panneau "Ma documentation" sur la page SiePortal "mySupport Links und Tools" (<https://support.industry.siemens.com/cs/ww/fr/my>):

mySupport - Liens et outils



L'exportation du manuel configuré est possible au format RTF, PDF ou XML.

Remarque

Les contenus Siemens qui prennent en charge l'application Documentation mySupport sont reconnaissables à la présence du lien "Configurer".

1.6 S.A.V. et assistance

Assistance produit

Pour plus d'informations sur le produit, voir sur Internet :

Assistance produit (<https://support.industry.siemens.com/cs/ww/fr/>)

Sous cette adresse figurent les éléments suivants :

- Informations produit actuelles (Informations sur le produit)
- FAQ (Foire aux Questions)
- Manuels
- Téléchargements
- Lettre d'information (newsletter) avec les dernières actualités sur vos produits
- Forum à destination des utilisateurs et des spécialistes pour un échange d'informations et d'expérience à l'échelle mondiale
- Interlocuteurs sur place via notre base de données d'interlocuteurs (→ "Contact")
- Informations sur le service après-vente, les réparations, les pièces de rechange et bien plus encore (→ "Services")

Assistance technique

Les numéros de téléphones spécifiques à chaque pays pour l'assistance technique sont disponibles à cette adresse (<https://support.industry.siemens.com/cs/ww/fr/sc/4868>) dans la zone "Contact".

Pour poser une question technique, utiliser le formulaire en ligne dans la zone de demande d'assistance "Support Request".

Formation

L'adresse (<https://www.siemens.com/sitrain>) suivante fournit des informations sur SITRAIN. SITRAIN propose une offre de formations pour les produits, systèmes et solutions d'entraînement et d'automatisation Siemens.

Assistance Siemens pour les déplacements



L'application primée "Industry Online Support" permet d'accéder à tout moment et en tout lieu à plus de 300 000 documents relatifs aux produits Siemens Industry. L'application assiste les clients notamment dans les domaines d'utilisation suivants :

- Résolution de problèmes lors de la réalisation d'un projet
- Dépannage en cas de défauts
- Extension ou replanification d'une installation

En outre, elle donne accès à un forum technique et à d'autres contributions créées spécialement pour nos clients par nos experts :

- FAQ
- Exemples d'application
- Manuels
- Certificats
- Informations sur les produits et bien plus encore

L'application "Industry Online Support" est disponible pour Apple iOS et Android.

1.7 Utilisation de OpenSSL

Ce produit peut contenir les logiciels suivants :

- Logiciel développé par le projet OpenSSL pour une utilisation dans la boîte à outils OpenSSL
- Logiciel cryptographique créé par Eric Young
- Logiciel développé par Eric Young

Pour plus d'informations, voir sur Internet :

- OpenSSL (<https://www.openssl.org>)
- Cryptsoft (<https://www.cryptsoft.com>)

1.8 Respect du règlement général sur la protection des données

Siemens respecte les principes de la protection des données, en particulier les règles de limitation des données (protection de la vie privée dès la conception).

Pour ce produit, cela signifie que :

Le produit ne traite et n'enregistre aucune donnée à caractère personnel mais uniquement des données techniques fonctionnelles (p. ex. horodatage). Si l'utilisateur relie ces données à d'autres données (p. ex. plannings d'équipes) ou s'il enregistre des données à caractère personnel sur le même support (p. ex. disque dur) et crée par là même un lien avec des personnes, il est tenu de garantir lui-même le respect des dispositions légales en matière de protection des données.

Consignes de sécurité élémentaires

2.1 Consignes de sécurité générales

 ATTENTION
Le non respect des consignes de sécurité et le manque de prise en compte des risques résiduels peuvent entraîner la mort
Le non respect des consignes de sécurité et des remarques relatives aux risques résiduels dans la documentation du matériel peut conduire à des accidents susceptibles d'entraîner la mort ou de causer des blessures graves. • Respecter les consignes de sécurité figurant dans la documentation du matériel. • Tenir compte des risques résiduels pour l'évaluation des risques.

 ATTENTION
Danger de mort lié à des dysfonctionnements de la machine suite à un paramétrage incorrect ou modifié
Un paramétrage incorrect ou modifié peut entraîner des dysfonctionnements sur les machines, susceptibles de provoquer des blessures, voire la mort. • Protéger le paramétrage contre l'accès non autorisé. • Prendre les mesures appropriées pour palier aux défauts éventuels (p. ex. un arrêt ou une coupure d'urgence).

2.2 Garantie et responsabilité pour les exemples d'application

Les exemples d'application sont sans engagement et n'ont aucune prétention d'exhaustivité concernant la configuration, les équipements et les éventualités de toutes sortes. Les exemples d'application ne constituent pas des solutions client spécifiques, mais ont uniquement pour objet d'apporter une aide dans la résolution de problèmes typiques.

L'utilisateur est seul responsable de la mise en œuvre des produits selon les règles de l'art. Les exemples d'application ne vous dispensent pas des obligations de précaution lors de l'utilisation, de l'installation, de l'exploitation et de la maintenance.

2.3 Note relative à la cybersécurité

Siemens commercialise des produits et solutions comprenant des fonctions de cybersécurité industrielle qui contribuent à une exploitation sûre des installations, systèmes, machines et réseaux.

Pour garantir la sécurité des installations, systèmes, machines et réseaux contre les cybermenaces, il est nécessaire de mettre en œuvre - et de maintenir en permanence - un concept de cybersécurité industrielle global et de pointe. Les produits et solutions de Siemens constituent un des éléments de ce concept.

Il incombe aux clients d'empêcher tout accès non autorisé à ses installations, systèmes, machines et réseaux. Ces systèmes, machines et composants doivent uniquement être connectés au réseau d'entreprise ou à Internet dans la mesure où cela est nécessaire et seulement si des mesures de protection adéquates (ex : pare-feu et/ou segmentation du réseau) ont été prises.

Pour plus d'informations à propos des mesures de protection pouvant être mises en œuvre dans le domaine de la cybersécurité industrielle, rendez-vous sur <https://www.siemens.com/cybersecurity-industry>.

Les produits et solutions Siemens font l'objet de développements continus pour qu'ils soient encore plus sûrs. Siemens recommande vivement d'effectuer les mises à jour dès que celles-ci sont disponibles et d'utiliser la dernière version des produits. L'utilisation de versions qui ne sont plus prises en charge et la non-application des dernières mises à jour peut augmenter le risque de cybermenaces de nos clients.

Pour être informé des mises à jour produit, abonnez-vous au flux RSS Siemens Industrial Cybersecurity à l'adresse suivante <https://www.siemens.com/cert>.

Plus d'informations, voir sur Internet :

Manuel de configuration Industrial Security (<https://support.industry.siemens.com/cs/ww/fr/view/108862708/en>)



ATTENTION

États de fonctionnement non sûrs suite à une manipulation du logiciel

Les manipulations des logiciels, p. ex. les virus, chevaux de Troie ou vers, peuvent provoquer des états de fonctionnement non sûrs de l'installation, susceptibles de causer la mort, des blessures graves et des dommages matériels.

- Les logiciels doivent être maintenus à jour.
- Intégrer les composants d'entraînement et d'automatisation dans un concept global de cybersécurité industrielle (Industrial Cybersecurity) de l'installation ou de la machine selon l'état actuel de la technique.
- Tenir compte de tous les produits mis en œuvre dans le concept global de cybersécurité industrielle.
- Il convient de protéger les données stockées sur les supports de mémoire amovibles contre les logiciels nuisibles avec les mesures de protection appropriées, par exemple avec un antivirus.
- Contrôler à l'issue de la mise en service toutes les fonctions relatives à la cybersécurité.

Description succincte

Généralités

Une action synchrone se compose d'une série d'instructions associées au sein d'un programme pièce, qui sont exécutées cycliquement dans la période d'appel de l'interpolateur, de manière synchrone par rapport aux blocs d'usinage.

Une action synchrone se divise essentiellement en deux parties, la partie condition facultative et la partie action obligatoire. La partie condition permet de rendre le moment d'exécution des actions dépendant d'un état donné du système. Les conditions sont évaluées de manière cyclique dans la période d'appel de l'interpolateur. Les actions constituent par conséquent une réaction à des états du système pouvant être définis par l'utilisateur. Leur exécution n'est donc pas tributaire des limites de blocs.

En outre, la validité de l'action synchrone (non modale, modale ou statique) et la fréquence de l'exécution des actions (unique, répétée) peuvent être définies.

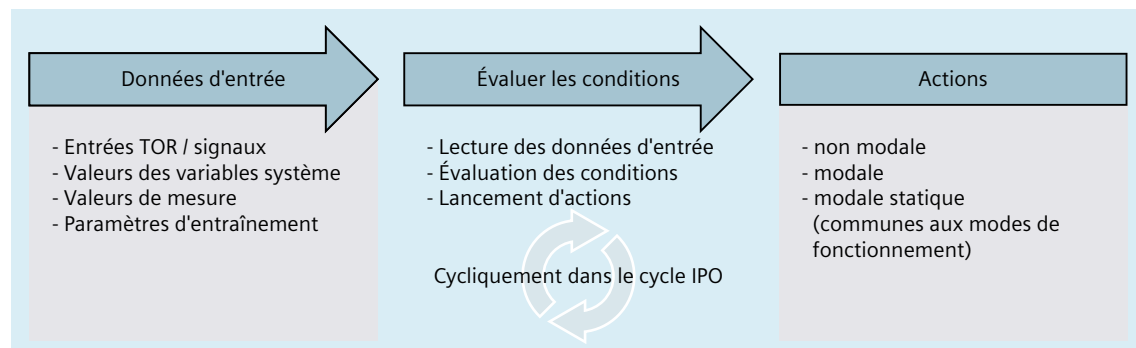
Exemples d'actions admissibles

- Envoi de fonctions auxiliaires à l'AP
- Lecture et écriture de variables en temps réel
- Déplacement d'axes de positionnement
- Activation de procédures synchrones comme :
 - Blocage de l'introduction via l'interface
 - Effacement de la distance restant à parcourir
 - Terminer l'arrêt du prétraitement des blocs
- Activation de cycles technologiques
- Calcul de valeurs fonctionnelles
- Corrections d'outil
- Activer/désactiver les couplages
- Mesure
- Blocage/déblocage d'actions synchrones

Exemples d'actions inadmissibles

- Déplacement d'axes à interpolation

Action synchrone schématique



Description détaillée

4.1 Définition d'une action synchrone

Une action synchrone se définit dans un bloc de programme pièce. Aucune autre instruction que les constituants de l'action synchrone ne doit être programmée dans ce bloc.

Composants d'une action synchrone

Une action synchrone se compose des éléments suivants :

Validité, n° d'identification (Page 20) (en option)	Partie condition (en option)			Partie action lorsque la condition est remplie			Partie action lorsque la condition n'est pas remplie (en option)		
	Fréquence (Page 21)	Instruction G (Page 22) (en option)	Condition (Page 23)	Mot-clé	Instruction G (Page 23) (en option)	Actions (Page 24)	Mot-clé	Instruction G (Page 23) (en option)	Actions (Page 24)
--- 1) ID=<n°> IDS=<n°>	--- 1) WHENEVER FROM WHEN EVERY	G...	Expression n logique	DO	G...	Action 1 ... Action n	ELSE	G...	Action 1 ... Action n

1) non programmée

Syntaxe

Exemples :

- DO <action 1...n>
- <fréquence> [<fonction G>] <condition> DO <action 1...n>
- ID=<n°><fréquence> [<fonction G>] <condition> DO <action 1...n>
- IDS=<n°><fréquence> [<fonction G>] <condition> DO <action 1...n>
- IDS=<n°> <fréquence> [<fonction G>] <condition> DO <action 1...n>
ELSE <action 1...n>

4.2 Composantes des actions synchrones

4.2.1 Validité, numéros d'identification (ID, IDS)

Validité

La validité définit où et quand l'action synchrone est exécutée :

Validité	Signification
--- 1)	Action synchrone à effet non modal Une action synchrone à effet non modal s'applique : • tant que le bloc principal suivant l'action synchrone est actif • uniquement en mode AUTOMATIQUE Exemple : L'action synchrone à partir de N10 s'applique tant que N20 est actif. <pre>N10 WHEN \$A_IN[1]==TRUE DO \$A_OUTA[1]=10 N20 G90 F1000 X100</pre>
ID=<n° ID>	Action synchrone modale Une action synchrone à effet modal s'applique : • jusqu'à la fin du programme pièce • uniquement en mode AUTOMATIQUE Plage de valeurs : voir sous "Numéro d'identification" > "Plage de valeurs" Exemple : <pre>N20 ID=1 EVERY \$A_IN[1]==TRUE DO \$A_OUTA[1]=10</pre>
IDS=<n° ID>	Action synchrone statique Une action synchrone statique s'applique : • dans tous les modes de fonctionnement sans limite de temps Plage de valeurs : voir sous "Numéro d'identification" > "Plage de valeurs" Exemple : <pre>N30 IDS=1 EVERY \$A_IN[1]==TRUE DO \$A_OUTA[1]=10</pre>

1) non programmée

Remarque

Actions synchrones statiques

Les actions synchrones statiques (IDS) peuvent être définies dans un ASUP et activées à tout moment en activant l'ASUP via le programme AP utilisateur.

Numéro d'identification ID/IDS

Plage de valeurs

Les numéros d'identification ID/IDS se trouvent dans différentes plages de numéros. Ces dernières sont affectées à différents utilisateurs.

ID / IDS	Utilisateur	Répertoire
1 ... 999	"Domaine général"	Répertoire quelconque
1000 ... 1199	Constructeur de machines	/_N_CMA_DIR
1200 ... 1399	Siemens	/_N_CST_DIR

Parallélisation

Si plusieurs actions synchrones doivent être actives en parallèle dans un canal, leurs numéros d'identification ID/IDS doivent être différents. Les actions synchrones dont le numéro d'action est identique se remplacent mutuellement dans un canal.

Ordre d'exécution

Les actions synchrones statiques et modales sont exécutées dans l'ordre de leurs numéros d'identification ID/IDS.

Les actions synchrones à effet non modal sont exécutées dans l'ordre de leur programmation, après l'exécution des actions synchrones modales.

Coordination par programmes pièce et actions synchrones

À l'aide de leurs numéros d'identification ID/IDS, les actions synchrones peuvent être coordonnées via des programmes pièce et actions synchrones (Page 131).

Coordination via AP

Les actions synchrones dont le numéro d'identification ID/IDS se trouve dans la plage de 1 à 64 peuvent être coordonnées via l'interface CN/AP à partir du programme utilisateur AP (Page 131).

4.2.2 Fréquence (WHENEVER, FROM, WHEN, EVERY)

La fréquence indique le nombre de fois que la condition doit être demandée et que l'action doit être exécutée lorsque la condition est remplie. La fréquence fait partie intégrante de la condition.

Fréquence	Signification
--- ¹⁾	Si aucune fréquence n'est indiquée, l'action est exécutée de manière cyclique à chaque période d'appel de l'interpolateur.
WHENEVER	Si la condition est remplie, l'action est exécutée de manière cyclique à chaque période d'appel de l'interpolateur.
FROM	Après que la condition a été remplie une fois, l'action est exécutée de manière cyclique à chaque période d'appel de l'interpolateur tant que l'action synchrone est active.

4.2 Composantes des actions synchrones

Fréquence	Signification
WHEN	Si la condition est remplie, l'action est exécutée une fois et la condition n'est plus vérifiée par la suite.
EVERY	Dans les cas suivants, l'action est exécutée une fois : - la condition est déjà remplie au démarrage de l'action synchrone (état : TRUE) - à chaque fois que la condition change d'état pour passer de FALSE à TRUE (front montant)

¹⁾ non programmée

Voir aussi

Cycles technologiques (Page 125)

4.2.3 Instruction G (condition)

Etat initial défini

En fonction du déroulement du programme pièce, les actions synchrones peuvent être exécutées à tout moment si la condition est remplie. Il est donc recommandé de déterminer le système de mesure nécessaire (anglo-saxon ou métrique) d'une action synchrone **avant** la condition et/ou dans la partie action. Une position de départ est ainsi définie pour l'évaluation de la condition et l'exécution de l'action, indépendamment de l'état actuel du programme pièce.

Instructions G

Les instructions G suivantes sont autorisées :

- G70 (cotation en pouces pour indications géométriques (longueurs))
- G71 (cotation métrique pour indications géométriques (longueurs))
- G700 (cotation en pouces pour indications géométriques et technologiques (longueurs, avance))
- G710 (cotation métrique pour indications géométriques et technologiques (longueurs, avance))

Remarque

Dans les actions synchrones, aucune instruction G autre que G70, G71, G700 et G710 n'est autorisée.

Validité

Une instruction G programmée dans la partie condition s'applique également à la partie action si aucune instruction G propre n'a été programmée dans la partie action.

Une instruction G programmée dans la partie action ne s'applique que dans celle-ci.

4.2.4 Condition

L'exécution de l'action peut être rendue dépendante d'une condition remplie. Tant que l'action synchrone est active, la condition est vérifiée de manière cyclique pendant la période d'appel de l'interpolateur. Si aucune condition n'est indiquée, l'action est exécutée de manière cyclique à chaque période d'appel de l'interpolateur.

Toutes les opérations fournissant une valeur de vérité (TRUE/FALSE) comme résultat peuvent être programmées comme une condition :

- Comparaisons de variables système et de constantes
- Comparaisons de variables système et de variables système
- Comparaisons de variables système et de résultats de calcul
- Combinaison de comparaisons par expressions booléennes

Exemples

Comparaisons

Code de programme

```
ID=1 WHENEVER $AA_IM[X] > $AA_IM[Y] DO ...
ID=2 WHENEVER $AA_IM[X] > (10.5 * SIN(45)) DO ...
```

Combinaisons booléennes

Code de programme

```
ID=1 WHENEVER ($A_IN[1]==1) OR ($A_IN[3]==0) DO ...
```

Voir aussi

Lecture et écriture (Page 26)

Exemples de conditions d'actions synchrones (Page 141)

Variable système pour actions synchrones (Page 25)

4.2.5 Instruction G (action)

Etat initial défini

En fonction du déroulement du programme pièce, les actions synchrones peuvent être exécutées à tout moment si la condition est remplie. Il est donc judicieux de déterminer le système de mesure nécessaire (anglo-saxon ou métrique) d'une action synchrone dans la partie action. Une position de départ est ainsi définie pour l'exécution de l'action, indépendamment de l'état actuel du programme pièce.

Instructions G

Les instructions G suivantes sont autorisées :

- G70 (cotation en pouces pour indications géométriques (longueurs))
- G71 (cotation métrique pour indications géométriques (longueurs))
- G700 (cotation en pouces pour indications géométriques et technologiques (longueurs, avance))
- G710 (cotation métrique pour indications géométriques et technologiques (longueurs, avance))

Validité

Une instruction G programmée dans la partie condition s'applique également à la partie action si aucune instruction G propre n'a été programmée dans la partie action.

Une instruction G programmée dans la partie action ne s'applique que dans celle-ci.

4.2.6 Actions lorsque la condition est remplie (DO)

La partie action d'une action synchrone est déclenchée par le mot clé DO.

Dans la partie action, vous pouvez programmer une ou plusieurs actions. Elles sont exécutées si la condition est remplie. Si plusieurs actions sont programmées dans une action synchrone, celles-ci seront exécutées dans la même période d'appel de l'interpolateur.

Exemple :

Si la valeur réelle de l'axe Y est supérieure ou égale à 35,7, la fonction auxiliaire M135 est envoyée à l'AP et la sortie TOR 1 est mise à 1 en même temps.

Code de programme

```
WHEN $AA_IM[Y] >= 35.7 DO M135 $A_OUT[1]=1
```

Cycle technologique

Un cycle technologique peut être appelé comme action (voir chapitre "Cycles technologiques (Page 125)").

4.2.7 Actions lorsque la condition n'est pas remplie (ELSE)

Avec le mot-clé ELSE, toute action synchrone peut être complétée par des actions devant être exécutées lorsque la condition n'est pas remplie. Cela permet le regroupement d'actions synchrones ayant des conditions contradictoires.

Exemple :

```
ID=101 WHENEVER $VA_IM[x] < 100 DO $AC_OVR=100
```

```
ID=102 WHENEVER $VA_IM[x] >= 100 DO $AC_OVR=50
```

La programmation de ELSE permet de regrouper les deux actions synchrones :

```
ID=101 WHENEVER $VA_IM[x] < 100 DO $AC_OVR=100 ELSE $AC_OVR=50
```

Autres propriétés

- Au sein de la branche ELSE, ce sont les mêmes conditions marginales que pour les actions lorsque la condition est remplie (Page 24) qui s'appliquent.
- ELSE est en principe possible pour toutes les conditions. Son utilisation pertinente relève de la responsabilité de l'utilisateur.

Exemple :

```
WHEN $AA_IW[X] > 100 DO $R1=1 ELSE $R1=2  
GO X100 F10
```

Ici, l'expression \$R1=2 est exécutée et le paramètre R est écrit dans chaque période d'appel de l'interpolateur, même lorsque cela n'est pas nécessaire. Cela doit être pris en compte par l'utilisateur.

- L'instruction de langage ELSE peut être utilisée avec toutes les variantes de fréquence. Exemples :
 - À chaque fois que la condition n'est pas remplie, la branche ELSE est exécutée :

```
WHENEVER $AA_IW[X] > 100 DO $R1=1 ELSE $R1=2  
EVERY $AA_IW[X] > 100 DO $R1=1 ELSE $R1=2
```
 - Tant que la condition n'est pas remplie, la branche ELSE est exécutée, puis c'est toujours l'action :

```
FROM $AA_IW[X] > 100 DO $R1=1 ELSE $R1=2
```
 - Tant que la condition n'est pas remplie, la branche ELSE est exécutée, puis c'est l'action, et l'action synchrone se termine :

```
WHEN $AA_IW[X] > 100 DO $R1=1 ELSE $R1=2
```

4.3 Variable système pour actions synchrones

Les variables système de la CN sont énumérées dans les Tables de paramètres "Variables système" avec leurs propriétés respectives. Les variables système qui peuvent être lues ou écrites dans les actions synchrones sont signalées par un "X" dans la ligne correspondante (Read ou Write) de la colonne "AS" (action synchrone).

Remarque

Les variables système qui sont utilisées dans les actions synchrones sont toujours implicitement lues et écrites de façon synchrone à l'exécution.

Plus d'informations

Pour une description détaillée des variables système, voir Tables de paramètres Variables système.

4.3.1 Lecture et écriture

Variables système

Toutes les variables système qui peuvent être utilisées dans les actions synchrones sont lues/écrites pendant l'**exécution des blocs**, à l'exception des éléments listés ci-dessous :

Les variables système pouvant être utilisées dans des actions synchrones sont signalées par un "X" dans la ligne "Read" et/ou "Write" de la colonne "AS" (action synchrone) dans les Tables de paramètres "Variables système".

Informations complémentaires

Tables de paramètres Variables système

Systématique des descripteurs

Les descripteurs des variables système qui sont lues / écrites pendant l'exécution des blocs avec utilisation dans des actions synchrones ont la systématique suivante :

\$A...	Données d'exécution courantes
\$V...	Données servo
\$R...	Paramètres R à lire/écrire pendant l'exécution des blocs
\$\$M...	Paramètres machine à lire/écrire pendant l'exécution des blocs
\$\$S...	Données de réglage à lire/écrire pendant l'exécution des blocs

Exceptions

Les variables suivantes sont également lues/écrites lors du prétraitement des blocs avec utilisation dans des actions synchrones :

- Variables définies par l'utilisateur : LUD, GUD
- Paramètres machine : \$M...
- Données de réglage : \$S...
- Paramètres R : R<numéro> ou R[<indice>]

4.3.2 Opérateurs et fonctions de calcul

Opérateurs

Opérateurs arithmétiques

Les variables systèmes de type RAL ou INT peuvent être traitées avec les opérateurs suivants :

Opérateur	Signification
+	Addition
-	Soustraction

Opérateur	Signification
*	Multiplication
/	- Division en actions synchrones : $\text{INT} / \text{INT} \Rightarrow \text{INT}$ - Divisions en actions synchrones avec résultat de type REAL à l'aide de la fonction ITOR() : $\text{ITOR}(\text{INT}) / \text{ITOR}(\text{INT}) \Rightarrow \text{REAL}$ - Divisions en programmes CN : $\text{INT} / \text{INT} \Rightarrow \text{REAL}$
DIV	Divisions d'entiers : $\text{INT} / \text{INT} \Rightarrow \text{INT}$
MOD	La division modulo, (uniquement type INT) livre le reste d'une division INT Exemple : $3 \text{ MOD } 4 = 3$

Remarque

Seules des variables de même type peuvent être combinées.

Opérateurs relationnels

Opérateur	Signification
==	égal à
<>	différent de
<	inférieur à
>	supérieur à
<=	inférieur ou égal à
>=	supérieur ou égal à

Opérateurs booléens

Opérateur	Signification
NOT	NON
AND	ET
OR	OU
XOR	OU exclusif

Opérateurs logiques sur bits

Opérateur	Signification
B_OR	OU sur bits
B_AND	ET sur bits
B_XOR	OU exclusif sur bits
B_NOT	Négation sur bits

Priorité des opérateurs

Les opérateurs ont les priorités suivantes lors de l'exécution dans l'action synchrone (priorité la plus élevée : 1) :

Prio	Opérateurs	Signification
1	NOT, B_NOT	Négation, négation sur bits
2	*, /, DIV, MOD	Multiplication, division
3	+, -	Addition, soustraction
4	B_AND	ET sur bits
5	B_XOR	OU exclusif sur bits
6	B_OR	OU sur bits
7	AND	ET
8	XOR	OU exclusif
9	OR	OU
10	<<	Concaténation de chaînes de caractères, type de résultat STRING
11	==, <>, <, >, >=, <=	Opérateurs relationnels

Remarque

Dans les expressions à plusieurs opérateurs, il est fortement recommandé de grouper les opérateurs par des parenthèses "(...)" pour s'assurer qu'ils soient traités dans l'ordre souhaité.

Exemple de condition avec une expression à plusieurs opérateurs

Code de programme

```
... WHEN ($AA_IM[X] > VALEUR) AND ($AA_IM[Y] > VALEUR1) DO ...
```

Fonctions de calcul

Opérateur	Signification
SIN()	Sinus
COS()	Cosinus
TAN()	Tangente
ASIN()	Arc sinus
ACOS()	Arc cosinus
ATAN2(,)	Arc tangente 2
SQRT()	racine carrée
ABS()	Valeur absolue
POT()	Fonction de puissance
TRUNC()	Partie entière Réglage de la précision des instructions de comparaison avec TRUNC
ROUND()	Arrondissement à un nombre entier
LN()	Logarithme naturel
EXP()	Fonction exponentielle

Indexation

L'indice d'une variable système de type "Champ de ..." peut comporter une autre variable système. L'indice est alors pris en compte dans le traitement de bloc dans la période d'appel de l'interpolateur.

Exemple

Code de programme

```
... WHEN ... DO $AC_PARAM[ $AC_MARKER[1] ] = 3
```

Limitations

- Une imbrication de l'indexage avec d'autres variables système n'est pas autorisée.
- L'indice ne doit pas comporter de variable de prétraitement. L'exemple suivant n'est donc **pas** autorisé car \$P_EP est une variable de prétraitement :
\$AC_PARAM[1] = \$P_EP[\$AC_MARKER[0]]

4.3.3 Conversions de type

Lors des affectations de valeurs et des transferts de paramètres avec des types de données différents, une conversion de type implicite est effectuée entre les types de données suivants :

- REAL
- INT
- BOOL

Remarque

Conversion de REAL vers INT

Lors de la conversion de REAL vers INT, en présence d'une valeur après la virgule $\geq 0,5$, la valeur est arrondie au nombre entier supérieur le plus proche. En présence d'une valeur après la virgule $< 0,5$, la valeur est arrondie au nombre entier inférieur le plus proche.
Comportement selon la fonction ROUND.

Si la valeur REAL se trouve en dehors de la plage de valeurs INT, une alarme s'affiche et la conversion n'est pas effectuée.

Conversion de REAL ou INT vers BOOL

- Valeur $\neq 0 \rightarrow \text{TRUE}$
 - Valeur $= 0 \rightarrow \text{FALSE}$
-

Exemples

Conversion : INT \$AC_MARKER $\rightarrow$ REAL \$AC_PARAM

Code de programme

```
$AC_MARKER[1]=561  
ID=1 WHEN TRUE DO $AC_PARAM[1] = $AC_MARKER[1]
```

4.3 Variable système pour actions synchrones

Conversion : **REAL \$AC_PARAM** → **INT \$AC_MARKER****Code de programme**

```
$AC_PARAM[1]=561.0
ID=1 WHEN TRUE DO $AC_MARKER[1] = $AC_PARAM[1]
```

Conversion : **INT \$AC_MARKER** → **BOOL \$A_OUT****Code de programme**

```
$AC_MARKER[1]=561
ID=1 WHEN $A_IN[1]==TRUE DO $A_OUT[0] = $AC_MARKER[1]
```

Conversion : **REAL \$R401** → **BOOL \$A_OUT****Code de programme**

```
R401 = 100.542
WHEN $A_IN[0]==TRUE DO $A_OUT[2] = $R401
```

Conversion : **BOOL \$A_OUT** → **INT \$AC_MARKER****Code de programme**

```
ID=1 WHEN $A_IN[2]==TRUE DO $AC_MARKER[4] = $A_OUT[1]
```

Conversion : **BOOL \$A_OUT** → **REAL \$R10****Code de programme**

```
WHEN $A_IN[3]==TRUE DO $R10 = $A_OUT[3]
```

4.3.4 **Mémentos/compteurs (\$AC_MARKER)**

Les variables \$AC_MARKER[<indice>] sont des champs spécifiques au canal de variables système à utiliser en tant que memento ou compteur.

Type de données : INT (Integer)

<indice> : Indice de tableau : 0, 1, 2, ... (nombre max. - 1)

Nombre par canal

Le nombre maximal de variables \$AC_MARKER par canal peut être réglé avec le paramètre machine :

PM28256 \$MC_MM_NUM_AC_MARKER = <nombre maximal>

Lieu de stockage

Le lieu de stockage des variables \$AC_MARKER peut être défini de manière spécifique au canal avec le paramètre machine :

PM28257 \$MC_MM_BUFFERED_AC_MARKER = <valeur>

Valeur	Lieu de stockage
0	Mémoire dynamique (réglage par défaut)
1	Mémoire statique

Remarque**Sauvegarde des données et espace mémoire**

- Les variables \$AC_MARKER créées dans la mémoire statique peuvent être sauvegardées de manière spécifique au canal au moyen de la sauvegarde des données. Bloc de données : _N_CH<numéro de canal>_ACM
 - Il convient de veiller à ce que l'espace mémoire nécessaire soit disponible dans la zone de mémoire sélectionnée. Un élément de tableau nécessite 4 octets d'espace mémoire.
-

Comportement reset

Le comportement reset dépend du lieu de stockage des variables \$AC_MARKER :

- Mémoire dynamique : initialisation avec la valeur "0"

Mémoire statique : conservation de la valeur actuelle

4.3.5 Paramètre (\$AC_PARAM)

Les variables \$AC_PARAM[<indice>] sont des champs spécifiques au canal de variables système à utiliser en tant qu'emplacement tampon général.

Type de données : REAL

<indice> : Indice de tableau : 0, 1, 2, ... (nombre max. - 1)

Nombre par canal

Le nombre maximal de variables \$AC_PARAM par canal peut être réglé avec le paramètre machine :

PM28254 \$MC_MM_NUM_AC_PARAM = <nombre maximal>

Lieu de stockage

Le lieu de stockage des variables \$AC_PARAM peut être défini de manière spécifique au canal avec le paramètre machine :

4.3 Variable système pour actions synchrones

PM28255 \$MC_MM_BUFFERED_AC_PARAM = <valeur>

Valeur	Lieu de stockage
0	Mémoire dynamique (réglage par défaut)
1	Mémoire statique

Remarque

Sauvegarde des données et espace mémoire

- Les variables \$AC_PARAM créées dans la mémoire statique peuvent être sauvegardées de manière spécifique au canal au moyen de la sauvegarde des données. Bloc de données : _N_CH<numéro de canal>_ACP
- Il convient de veiller à ce que l'espace mémoire nécessaire soit disponible dans la zone de mémoire sélectionnée. Un élément de tableau nécessite 4 octets d'espace mémoire.

Comportement reset

Le comportement reset dépend du lieu de stockage des variables \$AC_PARAM :

- Mémoire dynamique : initialisation avec la valeur "0"
- Mémoire statique : conservation de la valeur actuelle

4.3.6 Paramètres R (\$R)

Selon que les paramètres R sont précédés ou non d'un caractère \$, ils sont traités comme des variables de prétraitement ou d'exécution. Le choix de la notation est en principe libre. Pour l'utilisation dans des actions synchrones, les paramètres R doivent toutefois être utilisés en tant que variable d'exécution, donc avec un caractère \$:

- \$R[<indice>]
- \$R<numéro>

Type de données : REAL

<indice> : Indice de tableau : 0, 1, 2, ...

<numéro> : Numéro du paramètre R : 0, 1, 2, ...

Les notations avec indice ou numéro sont équivalentes.

Nombre paramétrable par canal

Le nombre maximal de paramètres R par canal peut être réglé avec le paramètre machine :

PM28050 \$MC_MM_NUM_R_PARAM = <nombre maximal>

Comportement reset

Les paramètres R sont enregistrés de manière persistante dans la mémoire statique de la CN. Par conséquent, les paramètres R conservent leurs valeurs au-delà de tous les types de reset en cas de :

- Reset Power On
- Reset CN
- Reset fin du programme pièce

Exemple

Affectation de valeurs à R10 dans la partie action de l'action synchrone, suivie d'une évaluation dans le programme pièce

```
WHEN $A_IN[1]==1 DO $R[10]=$AA_IM[Y] ; Affectation
G1 X100 F150
STOPRE
IF R[10] > 50 ... ; Évaluation dans le programme pièce
```

4.3.7 Paramètres machine et données de réglage (\$\$M, \$\$S)

Lecture et écriture de paramètres machine et de données de réglage

Lors de l'utilisation de paramètres machine et de données de réglage dans des actions synchrones, il convient de distinguer ceux qui restent inchangés pendant l'exécution de l'action synchrone et ceux qui sont modifiés par des processus parallèles :

- Les données qui restent **inchangées** peuvent être lues ou écrites dès le **prétraitement des blocs**.
- Les données qui sont **modifiées** ne peuvent être lues ou écrites qu'au moment de **l'exécution des blocs**.

Accès aux données lors du prétraitement des blocs

Les paramètres machine et les données de réglage qui peuvent être lus ou écrits dans les actions synchrones dès le prétraitement des blocs sont programmés avec les mêmes descripteurs que dans le programme pièce :

\$M... ou \$S...

Code de programme

```
; La position d'inversion de l'axe Z $SA_OSCILL_REVERSE_POS2[Z]
; reste inchangée pendant toute la durée de traitement
ID=2 WHENEVER $AA_IM[z] < $SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Accès aux données lors de l'exécution des blocs

Pour les paramètres machine et les données de réglage qui ne doivent être lus ou écrits dans les actions synchrones qu'au moment de l'exécution des blocs, un caractère "\$" supplémentaire doit être placé avant le descripteur :

\$\$M... ou \$\$S...

Code de programme

```
; La position d'inversion de l'axe Z $SA_OSCILL_REVERSE_POS2[Z]
; peut être modifiée à n'importe quel moment par une action de l'opérateur
ID=1 WHENEVER $AA_IM[z] < $$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X] = 0
```

Écriture lors de l'exécution des blocs

Les conditions suivantes doivent être remplies pour l'écriture lors de l'exécution des blocs :

- Le droit d'accès au moment de l'écriture doit autoriser l'écriture.
- Le paramètre machine ou la donnée de réglage doit posséder la propriété "immédiatement effectif".

Code de programme

```
; La position de commutation des cames logicielles $SN_SW_CAM... ne doit,
selon
; la consigne actuelle de l'axe X dans le SCP $AA_IW[X], être écrite qu'au
moment
; de l'exécution des blocs
ID=2 WHEN $AA_IW[X] > 10 DO $$SN_SW_CAM_PLUS_POS_TAB_1[0] = 20
                                $$SN_SW_CAM_MINUS_POS_TAB_1[0] = 20
```

Une vue d'ensemble complète des propriétés des paramètres machine et des données de réglage est disponible dans :

Plus d'informations

- Tables de paramètres Variables système
- Tables de paramètres Paramètres machine

4.3.8 Temporisation (\$AC_TIMER)

Les variables \$AC_TIMER[<indice>] sont des champs spécifiques au canal de variables système.

Type de données :

REAL

<indice> :

Indice de tableau : 0, 1, 2, ... (nombre max. - 1)

Unité :

Seconde

Nombre par canal

Le nombre maximal de variables \$AC_TIMER par canal peut être réglé avec le paramètre machine :

PM28258 \$MC_MM_NUM_AC_TIMER = <nombre maximal>

Fonction

Démarrer

Le démarrage d'une temporisation s'effectue par l'affectation d'une valeur ≥ 0 :

`$AC_TIMER[<indice>] = <valeur de départ>` ; avec une valeur de départ ≥ 0

Incrémentation

Pour chaque période d'appel de l'interpolateur, la valeur de la temporisation est incrémentée de la durée de la période d'appel de l'interpolateur réglée (PM10071 IPO_CYCLE_TIME).

`$AC_TIMER[<indice>] += <période d'appel de l'interpolateur>`

Arrêt

L'arrêt d'une temporisation s'effectue par l'affectation d'une valeur < 0 :

`$AC_TIMER[<indice>] = <valeur de coupure>` ; avec une valeur de coupure < 0

Lors de l'affectation d'une valeur de coupure, seule la poursuite de l'incrémentation de la temporisation est arrêtée. La valeur de coupure n'est pas affectée. Après l'arrêt de la temporisation, la dernière valeur valide est conservée et reste accessible à la lecture.

Remarque

La valeur courante d'une temporisation peut être lue que la temporisation soit incrémentée ou arrêtée.

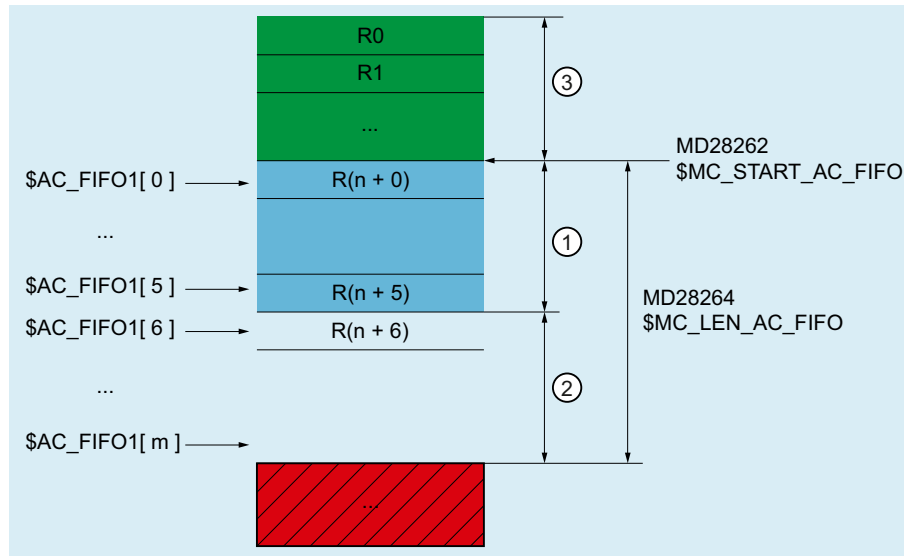
Exemple

Sortie de la valeur réelle de l'axe X en tant que valeur de tension via la sortie analogique `$A_OUTA[3]`, 500 ms après la détection de l'entrée TOR `$A_IN[1]` :

Code de programme	Commentaire
WHEN <code>\$A_IN[1]</code> ==1 DO <code>\$AC_TIMER[1]</code> =0	; Démarrer la temporisation, valeur de départ 0
WHEN <code>\$AC_TIMER[1]</code> >=0.5 DO <code>\$A_OUTA[3]</code> =\$AA_IM[X] <code>\$AC_TIMER[1]</code> =-1	

4.3.9 Variables FIFO (\$AC_FIFO)

Une variable FIFO met à disposition une structure de données complexe basée sur les paramètres R. La structure de données se compose d'une partie gestion et d'une partie données utiles. La partie données utiles est organisée en tant que pile (Stack) selon le principe FIFO (First In, First Out). Les différentes fonctions dans la partie gestion ainsi que les données utiles sont activées par l'indice des variables FIFO.



- ① Partie gestion
- ② Partie données utiles
- ③ Plage de paramètres R au-dessus des variables FIFO : écriture et lecture autorisées
- ④ Plage de paramètres R au-dessous des variables FIFO : seule la lecture est autorisée

Remarque

Les informations données au sujet des paramètres R sont également valables pour les variables FIFO. Voir chapitre "Paramètres R (\$R) (Page 32)".

Syntaxe

Écriture

```
$AC_FIFO<n>[<i>] = <valeur>
$AC_FIFO[<n>, <i>] = <valeur>
```

Lecture

```
<variable> = $AC_FIFO<n>[<i>]
<variable> = $AC_FIFO[<n>, <i>]
```

Signification

\$AC_FIFO :	Structure de données FIFO dans les paramètres R, commençant à partir de la valeur du MD28262 \$MC_START_AC_FIFO	
	Type de données :	REAL
<n> :	Numéro des variables FIFO	
	Type de données :	INT
	Plage de valeurs :	1, 2, ... numéro max. (voir Plus d'informations)

4.3 Variable système pour actions synchrones

<i> :	Indice des variables FIFO par le biais duquel s'effectue l'accès aux différentes fonctions et données au sein de la structure de données des variables FIFO.	
	Plage de valeurs :	0, 1, 2, ... (MD28264 \$MC_LEN_AC_FIFO - 1)
	Valeur	Signification
	Données de gestion	
	0	Écriture L'écriture d'une valeur sur la pile FIFO s'effectue par l'affectation d'une valeur à la pile FIFO via l'indice 0 (\$AC_FIFO[0] = <valeur>). La valeur affectée est alors écrite au prochain emplacement libre de la pile FIFO. Si tous les emplacements de la pile FIFO sont déjà occupés, une alarme s'affiche : - Lors de l'écriture dans un programme CN : Alarme 20149 - Lors de l'écriture dans une action synchrone : Alarme 17020 Lecture La lecture d'une valeur de la pile FIFO s'effectue par l'affectation de la pile FIFO avec l'indice 0 à une variable (<variable> = \$AC_FIFO[0]). La valeur la plus ancienne est alors lue, puis supprimée de la pile FIFO. Remarque - Lecture dans le programme CN / l'action synchrone Si la lecture a lieu avec l'indice 0 dans un programme CN / une action synchrone, la valeur la plus ancienne est lue et supprimée de la pile FIFO comme décrit plus haut. - Lecture sur l'interface utilisateur, par ex. SINUMERIK Operate Si la lecture ou l'affichage a lieu avec l'indice 0 sur l'interface utilisateur, par ex. SINUMERIK Operate : "Diagnostic" > "Variables CN/AP", la lecture se fait en interne avec l'indice 1 (valeur la plus ancienne) sans modification de la pile FIFO.
	1	Écriture/lecture : La donnée utile "la plus ancienne" est activée ; la pile FIFO n'est pas modifiée
	2	Écriture/lecture : La donnée utile "la plus récente" est activée ; la pile FIFO n'est pas modifiée
	3	Lecture : Fournit la somme des valeurs de toutes les données utiles Déblocage avec le MD28266 \$MC_MODE_AC_FIFO, bit 0 nécessaire. Voir ci-dessous la section "Calcul de la somme de toutes les données utiles"
	4	Lecture : Fournit le nombre de données utiles se trouvant actuellement dans la pile FIFO Écriture : Le rétablissement de l'état initial s'effectue par l'écriture de la valeur 0 dans l'indice 4 des variables FIFO. Exemple : \$AC_FIFO1[4] = 0
	5	Lecture : Fournit l'indice d'écriture actuel par rapport au début de la pile FIFO
	Données utiles	
	6	Écriture/lecture : Le 1er élément de tableau de la zone de données utiles est activé
	7	Écriture/lecture : Le 2e élément de tableau de la zone de données utiles est activé
	n	Écriture/lecture : Le n-ième élément de tableau de la zone de données utiles est activé

Informations complémentaires

Tables de paramètres Variables système

Paramètres machine**Nombre de variables FIFO par canal**

Le nombre de variables FIFO par canal se règle via :

MD28260 \$MC_NUM_AC_FIFO = <nombre de variables FIFO par canal>

Début de la plage de paramètres R des variables FIFO

Le paramètre R à partir duquel débute la plage de variables FIFO pour le canal se règle via :

MD28262 \$MC_START_AC_FIFO = <numéro du paramètre R de départ>

Remarque**Paramètres R libres**

Seuls les paramètres R qui se trouvent en dessous du paramètre R de départ des variables FIFO en termes de numérotation peuvent être écrits dans le programme CN.

Nombre d'éléments de tableau par variable FIFO

Le nombre maximal d'éléments de tableau par variable FIFO se règle via :

MD28264 \$MC_LEN_AC_FIFO = <nombre d'éléments de tableau par variable FIFO>

Nombre total de paramètres R dans le canal

Le nombre total de paramètres R nécessaires dans le canal se règle via :

$$\begin{aligned} \text{MD28050 \$MC_MM_NUM_R_PA-} & \quad \text{MD28262 \$MC_START_AC_FIFO +} \\ \text{RAM =} & \quad \text{MD28260 \$MC_NUM_AC_FIFO *} \\ & \quad (\text{MD28264 \$MC_LEN_AC_FIFO} + 6) \end{aligned}$$
Calcul de la somme de toutes les données utiles

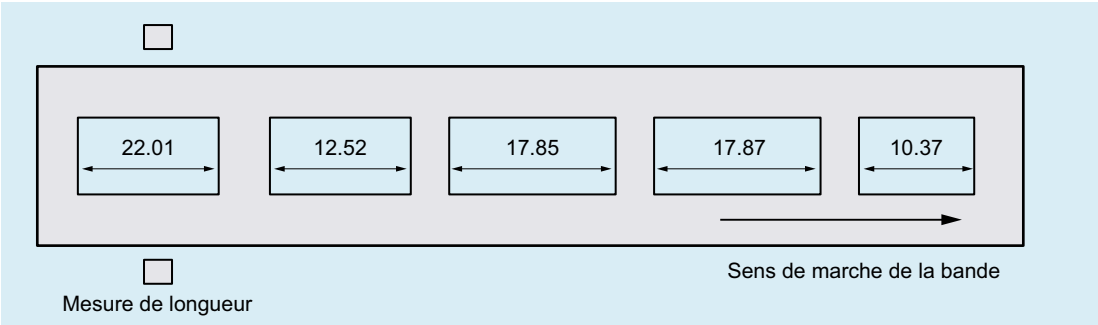
La somme des valeurs de toutes les données utiles n'est mise à disposition via \$AC_FIFO[3] que lorsque la fonction est activée avec le paramètre machine :

MD28266 \$MC_MODE_AC_FIFO, bit 0 = <valeur>

Valeur	Signification
FALSE	La somme des valeurs de toutes les données utiles n'est pas mise à disposition
TRUE	La somme des valeurs de toutes les données utiles est mise à disposition

Exemple

Détermination en série de la longueur des pièces qui sont déplacées par une bande transporteuse devant une station de mesure automatique.



Les résultats de mesure sont écrits avec des actions synchrones dans la variable système \$AC_FIFO1 ou sont lus à partir de cette dernière.

- **Lecture** : Avec l'indice 0, l'élément de données utiles "le plus ancien" est lu et supprimé de la pile FIFO.
- **Écriture** : Avec l'indice 0, la valeur est écrite dans le prochain élément de données utiles libre.

1 Lecture :
élément le plus ancien
N10 \$R1 = \$AC_FIFO1[0]

2 Écriture :
prochain élément
N20 \$AC_FIFO1[0] = 22.01

avant		
\$AC_FIFO1[0 ... 12]		
	Désignation	Valeur
[0]	—	non défini
[1]	élément le plus ancien	[6]
[2]	élément le plus récent	[9]
[3]	Somme de toutes les	80.62
[4]	valeurs	4
[5]	Nombre d'éléments	10
[6]	affectés	10.37
[7]	Indice d'écriture actuel	17.87
[8]	Élément de données utiles	17.85
[9]	Élément de données utiles	12.52
[10]	Élément de données utiles	non défini
[11]	Élément de données utiles	non défini
[12]	Élément de données utiles	non défini

après lecture et écriture		
\$AC_FIFO1[0 ... 12]		
	Désignation	Valeur
[0]	—	non défini
[1]	élément le plus ancien	[7]
[2]	élément le plus récent	[10]
[3]	Somme de toutes les	90.26
[4]	valeurs	4
[5]	Nombre d'éléments	11
[6]	affectés	non défini
[7]	Indice d'écriture actuel	17.87
[8]	Élément de données utiles	17.85
[9]	Élément de données utiles	12.52
[10]	Élément de données utiles	22.01
[11]	Élément de données utiles	non défini
[12]	Élément de données utiles	non défini

4.3.10 Angle de la tangente à la trajectoire (\$AC_TANEB)

La variable système spécifique au canal \$AC_TANEB (Tangent **AN**gle at **EN**d of **B**lock) permet de lire l'angle entre la tangente au point final du bloc actuel et la tangente au point de départ du bloc suivant.

Type de données : REAL

L'angle de la tangente est toujours indiqué positivement dans la plage de 0,0° à 180,0°.

Si l'angle de la tangente ne peut pas être déterminé, la valeur -180,0° est émise.

Utilisation uniquement pour les blocs programmés

Il est recommandé de lire l'angle de la tangente uniquement pour les blocs programmés, et pas pour les blocs intermédiaires générés par le système. Une distinction peut être établie avec la variable système \$AC_BLOCKTYPE :

\$AC_BLOCKTYPE == 0 (bloc programmé)

Exemple :

Code de programme

```
ID=2 EVERY $AC_BLOCKTYPE==0 DO $R1=$AC_TANEB
```

4.3.11 Correction (\$A...OVR)

Correction actuelle

Correction spécifique au canal

La variable système spécifique au canal \$AC_OVR permet de modifier l'avance tangentielle.

Type de données : REAL

Unité : %

Plage de valeurs : 0,0 jusqu'au paramètre machine

- avec commutateur de correction en code **binaire**
PM12100 \$MN_OVR_FACTOR_LIMIT_BIN
- avec commutateur de correction codé en **gray**
PM12030 \$MN_OVR_FACTOR_FEEDRATE[30]

La variable système \$AC_OVR doit être réécrite à chaque période d'appel de l'interpolateur, faute de quoi la valeur "100%" s'applique.

Correction du rapide spécifique au canal

Pour les blocs G0 (rapide), l'avance rapide peut être influencée par la donnée de réglage SD42122 \$SC_OVR_RAPID_FACTOR en plus de la variable système \$AC_OVR.

Condition : Déblocage de la correction de rapide via l'interface utilisateur.

4.3 Variable système pour actions synchrones

Correction spécifique à l'axe

Le variable système spécifique à l'axe \$AA_OVR permet de modifier l'avance axiale :

Type de données :	REAL
Unité :	%
Plage de valeurs :	0,0 jusqu'au paramètre machine - avec commutateur de correction en code binaire PM12100 \$MN_OVR_FACTOR_LIMIT_BIN - avec commutateur de correction codé en gray PM12030 \$MN_OVR_FACTOR_FEEDRATE[30]

La variable système \$AA_OVR doit être réécrite à chaque période d'appel de l'interpolateur, faute de quoi la valeur "100%" s'applique.

Correction de l'AP

Correction spécifique au canal

Le variable système spécifique au canal \$AC_PLC_OVR permet de lire la correction spécifique au canal (<Chan>.basic.out.pathFeedrateOvrFactor) réglée via le tableau de commande machine :

Type de données :	REAL
Unité :	%
Plage de valeurs :	0,0 jusqu'à valeur maximale

Correction spécifique à l'axe

Le variable système spécifique à l'axe \$AA_PLC_OVR permet de lire la correction spécifique à l'axe (<Axis>.basic.out.feedrateOvrFactor) réglée via le tableau de commande machine :

Type de données :	REAL
Unité :	%
Plage de valeurs :	0,0 jusqu'à valeur maximale

Correction active

Correction active spécifique au canal

La correction active spécifique au canal peut être lue via la variable système spécifique au canal \$AC_TOTAL_OVR :

Type de données : REAL
Unité : %
Plage de valeurs : 0,0 jusqu'à valeur maximale

Correction active spécifique à l'axe

La correction active spécifique à l'axe peut être lue via la variable système spécifique à l'axe \$AA_TOTAL_OVR :

Type de données : REAL
Unité : %
Plage de valeurs : 0,0 jusqu'à valeur maximale

Signaux AP

CN → AP

Basic Program Plus	Basic Program	
<Chan>.basic.out.pathFeedrateOvrFactor	LBP_Chan*.A_FD_OR	DB21,DBB4
<Axis>.basic.out.feedrateOvrFactor	LBP_Axis*.A_FD_OR	DB31,DBB0

4.3.12 Évaluation de l'utilisation (\$AN_IPO..., \$AN/AC_SYNC..., \$AN_SERVO)

Les variables système suivantes permettent de lire les valeurs de l'utilisation courante, maximale et moyenne du système en raison des actions synchrones :

Variable système spécifique à la CN	Signification
\$AN_IPO_ACT_LOAD	Temps de calcul courant du niveau de l'interpolateur (incluant les actions synchrones de tous les canaux)
\$AN_IPO_MAX_LOAD	Temps de calcul le plus long du niveau de l'interpolateur (incluant les actions synchrones de tous les canaux)
\$AN_IPO_MIN_LOAD	Temps de calcul le plus court du niveau de l'interpolateur (incluant les actions synchrones de tous les canaux)
\$AN_IPO_LOAD_PERCENT	Temps de calcul courant du niveau de l'interpolateur par rapport à la période d'appel de l'interpolateur (%).

4.3 Variable système pour actions synchrones

Variable système spécifique à la CN	Signification
\$AN_SYNC_ACT_LOAD	Temps de calcul courant des actions synchrones pour tous les canaux
\$AN_SYNC_MAX_LOAD	Temps de calcul le plus long des actions synchrones pour tous les canaux
\$AN_SYNC_TO_IPO	Pourcentage des actions synchrones dans le temps de calcul total du niveau de l'interpolateur (pour tous les canaux)
\$AN_SERVO_ACT_LOAD	Temps de calcul courant du régulateur de position
\$AN_SERVO_MAX_LOAD	Temps de calcul le plus long du régulateur de position
\$AN_SERVO_MIN_LOAD	Temps de calcul le plus court du régulateur de position

Variable système spécifique au canal	Signification
\$AC_SYNC_ACT_LOAD	Temps de calcul courant pour les actions synchrones dans le canal
\$AC_SYNC_MAX_LOAD	Temps de calcul le plus long pour les actions synchrones dans le canal
\$AC_SYNC_AVERAGE_LOAD	Temps de calcul moyen pour les actions synchrones dans le canal

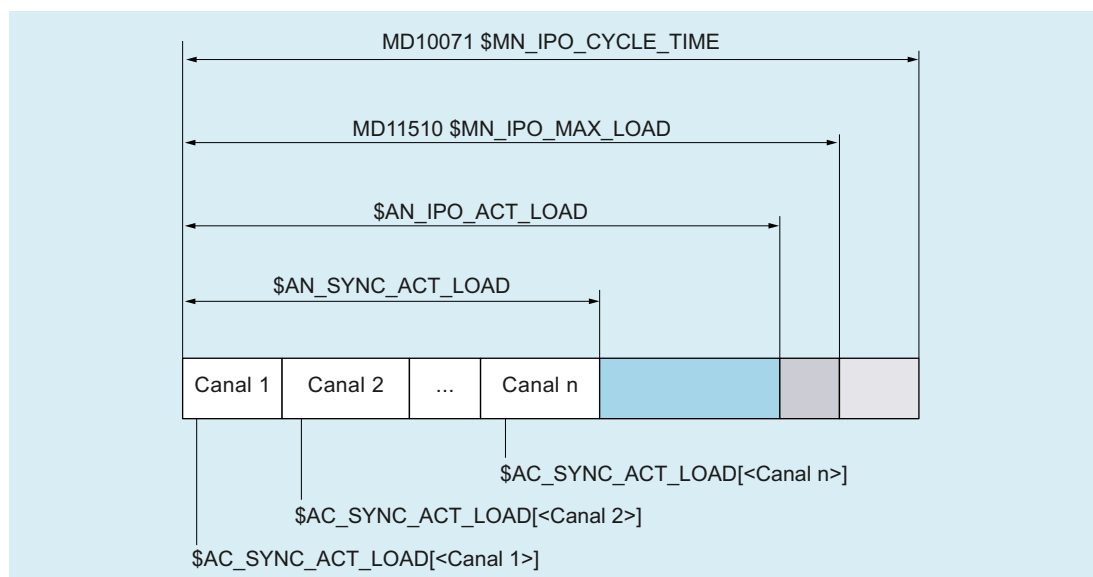


Figure 4-1 Composantes de temps de calcul des actions synchrones dans la période d'appel de l'interpolateur

Activation

Les variables système contiennent des valeurs valides uniquement lorsque la fonction de diagnostic "Évaluation de l'utilisation sur les actions synchrones" est active.

Pour cela, le paramètre machine suivant doit être défini sur une valeur supérieure à zéro :

PM11510 \$MN_IPO_MAX_LOAD > 0 (utilisation maximale autorisée de l'interpolateur)

Lorsque la fonction est activée, la ligne "Temps requis pour les actions synchrones" avec les valeurs courantes est affichée dans le groupe fonctionnel "Diagnostic" > "Utilisation du système".

Remarque

Les variables système comprennent toujours les valeurs de la période d'appel de l'interpolateur précédente.

Limite de surcharge

La valeur réglée avec le PM11510 \$MN_IPO_MAX_LOAD permet de définir une limite de surcharge :

PM11510 \$MN_IPO_MAX_LOAD = <utilisation maximale autorisée en % de la période d'appel de l'interpolateur>

Si la valeur réglée dans le paramètre machine est dépassée, la variable système suivante est activée :

\$AN_IPO_LOAD_LIMIT = TRUE

En cas de passage sous la valeur réglée, la variable système est réinitialisée :

\$AN_IPO_LOAD_LIMIT = FALSE

Application

La variable système \$AN_IPO_LOAD_LIMIT permet de mettre en place une stratégie propre à l'utilisateur pour éviter un dépassement de plan.

Réinitialisation des valeurs min./max.

L'écriture de valeurs quelconques réinitialise les variables système suivantes pour les valeurs min./max. :

Variable système	Signification
\$AN_SERVO_MAX_LOAD	Temps de calcul le plus long du régulateur de position
\$AN_SERVO_MIN_LOAD	Temps de calcul le plus court du régulateur de position
\$AN_IPO_MAX_LOAD	Temps de calcul le plus long du niveau de l'interpolateur (incluant les actions synchrones de tous les canaux)
\$AN_IPO_MIN_LOAD	Temps de calcul le plus court du niveau de l'interpolateur (incluant les actions synchrones de tous les canaux)
\$AN_SYNC_MAX_LOAD	Temps de calcul le plus long des actions synchrones pour tous les canaux
\$AC_SYNC_MAX_LOAD	Temps de calcul le plus long pour les actions synchrones dans le canal

4.3 Variable système pour actions synchrones

Exemple

Code de programme	Commentaire
\$MN_IPO_MAX_LOAD=80	; Limite de surcharge
;	
; Initialisation des valeurs min./max.	
N01 \$AN_SERVO_MAX_LOAD=0	
N02 \$AN_SERVO_MIN_LOAD=0	
N03 \$AN_IPO_MAX_LOAD=0	
N04 \$AN_IPO_MIN_LOAD=0	
N05 \$AN_SYNC_MAX_LOAD=0	
N06 \$AC_SYNC_MAX_LOAD=0	
;	
; Alarme 63111 en cas de dépassement de la limite de surcharge	
N10 IDS=1 WHENEVER \$AN_IPO_LOAD_LIMIT==TRUE DO M4711 SETAL(63111)	
;	
; Alarme 63222 en cas de dépassement de la composante de temps de calcul	
; des actions synchrones sur tous les canaux de 30 % de la période d'appel de l'in-	
terpolateur	
N20 IDS=2 WHENEVER \$AN_SYNC_TO_IPO > 30 DO SETAL(63222)	
;	
N30 G0 X0 Y0 Z0	
...	
N999 M30	

4.3.13 Limitation de la zone de travail (\$SA_WORKAREA...)

Pour les axes de commande déplaçables dans des actions synchrones, seule l'activation avec les données de réglage suivantes est effective en ce qui concerne la limitation programmable de la zone de travail G25 / G26 :

- \$SA_WORKAREA_PLUS_ENABLE
- \$SA_WORKAREA_MINUS_ENABLE

Une activation/désactivation de la limitation de la zone de travail par le biais des instructions WALIMON / WALIMOF dans le programme pièce n'a aucun effet sur les axes de commande déplacés avec des actions synchrones.

4.3.14 Positions et temps des cames logicielles (\$\$SN_SW_CAM...)

Les données de réglage suivantes permettent de lire et d'écrire les valeurs des positions et des temps des cames logicielles :

Donnée de réglage spécifique à la CN	Signification
\$SN_SW_CAM_MINUS_POS_TAB_1[0..7]	Positions de la came moins
\$SN_SW_CAM_MINUS_POS_TAB_2[0..7]	Positions de la came moins

Donnée de réglage spécifique à la CN	Signification
\$SN_SW_CAM_PLUS_POS_TAB_1[0..7]	Positions de la came plus
\$SN_SW_CAM_PLUS_POS_TAB_2[0..7]	Positions de la came plus
\$SN_SW_CAM_MINUS_TIME_TAB_1[0..7]	Constante de temps ou temporisation de la came moins
\$SN_SW_CAM_MINUS_TIME_TAB_2[0..7]	Constante de temps ou temporisation de la came moins
\$SN_SW_CAM_PLUS_TIME_TAB_1[0..7]	Constante de temps ou temporisation de la came plus
\$SN_SW_CAM_PLUS_TIME_TAB_2[0..7]	Constante de temps ou temporisation de la came plus

Remarque

La définition d'une came logicielle par des actions synchrones ne doit pas se produire immédiatement avant d'atteindre la came. Il doit encore rester au moins 3 périodes d'appel de l'interpolateur avant d'atteindre la came.

Une description détaillée de la fonction "Cames logicielles" est disponible dans :

Informations complémentaires

Description fonctionnelle Axes et broches

Exemples**Code de programme**

```
; Modification d'une position de came
ID=1 WHEN $AA_IW[x] > 0 DO $$SN_SW_CAM_MINUS_POS_TAB_1[0] = 50.0
...
; Modification d'un temps de dérivation
ID=1 WHEN $AA_IW[x] > 0 DO $$SN_SW_CAM_MINUS_TIME_TAB_1[0] = 1.0
```

Voir aussi

Paramètres machine et données de réglage (\$\$M, \$\$S) (Page 33)

4.3.15 Évaluation de longueur de déplacement / maintenance de la machine (\$AA_TRAVEL..., \$AA_JERK...)

Les variables système présentées ci-dessous permettent de lire les données de l'évaluation de longueur de déplacement, pour la maintenance de la machine par exemple.

Activation

L'activation pour l'enregistrement des données de l'évaluation de longueur de déplacement s'effectue via :

4.3 Variable système pour actions synchrones

MD18860 \$MN_MM_MAINTENANCE_MON = 1

Le paramètre machine suivant spécifique à l'axe permet de sélectionner les données qui sont enregistrées pour l'axe spécifique :

MD33060 \$MA_MAINTENANCE_DATA[<axe>], bit n = 1

Bit	Signification
0	Enregistrement de la distance de déplacement totale, du temps de déplacement total et du nombre de processus de déplacement de l'axe
1	Enregistrement de la distance de déplacement totale, du temps de déplacement total et du nombre de processus de déplacement à grande vitesse de l'axe
2	Enregistrement de la somme totale de l'à-coup de l'axe, du temps pendant lequel l'axe est déplacé avec à-coup et du nombre de processus de déplacement avec à-coup.

Variable système

Variable système	Signification	n
\$AA_TRAVEL_DIST	Distance de déplacement totale : Somme de tous les changements de position de consigne dans le SCM en [mm] ou [degré].	0
\$AA_TRAVEL_TIME	Temps de déplacement total : Somme de toutes les périodes d'appel de l'interpolateur des changements de position de consigne dans le SCM en [s] (résolution : 1 période d'appel de l'interpolateur)	
\$AA_TRAVEL_COUNT	Nombre total de processus de déplacement : Un processus de déplacement d'un axe machine se caractérise par la séquence d'états suivante par rapport à la position de consigne : arrêt > déplacement > arrêt	
\$AA_TRAVEL_DIST_HS	Distance de déplacement totale à grande vitesse de l'axe ¹⁾	1
\$AA_TRAVEL_TIME_HS	Temps de déplacement total à grande vitesse de l'axe ¹⁾	
\$AA_TRAVEL_COUNT_HS	Nombre total de processus de déplacement à grande vitesse de l'axe ³⁾	
\$AA_JERK_TOT	Somme totale de l'à-coup de l'axe : Somme de toutes les consignes de l'à-coup en [m/s ³] ou [degré/s ³].	2
\$AA_JERK_TIME	Temps de déplacement total avec à-coup : Somme des périodes d'appel de l'interpolateur des changements de consigne de l'à-coup en [s] (résolution : 1 période d'appel de l'interpolateur)	
\$AA_JERK_COUNT	Nombre total de processus de déplacement avec à-coup	

¹⁾ Vitesse réelle de l'axe machine ≥ 80 % de la vitesse d'axe maximale paramétrée (MD32000 MAX_AX_VELO)

Informations complémentaires

Une description détaillée de la fonction est disponible dans :

Description fonctionnelle Axes et broches

4.3.16 Coefficients/paramètres de polynôme (\$AC_FCT...)

Fonction

La fonction FCTDEF permet de définir des polynômes du 3^è degré au maximum :

$$f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3$$

Remarque

La définition doit se faire dans un **programme pièce**.

Syntaxe

FCTDEF (<Poly_Nr>, <Lo_Limit>, <Up_Limit>, a₀, a₁, a₂, a₃)

Signification

Paramètres	Signification
<Poly_Nr> :	Numéro de la fonction polynôme
<Lo_Limit> :	Limite inférieure des valeurs de fonction
<Up_Limit> :	Limite supérieure des valeurs de fonction
a ₀ , a ₁ , a ₂ , a ₃ :	coefficients du polynôme

Remarque

Les coefficients du polynôme inutiles (a₂, a₃) peuvent être omis lors de la programmation de la fonction FCTDEF (...).

Variable système

A partir des actions synchrones, il est possible d'accéder aux coefficients et paramètres du polynôme en lecture et en écriture via les variables système suivantes :

Variable système	Signification
\$AC_FCTL[<Poly_Nr>] :	Limite inférieure de la valeur de la fonction
\$AC_FCTUL[<Poly_Nr>] :	limite supérieure de la valeur de la fonction
\$AC_FCT0[<Poly_Nr>] :	a ₀
\$AC_FCT1[<Poly_Nr>] :	a ₁
\$AC_FCT2[<Poly_Nr>] :	a ₂
\$AC_FCT3[<Poly_Nr>] :	a ₃
<Poly_Nr> :	Le numéro de la fonction du polynôme indiqué lors de la définition (voir ci-dessus : syntaxe)

4.3 Variable système pour actions synchrones

Programme pièce

Lors de l'écriture des variables système dans le programme pièce, l'arrêt de prétraitement `STOPRE` doit être programmé de manière explicite pour une écriture synchrone par bloc.

Remarque

Ecriture synchrone par bloc dans le programme pièce

Pour que les variables système soient écrites de manière synchrone par bloc dans le programme pièce, il convient d'utiliser l'instruction `STOPRE` (arrêt du prétraitement) après l'écriture des variables système.

Action synchrone

Lors de l'écriture de variables système dans des actions synchrones, celles-ci sont actives immédiatement.

Utilisation

La valeur de fonction $f(x)$ du polynôme des actions synchrones peut être utilisée comme valeur d'entrée, par exemple pour les fonctions suivantes :

- "Evaluation du polynôme (SYNFCT) (Page 75)"
- "Correction d'outil en ligne (FTOC) (Page 81)"

Exemple : dépendance linéaire

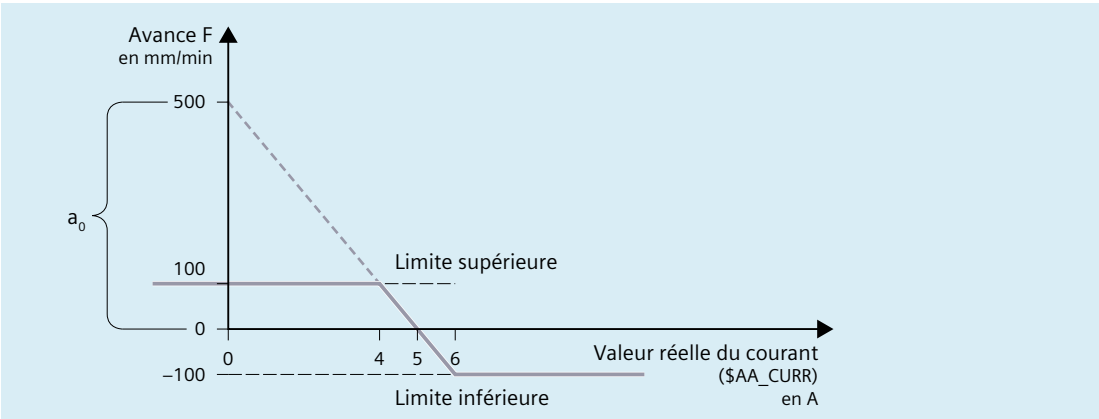


Figure 4-2 Exemple de dépendance linéaire

Paramètres	Signification
<Poly_Nr> :	numéro du polynôme, par ex. = 1
<Lo_Limit> :	Limite inférieure des valeurs de fonction = -100
<Up_Limit> :	Limite supérieure des valeurs de fonction = 100
a_0 :	Section d'axe sur l'ordonnée (avance) : $(5 - 4) / 100 = 5 / a_0$ $a_0 = 100 * 5 / (5 - 4) = 500$

Paramètres	Signification
a_1 :	Pente de la droite : $a_1 = 100 / (4 - 5) = -100$
a_2 :	= 0 (pas de terme quadratique)
a_3 :	= 0 (pas de terme cubique)

Code de programme

```
FCTDEF(1, -100, 100, 500, -100, 0, 0)
; ou en abrégé sans les paramètres a2 et a3
FCTDEF(1, -100, 100, 500, -100)
```

4.3.17 Déplacements forcés (\$AA_OFF)

Déplacements forcés

La variable système \$AA_OFF permet de spécifier un offset de position dans un axe de canal qui est effectué immédiatement :

$\$AA_OFF[\text{<axe de canal>}] = \text{<offset de position>}$

Le paramètre machine suivant permet de régler si l'offset de position est affecté à la variable système ou s'il est ajouté (intégré) dans celle-ci :

MD36750 \$MA_AA_OFF_MODE, bit 0 = <valeur>

<Valeur>	Signification
0	Affectation : $\$AA_OFF = \text{<offset de position>}$
1	Totalisation (intégration) : $\$AA_OFF += \text{<offset de position>}$

Limitation de la vitesse de superposition

La vitesse maximale autorisée à laquelle l'offset de position est réalisé peut être réglée via le paramètre machine :

MD32070 \$MA_CORR_VELO (vitesse d'axe pour correction)

Limitation axiale des à-coups

La définition du paramètre machine suivant peut activer une limitation axiale des à-coups pour la correction de \$AA_OFF :

MD32420 \$MA_JOG_AND_POS_JERK_ENABLE (réglage de base de la limitation axiale des à-coups) = 1

4.3 Variable système pour actions synchrones

L'à-coup axial est limité à la valeur paramétrée dans MD32430
\$MA_JOG_AND_POS_MAX_JERK (à-coup axial).

Remarque

Pour le mouvement \$AA_OFF de correction, aucun pilotage anticipé de la vitesse n'est possible. Cela peut entraîner, notamment dans le cas d'une spécification périodique (via des actions synchrones) des valeurs de correction de \$AA_OFF, une courbe de vitesse discontinue. Dans ce cas, il est recommandé, dans la mesure du possible, de désactiver la limitation des à-coups.

Limite supérieure de la valeur de correction

La donnée de réglage suivante permet de limiter la valeur de \$AA_OFF :

SD43350 \$SA_AA_OFF_LIMIT (limite supérieure de la valeur de correction \$AA_OFF en régulation d'écartement)

La lecture de la limitation peut s'effectuer avec la variable système suivante :

\$AA_OFF_LIMIT[<axe>] == <valeur>

Valeur	Signification
-1	La limite de la valeur de correction en sens négatif a été atteinte.
1	La limite de la valeur de correction en sens positif a été atteinte.
0	Aucune limitation de la valeur de correction

Comportement reset

Pour les actions synchrones statiques (IDS = <Numéro> DO \$AA_OFF = <Valeur>), la désélection de l'offset de position actif dans \$AA_OFF entraîne une répétition immédiate du déplacement forcé. Le comportement au Reset relatif à \$AA_OFF peut donc être réglé via le paramètre machine suivant :

MD36750 \$MA_AA_OFF_MODE, bit 1 = <valeur>

<Valeur>	Signification
0	L'offset de position dans \$AA_OFF est désélectionné avec RESET
1	L'offset de position dans \$AA_OFF est conservé après RESET

Mode de fonctionnement Jog

L'exécution d'un déplacement forcé commandé par \$AA_OFF peut également être débloquée pour le mode de fonctionnement JOG :

MD36750 \$MA_AA_OFF_MODE, bit 2 = <valeur>

<Valeur>	Signification
0	Mode de fonctionnement JOG : déplacement forcé commandé par \$AA_OFF bloqué
1	Mode de fonctionnement JOG : déplacement forcé commandé par \$AA_OFF débloqué

Un changement de mode de fonctionnement en mode JOG n'est autorisé que si l'offset de position actuel a été effectué. Sinon, l'alarme suivante est affichée :

Alarme "16907 Action ... possible uniquement à l'état de suspension de programme"

Autres conditions à prendre en compte

- **Routines d'interruption et ASUP**
Lors de l'activation d'une routine d'interruption, les actions synchrones modales de déplacement sont conservées et sont également actives dans l'ASUP. Si un retour de sous-programme n'est pas réalisé avec `REPOS`, les actions synchrones modales modifiées dans le sous-programme asynchrone restent actives dans le programme principal.
- **REPOS**
Dans le bloc restant, les actions synchrones restent actives comme dans le bloc d'interruption. Les modifications apportées aux actions synchrones modales dans l'ASUP ne sont pas actives dans le programme interrompu. Les coefficients du polynôme programmés avec `FCTDEF` ne sont pas influencés par ASUP et `REPOS`.
Dans l'ASUP, les coefficients du polynôme issus du programme appelant sont actifs. Dans le programme appelant, les coefficients du polynôme issus de l'ASUP sont actifs.
- **Fin du programme**
Les coefficients du polynôme programmés avec `FCTDEF` sont toujours effectifs après la fin du programme.
- **Recherche de bloc : Collecte des coefficients du polynôme**
En cas de recherche de bloc avec calcul, les coefficients du polynôme sont collectés dans les variables système.
- **Recherche de bloc : Désactivation de déplacements forcés actifs**
Lors de la recherche par bloc, les instructions `CORROF` et `DRFOF` sont collectées et transmises sous forme de bloc d'action. Tous les décalages DRF désactivés sont alors collectés dans le dernier bloc qui contient `CORROF` ou `DRFOF`.
Les instructions de désactivation de déplacement forcé `CORROF (<Axe>, "AA_OFF")` ne sont pas collectées lors de la recherche par bloc. Si un utilisateur veut continuer d'utiliser cette recherche, il peut le faire avec la recherche par bloc avec le test de programme "SERUPRO".

Informations complémentaires

Description fonctionnelle Fonctions de base

- **Désactivation de l'offset de position avec actions synchrones actives**
Si une action synchrone est active lors de la désactivation de l'offset de position via l'instruction `COROFF (<Axe>, "AA_OFF")`, l'alarme 21660 s'affiche. Au même moment `$AA_OFF` est désactivé et n'est pas remis à 1. Si l'action synchrone est activée ultérieurement dans le bloc après `CORROF`, `$AA_OFF` reste à 1 et l'interpolation d'un offset de position est effectuée.

Remarque

Selon le système de coordonnées (SCB ou SCP) dans lequel une variable d'exécution est définie, les frames sont pris en compte ou non.

Les distances sont toujours calculées dans le système de base réglé (métrique ou anglo-saxon). Une commutation avec G70 ou G71 n'a pas d'effet.

Les décalages DRF, les décalages d'origine externes, etc. ne sont pris en considération qu'avec les variables d'exécution définies dans le SCM.

4.3.18 Correction de longueur d'outil en ligne (\$AA_TOFF)

Fonction

En liaison avec une transformation d'orientation active ou avec un organe porte-outil actif, des corrections de longueur d'outil peuvent être effectuées en temps réel pendant l'usinage. La modification de la longueur d'outil effective par la correction de longueur d'outil en ligne entraîne, en cas de changements d'orientation, des déplacements compensatoires modifiés des axes impliqués dans la transformation. Les vitesses qui en résultent peuvent être supérieures ou inférieures, en fonction de la cinématique de la machine et des positions d'axe actuelles.

La vitesse et l'accélération avec lesquelles sont effectués les déplacements correspondant aux corrections de longueur d'outil spécifiées par la variable système \$AA_TOFF peuvent être spécifiées avec les paramètres machine suivants :

- MD21194 \$MC_TOFF_VELO (vitesse, correction en ligne dans le sens de l'outil)
- MD21196 \$MC_TOFF_ACCEL (accélération, correction en ligne dans le sens de l'outil)

Plus d'informations sur l'activation de la fonction :

Plus d'informations

Manuel de programmation Programmation CN

Applications dans les actions synchrones

Dans les actions synchrones, les corrections de longueur d'outil peuvent être appliquées dans les trois dimensions par le biais de la variable système \$AA_TOFF. Les trois noms d'axes géométriques X, Y, Z servent d'indices. Les trois sens de correction peuvent être actifs simultanément.

Lorsque la transformation d'orientation est active ou qu'un organe porte-outil orientable est actif, les corrections sont effectives dans les différents sens de l'outil. Avant l'activation ou la désactivation d'une transformation, un déplacement forcé doit être désactivé avec TOFFOF.

Après la désactivation de la correction de longueur d'outil dans une dimension, la valeur des variables système \$AA_TOFF dans cette dimension est égale à 0.

Mode d'action de la correction dans le sens de l'outil

Les corrections de longueur d'outil ne modifient pas les paramètres d'outil, mais elles sont prises en compte dans la transformation ou l'organe porte-outil orientable de manière à ce qu'elles se reflètent dans le système de coordonnées d'outil.

Le paramètre machine suivant permet de définir pour chaque dimension si la correction de longueur d'outil indiquée dans \$AA_TOFF doit être prise en compte de façon absolue ou relative (en intégratif) :

MD21190 \$MC_TOFF_MODE (mode d'action de la correction dans le sens de l'outil)

La valeur actuelle de la correction de longueur d'outil peut être lue avec la variable système \$AA_TOFF_VAL.

Remarque

Une évaluation des variables \$AA_TOFF_VAL n'a de sens qu'en liaison avec une transformation d'orientation active ou avec un organe porte-outil actif.

Exemples

Activation de la correction de longueur d'outil en ligne

Paramètres machine pour la correction de longueur d'outil en ligne :

- MD21190 \$MC_TOFF_MODE = 1
- MD21194 \$MC_TOFF_VEL[0] = 10000
- MD21194 \$MC_TOFF_VEL[1] = 10000
- MD21194 \$MC_TOFF_VEL[2] = 10000
- MD21196 \$MC_TOFF_ACC[0] = 1
- MD21196 \$MC_TOFF_ACC[1] = 1
- MD21196 \$MC_TOFF_ACC[2] = 1

Activer la correction de longueur d'outil en ligne dans le programme pièce :

Code de programme

```
N5 DEF REAL XOFFSET
; Activer la transformation d'orientation
N10 TRAORI
; Activer la correction de longueur d'outil en direction Z
N20 TOFFON(Z)
; Correction de longueur d'outil en direction Z : 10 mm
N30 WHEN TRUE DO $AA_TOFF[Z] = 10
G4 F5
...
; Action synchrone statique : Correction de longueur d'outil en direction X
; en fonction de la position de l'axe X2 dans le SCP
N50 ID=1 DO $AA_TOFF[X] = $AA_IW[X2]
```

4.3 Variable système pour actions synchrones

Code de programme

```
G4 F5
...
; Memento : correction de longueur d'outil totale actuelle en direction X
N100 XOFFSET = $AA_TOFF_VAL[X]
; Faire revenir à 0 la correction de longueur d'outil en direction X
N120 TOFFON(X, -XOFFSET)
G4 F5
```

Désactivation de la correction de longueur d'outil en ligne

Code de programme

```
; Activer la transformation d'orientation
N10 TRAORI
; Activer la correction de longueur d'outil en direction X
N20 TOFFON(X)
; Correction de longueur d'outil en direction X : 10 mm
N30 WHEN TRUE DO $AA_TOFF[X] = 10
G4 F5
...
; Supprimer la correction de longueur d'outil en direction X
; Aucun axe n'est déplacé. L'offset de position est
; ajouté à la position actuelle dans le SCP en fonction de l'orientation
; actuelle.
N80 TOFFOF(X)
N90 TRAFOOF
```

Activation et désactivation dans le programme pièce

La correction de longueur d'outil en ligne s'active avec `TOFFON` et se désactive avec `TOFFOF` dans le programme pièce. Dès que la fonction est activée, une valeur d'offset, par ex. `TOFFON(Z, 25)`, peut être indiquée pour les différents sens de correction ; cette valeur fait immédiatement l'objet d'un mouvement de correction. L'état de la correction de longueur d'outil en ligne est affiché sur l'interface CN/AP avec les signaux suivants :

- `<Chan>.basic.in.toffonActive` (TOFF actif)
- `<Chan>.basic.in.toffonMotionActive` (mouvement TOFF actif)

Remarque

La correction de longueur d'outil en ligne reste inactive tant qu'elle n'est pas à nouveau activée avec `TOFFON` dans le programme pièce.

Comportement après Reset et Power On

Le comportement après Reset peut être réglé avec le paramètre machine :

MD21190 \$MC_TOFF_MODE, bit 0 = <valeur> (mode d'action de la correction dans le sens de l'outil)

Valeur	Signification
0	L'offset de longueur d'outil \$AA_TOFF est désactivé lors du Reset
1	L'offset de longueur d'outil \$AA_TOFF est conservé lors d'un Reset

Cela est toujours nécessaire avec les actions synchrones statiques IDS=<numéro> DO \$AA_TOFF[n]=<valeur>, car, dans le cas contraire, une nouvelle correction de longueur d'outil se produirait immédiatement.

De même, une transformation ou un organe porte-outil orientable peut être désactivé **après** le Reset avec le paramètre machine suivant :

MD20110 \$MC_RESET_MODE_MASK (position de base après un Reset)

Là encore, la correction de longueur d'outil doit être supprimée.

Lorsqu'une correction de longueur d'outil doit rester active au-delà d'un Reset, et qu'un changement de transformation ou un changement de l'organe porte-outil orientable a lieu, l'alarme 21665 "Canal %1 \$AA_TOFF[] remis à zéro" est émise. La correction de longueur d'outil est alors mise à 0.

Après un Power On, tous les offsets de longueur d'outil sont mis à 0.

La fonction est désactivée après un POWER ON.

Comportement lors d'un changement de mode de fonctionnement

La correction de longueur d'outil reste active au-delà d'un changement de mode de fonctionnement. La correction est exécutée dans tous les modes de fonctionnement à l'exception de JOG et REF.

Si, lors du changement de mode de fonctionnement, le déplacement correspondant à une correction de longueur d'outil est exécuté en raison de \$AA_TOFF[], la commutation du mode de fonctionnement a lieu seulement après l'exécution dudit déplacement. L'alarme 16907 "Canal %1 Action %2 <ALNX> possible uniquement à l'état de suspension de programme" s'affiche.

Comportement en mode REPOS

La correction de longueur d'outil est active dans le mode de fonctionnement REPOS.

Autres conditions à prendre en compte

En présence d'un offset de longueur d'outil, les autres conditions suivantes sont à prendre en compte :

- Une transformation doit être désactivée avec TRAFOOF
- Avant l'activation d'une transformation dans le programme pièce, un offset de longueur d'outil actif doit être supprimé avec TOFFOF.

4.3 Variable système pour actions synchrones

- Lors de la commutation de CP vers PTP, une transformation est coupée. Un offset de longueur d'outil doit être supprimé **avant** la commutation. Lors d'un passage à un déplacement manuel spécifique à l'axe en mode JOG, si une correction de longueur d'outil est active, le basculement sur PTP n'est pas exécuté. CP reste actif jusqu'à ce que la correction de longueur d'outil soit supprimée avec TOFFOF.
- Avant une permutation d'axe géométrique, un offset de longueur d'outil actif en direction de l'axe géométrique doit être supprimé avec TOFFOF.
- Avant un changement de plan, un offset de longueur d'outil actif doit être supprimé avec TOFFOF.
- Les instructions TOFFON et TOFFOF ne sont pas collectées lors d'une recherche de bloc et ne sont pas transmises dans le bloc d'action.

Signaux AP

CN → AP

Basic Program Plus	Basic Program	
<Chan>.basic.in.toffonActive	LBP_Chan*.E_TOFF	DB21,DBX318.2
<Chan>.basic.in.toffonMotionActive	LBP_Chan*.E_TOFFmovem	DB21,DBX318.3

4.3.19 Bloc actuel dans l'interpolateur (\$AC_BLOCKTYPE, \$AC_BLOCKTYPEINFO, \$AC_SPLITBLOCK)

Les variables système suivantes permettent de lire dans les actions synchrones les informations sur le bloc en cours de traitement durant l'exécution des blocs.

\$AC_BLOCKTYPE et \$AC_BLOCKTYPEINFO

La variable système \$AC_BLOCKTYPE contient le type de bloc ou l'identificateur de la fonction ayant généré le bloc.

La variable système \$AC_BLOCKTYPEINFO contient, en plus du type de bloc (position des milliers), la cause spécifique à la fonction de la génération du bloc intermédiaire.

\$AC_BLOCKTYPE		\$AC_BLOCKTYPEINFO	
Valeur	Signification : Le bloc actuel a été généré en raison de ...	Valeur	Signification
0	Bloc programmé !	-	-
1	CN en tant que bloc intermédiaire	1000	Ne contient aucune autre information
2	Chanfreins/arrondis	2001	complétée
		2002	Cercle
3	Accostage/retrait en douceur (ARD)	3001	accostage sur une droite
		3002	accostage sur un quart de cercle
		3003	accostage sur un demi-cercle

4.3 Variable système pour actions synchrones

\$AC_BLOCKTYPE		\$AC_BLOCKTYPEINFO	
Valeur	Signification : Le bloc actuel a été généré en raison de ...	Valeur	Signification
4	Correction d'outil	4001	bloc d'accostage après STOPRE
		4002	Blocs de liaison dans le point d'intersection non trouvé
		4003	Cercle sous forme de point dans les angles rentrants (uniquement dans TRACYL)
		4004	Arc de contournement (ou section conique) dans les angles saillants
		4005	Blocs d'accostage dans l'inhibition de la correction
		4006	Blocs d'accostage dans l'activation répétée de la correction de rayon d'outil
		4007	Dédoublage du bloc en raison d'une courbure trop forte
		4008	Blocs de compensation pour le fraisage en bout 3D (vecteur d'outil parallèle au vecteur de surface)
5	Arrondissement	5001	Contour d'arrondissement par G641
		5002	Contour d'arrondissement par G642
		5003	Contour d'arrondissement par G643
		5004	Contour d'arrondissement par G644
6	Asservissement tangentiel (TLIFT)	6001	Déplacement linéaire de l'axe tangentiel sans mouvement de retrait
		6002	Déplacement non linéaire de l'axe tangentiel (polynôme) sans mouvement de retrait
		6003	Mouvement de retrait : le déplacement de l'axe tangentiel et le mouvement de retrait commencent simultanément
		6004	Mouvement de retrait : l'axe tangentiel ne commence que lorsqu'une position de retrait donnée est atteinte
7	Segmentation des déplacements	7001	la segmentation du déplacement est programmée sans que le poinçonnage ou le grignotage ne soit actif
		7002	la segmentation du déplacement est programmée avec un poinçonnage ou un grignotage actif
		7003	la segmentation du déplacement est automatiquement générée en interne
8	Cycle de compilation	x	x: ID de l'application des cycles de compilation ayant créé le bloc
9	Interpolation d'orientation par rapport à la trajectoire (ORIPATH/ORIOTC)	9000	Interpolation de l'orientation de l'outil avec ORIPATH
		9001	Interpolation de la rotation de l'outil avec ORIOTC
10	Traitement des pôles avec la transformation d'orientation	10 000	Positionnement par anticipation de l'axe polaire
		10001	Traversée du cône polaire

\$AC_SPLITBLOCK

La variable système \$AC_SPLITBLOCK permet de déterminer si un bloc généré en interne ou un bloc programmé mais tronqué par la CN est présent.

\$AC_SPLITBLOCK	
Valeur	Signification :
0	Bloc programmé. Un bloc généré par le compresseur est également traité en tant que bloc programmé.
1	Bloc généré en interne ou bloc d'origine tronqué
3	Dernier bloc d'une chaîne de blocs générés en interne ou de blocs d'origine tronqués

Exemple

Actions synchrones pour le comptage de blocs d'arrondissement.

L'interrogation de la variable système \$AC_TIMEC == 0 (périodes d'appel de l'interpolateur depuis le début du bloc) permet de s'assurer que le type de bloc n'est déterminé qu'une fois au début du bloc.

Code de programme	Commentaire
\$AC_MARKER[0]=0	; Compteur pour tous les blocs d'arrondissement
\$AC_MARKER[1]=0	; Compteur pour les blocs d'arrondissement G641
\$AC_MARKER[2]=0	; Compteur pour les blocs d'arrondissement G642
...	
; Action synchrone pour le comptage de tous les blocs d'arrondissement	
ID=1 WHENEVER (\$AC_TIMEC==0) AND (\$AC_BLOCKTYPE==5) DO	
\$AC_MARKER[0] = \$AC_MARKER[0] + 1	
...	
; Action synchrone pour le comptage des blocs d'arrondissement G641	
ID=2 WHENEVER (\$AC_TIMEC==0) AND (\$AC_BLOCKTYPEINFO==5001) DO	
\$AC_MARKER[1] = \$AC_MARKER[1]+1	
...	
; Action synchrone pour le comptage des blocs d'arrondissement G642	
ID=3 WHENEVER (\$AC_TIMEC==0) AND (\$AC_BLOCKTYPEINFO==5002) DO	
\$AC_MARKER[2] = \$AC_MARKER[2] + 1	
...	

4.3.20 Initialisation de variables de tableau (SET, REP)

Fonction

Les variables de champ, qu'il est possible d'écrire dans les actions synchrones, peuvent également être initialisées à l'aide des instructions SET et REP dans les actions synchrones.

Exemple

Affectation de valeurs à des variables utilisateur INTEGER pour les actions synchrones dans les blocs de données SGD (SYG_IS)

Condition : MD18661 \$MM_NUM_SYNACT_GUD_INT[<x>] ≥ 5

Code de programme	
PROC MAIN	
N10 DEF INT SYG_IS[4]	
...	
WHEN TRUE DO SYG_IS[0]=REP(1,3)	; Lorsque la condition est remplie : régler sur la valeur "1" les variables utilisateur INTEGER 1 à 3 pour les actions synchrones dans les blocs de données SGUD.
WHEN TRUE DO SYG_IS[3]=SET(2,4)	; Lorsque la condition est remplie : régler sur la valeur "2" ou "4" les variables utilisateur INTEGER 4 et 5 pour les actions synchrones dans les blocs de données SGUD.
...	

Plus d'informations

Informations détaillées concernant SET et REP : → manuel de programmation CN

4.3.21 Variables système spécifiques à la rectification (\$AC_IN_KEY_G...)

Dans la technologie de rectification, les signaux d'entrée doivent être détectés et les actions correspondantes être intégrées à l'exécution du programme de manière asynchrone au processus d'usinage. Pour cela, les variables système et les signaux d'interface CN/AP suivants sont disponibles :

Variable système	Interface CN/AP DB21, ...	Description
Communication interne à la CN		
\$AC_IN_KEY_G_ENABLE[1 ... 8] ¹	---	Déblocage du signal d'entrée côté CN
Communication CN → AP ²⁾		
\$AC_IN_KEY_G_IENABLE[1 ... 8] ¹⁾	<Chan>.grinding.in.signalEnabled	Déblocage du signal d'entrée
\$AC_IN_KEY_G_RUN_OUT[1 ... 8]	<Chan>.grinding.in.requested	Demande de déblocage de l'action côté CN (facultatif)
Communication AP → CN ³⁾		
\$AC_IN_KEY_G[1 ... 8]	<Chan>.grinding.out.set	Signal d'entrée
---	<Chan>.grinding.out.disable-Signal ¹⁾	Blocage du signal d'entrée côté AP

4.3 Variable système pour actions synchrones

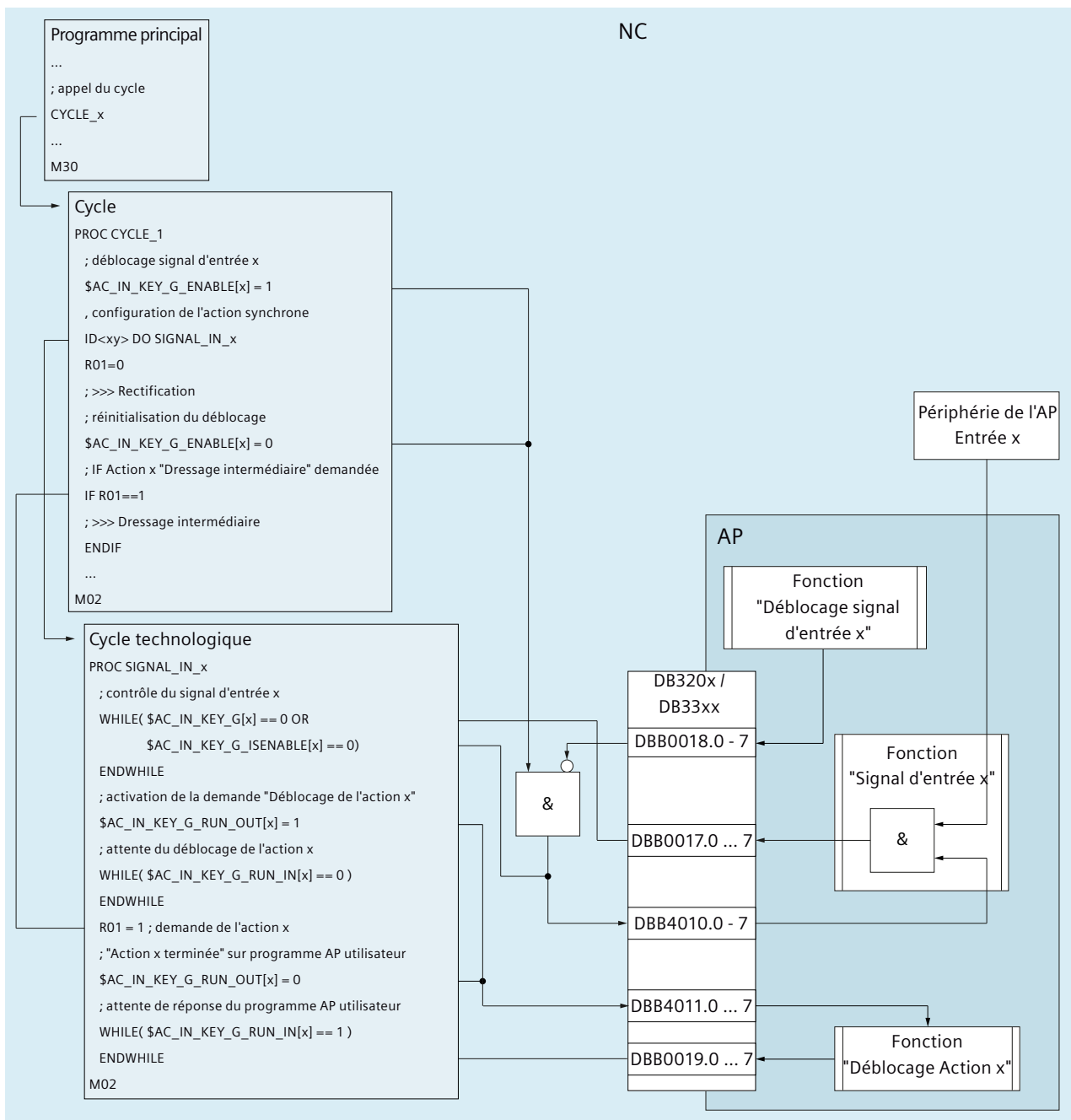
Variable système	Interface CN/AP DB21, ...	Description
\$AC_IN_KEY_G_RUN_IN[1 ... 8]	<Chan>.grinding.out.ackFuncExecution	Déblocage de l'action côté AP (facultatif)
1) Par l'opération logique ET côté CN du signal de déblocage CN dans \$AC_IN_KEY_G_ENABLE et du signal de validation AP NOT (<Chan>.grinding.out.disableSignal), le signal de validation est reproduit dans \$AC_IN_KEY_G_IENABLE et <Chan>.grinding.in.signalEnabled. 2) Le contenu de la variable système est transféré vers l'interface CN/AP 3) Le contenu de l'interface CN/AP est transféré vers la variable système		

Exemple

Valeurs par défaut

- Un cycle (CYCLE_1) dans lequel sont exécutés aussi bien la rectification que le dressage intermédiaire est lancé à partir d'un programme principal.
- L'action "Dressage intermédiaire" peut être demandée de manière asynchrone à l'usinage par un opérateur via un signal d'entrée de la périphérie de l'AP.
- La détection du signal d'entrée et la demande de l'action s'opèrent dans un cycle technologique. Le cycle technologique ("SIGNAL_IN_x") est appelé dans la partie action de l'action synchrone (ID <xy>) configurée dans le cycle.

La figure ci-dessous illustre le schéma d'appel, les instructions pertinentes et le flux de signaux.



- ① <Chan>.grinding.out.disableSignal
- ② <Chan>.grinding.out.set
- ③ <Chan>.grinding.in.signalEnabled
- ④ <Chan>.grinding.in.requested
- ⑤ <Chan>.grinding.out.ackFuncExecution

Les fonctions côté AP, comme "Déblocage Signal d'entrée x", doivent être mises à disposition par le programme AP utilisateur.

Déroulement

- Programme principal
 - Appel du cycle "CYCLE_1"
- Cycle "CYCLE_1"
 - Mise à 1 du déblocage du signal d'entrée x (\$AC_IN_KEY_G_ENABLE)
 - Configuration de l'action synchrone avec le cycle technologique "SIGNAL_IN_x"
 - Initialisation du déclencheur pour l'action x "Dressage intermédiaire" (R01)
 - Exécution de la rectification
- Cycle technologique "SIGNAL_IN_x" (parallèle au cycle)
 - Détection du signal d'entrée (\$AC_IN_KEY_G) lorsque le déblocage est actif (\$AC_IN_KEY_G_IENABLE)
 - Demande du déblocage de l'action x par l'AP (\$AC_IN_KEY_G_RUN_OUT)
 - Attente du déblocage de l'action x par l'AP (\$AC_IN_KEY_G_RUN_IN)
 - Demande de l'action x dans le cycle (R01)
 - Signalisation en retour de la demande à l'AP (\$AC_IN_KEY_G_RUN_OUT)
 - Attente de l'acquittement par l'AP (\$AC_IN_KEY_G_RUN_IN)
- Cycle "CYCLE_1" (une fois la rectification terminée)
 - Mise à 0 du déblocage du signal d'entrée x (\$AC_IN_KEY_G_ENABLE)
 - Si l'action x a été demandée ⇒ exécution du dressage intermédiaire
 - ... (exécuter éventuellement à nouveau la séquence rectification / dressage intermédiaire, etc.)
- Programme AP utilisateur
 - Fonction "Déblocage Signal d'entrée x" :
Mise à 1 du signal d'interface <Chan>.grinding.out.disableSignal
 - Fonction "Signal d'entrée x" :
Combinaison (opération logique ET) du signal d'entrée de la périphérie de l'AP et du déblocage du signal d'entrée (<Chan>.grinding.in.signalEnabled) et mise à 1 du résultat dans l'interface (<Chan>.grinding.out.set)
 - Fonction "Déblocage Action x" :
Vérification/mise à 1 des signaux d'interface <Chan>.grinding.in.requested / <Chan>.grinding.out.ackFuncExecution

Signaux AP**CN → AP**

Basic Program Plus	Basic Program	
<Chan>.grinding.in.signalEnabled	LBP_Chan*.E_InKeyGIsEn1-8	DB21,DBX390.0..7
<Chan>.grinding.in.requested	LBP_AxisX*.E_InKeyGRunOut1-8	DB21,DBX391.0..7
<Chan>.grinding.out.set	LBP_Chan*.A_InKeyG<n>	DB21,DBX385.0..7

Basic Program Plus	Basic Program	
<Chan>.grinding.out.disableSignal	LBP_Chan*.A_InKeyGEn<n>	DB21,DBX386.0..7
<Chan>.grinding.out.ackFuncExecution	LBP_Chan*.A_InKeyGRunIn<n>	DB21,DBX387.0..7

4.3.22 État du blocage d'action synchrone (\$AC_SYNA_STATE)

La variable système \$AC_SYNA_STATE indique, sous forme de valeur codée sur bits, si une action synchrone est bloquée par l'AP (Page 131) ou une action synchrone (Page 131).

Les bits ont la signification suivante :

Bit	Valeur	Signification
0	0	Aucun blocage
	1	Blocage par l'AP ou une action synchrone
1	0	Aucun blocage par l'AP
	1	Blocage par l'AP
2	0	Aucun blocage par une action synchrone
	1	Blocage par une action synchrone

Les blocages par l'AP ou une action synchrone ont une priorité différente. L'ordre suivant s'applique :

Priorité 1 (priorité la plus élevée) :	Blocage par l'AP pour l'ensemble du canal (⇒ toutes les actions synchrones du canal sont bloquées)
Priorité 2 :	Blocage par une action synchrone
Priorité 3 :	Blocage unique par l'AP (⇒ une seule action synchrone du canal est bloquée)

\$AC_SYNA_STATE fournit toujours uniquement la valeur du blocage ayant la priorité la plus élevée, même lorsque des blocages par l'AP **et** par une action synchrone sont actifs simultanément :

Etat	Priorité la plus élevée	\$AC_SYNA_STATE			
		Bit 2	Bit 1	Bit 0	Valeur codée sur bits
Un blocage par l'AP pour l'ensemble du canal est actif. En outre, un blocage par une action synchrone peut être actif.	Blocage par l'AP pour l'ensemble du canal	0	1	1	3
Aucun blocage par l'AP pour l'ensemble du canal n'est actif. Un blocage par une action synchrone est actif. En outre, un blocage unique par l'AP peut être actif.	Blocage par une action synchrone	1	0	1	5

4.4 Variables définies par l'utilisateur pour actions synchrones

Etat	Priorité la plus élevée	\$AC_SYNA_STATE			
		Bit 2	Bit 1	Bit 0	Valeur codée sur bits
Aucun blocage par l'AP pour l'ensemble du canal n'est actif. Aucun blocage par une action synchrone n'est actif. Un blocage unique par l'AP est actif.	Blocage unique par l'AP	0	1	1	3
Aucun blocage n'est actif.	-	0	0	0	0

4.4 Variables définies par l'utilisateur pour actions synchrones

Variables GUD à action synchrone

Outre des variables système spécifiques, il est également possible d'utiliser des variables utilisateur globales prédéfinies à action synchrone dans les actions synchrones (GUD à action synchrone). Le nombre de variables GUD à action synchrone mises à disposition de l'utilisateur est paramétré selon le type de données et le mode d'accès dans les paramètres machine suivants :

- PM18660 \$MM_NUM_SYNACT_GUD_REAL[<x>] = <nombre>
- PM18661 \$MM_NUM_SYNACT_GUD_INT[<x>] = <nombre>
- PM18662 \$MM_NUM_SYNACT_GUD_BOOL[<x>] = <nombre>
- PM18663 \$MM_NUM_SYNACT_GUD_AXIS[<x>] = <nombre>
- PM18664 \$MM_NUM_SYNACT_GUD_CHAR[<x>] = <nombre>
- PM18665 \$MM_NUM_SYNACT_GUD_STRING[<x>] = <nombre>

L'indice <x> indique le bloc de données (droits d'accès), et la valeur <nombre> indique le nombre de variables GUD à action synchrone du type de données correspondant (REAL, INT, etc.). Une variable de tableau unidimensionnelle est ensuite créée dans le bloc de données respectif pour chaque type de données, avec le schéma de nommage suivant : SYG_<type de données><droit d'accès>[<indice>] :

Indice <x>		Type de données (PM18660 ... PM18665)					
	Bloc	REAL	INT	BOOL	AXIS	CHAR	STRING
0	SGUD	SYG_RS[i]	SYG_IS[i]	SYG_BS[i]	SYG_AS[i]	SYG_CS[i]	SYG_SS[i]
1	MGUD	SYG_RM[i]	SYG_IM[i]	SYG_BM[i]	SYG_AM[i]	SYG_CM[i]	SYG_SM[i]
2	UGUD	SYG_RU[i]	SYG_IU[i]	SYG_BU[i]	SYG_AU[i]	SYG_CU[i]	SYG_SU[i]
3	GUD4	SYG_R4[i]	SYG_I4[i]	SYG_B4[i]	SYG_A4[i]	SYG_C4[i]	SYG_S4[i]
4	GUD5	SYG_R5[i]	SYG_I5[i]	SYG_B5[i]	SYG_A5[i]	SYG_C5[i]	SYG_S5[i]
5	GUD6	SYG_R6[i]	SYG_I6[i]	SYG_B6[i]	SYG_A6[i]	SYG_C6[i]	SYG_S6[i]
6	GUD7	SYG_R7[i]	SYG_I7[i]	SYG_B7[i]	SYG_A7[i]	SYG_C7[i]	SYG_S7[i]
7	GUD8	SYG_R8[i]	SYG_I8[i]	SYG_B8[i]	SYG_A8[i]	SYG_C8[i]	SYG_S8[i]

4.4 Variables définies par l'utilisateur pour actions synchrones

Indice <x>		Type de données (PM18660 ... PM18665)					
8	GUD9	SYG_R9[i]	SYG_I9[i]	SYG_B9[i]	SYG_A9[i]	SYG_C9[i]	SYG_S9[i]
avec i = 0 à (<nombre> - 1) bloc : _N_DEF_DIR/_N_ ... _DEF, par ex. pour SGUD ⇒ _N_DEF_DIR/_N_SGUD_DEF							

Propriétés

Les variables GUD à action synchrone présentent les propriétés suivantes :

- La lecture et l'écriture des variables GUD à action synchrone sont possibles dans les actions synchrones et les programmes pièce / cycles
- L'accès aux variables GUD à action synchrone peut être réalisé avec BTSS
- L'affichage des variables GUD à action synchrone s'effectue dans l'interface utilisateur de l'IHM, dans la zone de commande "Paramètres"
- Les variables GUD à action synchrone peuvent être utilisées sur l'IHM dans l'assistant, dans la vue des variables et dans le journal des variables
- La dimension de tableau pour les variables GUD à action synchrone de type STRING est définie sur 32 (31 caractères + \0).
- Même si aucun fichier de définition n'a été créé manuellement pour les données utilisateur globales (GUD), la lecture des variables GUD à action synchrone définies dans des paramètres machine est possible dans le bloc GUD correspondant de l'IHM.

Remarque

La définition de variables utilisateur (GUD, PUD, LUD) portant le même nom que des variables GUD à action synchrone (DEF . . . SYG_xy) n'est donc possible que si aucune variable GUD à action synchrone de même nom n'a été paramétrée (PM18660 - PM18665). Ces variables GUD définies par l'utilisateur ne peuvent **pas** être utilisées dans des actions synchrones.

Droits d'accès

Les droits d'accès définis dans un fichier de définition GUD restent valables et s'appliquent exclusivement aux variables GUD définies dans ce fichier de définition GUD.

Comportement à la suppression

Lorsque le contenu d'un fichier de définition GUD donné est réactivé, l'ancien bloc de données GUD est d'abord supprimé dans le système de fichier actif. Les variables GUD à action synchrone configurées sont également réinitialisées. Cette procédure est également possible sur l'IHM, dans le groupe fonctionnel "Services" > "Définir et activer des données utilisateur (GUD)".

4.5 Éléments de langage pour les actions synchrones et les cycles technologiques

Les éléments de langage suivants peuvent être utilisés dans les actions synchrones et les cycles technologiques :

Adresses fixes	
L	Numéro de sous-programme
F	Avance
S ^{1) 2)}	Broche
M ^{1) 2)}	Fonction M
H ¹⁾	Fonction H
1) Chapitre : "Sortie des fonctions auxiliaires M, S et H en direction de l'AP (Page 74)"	
2) Chapitre : "Déplacement de broches (M, S, SPOS) (Page 100)"	

Adresses fixes avec extension axiale : divers	
POS	Déplacement d'axes en position (Page 86)
POSA	Axe de positionnement modal
SPOS	Positionnement de la broche (Page 100)
MOV ¹⁾	Déplacement d'axes, sans fin (Page 93)
FA	Avance axiale (Page 93)
OVRA	Correction axiale
ACC	Accélération axiale
MEASA	Mesure axiale avec effacement de la distance restant à parcourir
MEAWA	Mesure axiale sans effacement de la distance restant à parcourir (Page 117)
MEAC	Mesure cyclique (Page 117)
SCPARA	Commutation de jeux de paramètres
VELOLIMA	Limitation axiale de la vitesse linéaire / vitesse de rotation
ACCLIMA	Limitation axiale de l'accélération
JERKLIMA	Limitation axiale des à-coups
1) non autorisé dans les cycles technologiques	

Adresses réglables : accostage de butée fixe ¹⁾	
FXS	Activation de l'accostage de butée fixe
FXST	Limite de couple pour l'accostage de butée fixe
FXSW	Fenêtre de surveillance pour l'accostage de butée fixe
FOC	Limitation de couple/force à effet non modal
FOCON	Activation du déplacement avec réduction de couple/force
FOCOF	Désactivation du déplacement avec réduction de couple/force
1) Chapitre : "Accostage d'une butée (FXS, FXST, FXSW, FOCON, FOCOF, FOC) (Page 120)"	

4.5 Éléments de langage pour les actions synchrones et les cycles technologiques

Adresses réglables : Couplages > Couplage générique ¹⁾	
CPBC	Critère de changement de bloc pour couplage actif
CPDEF	Création d'un module de couplage
CPDEL	Suppression d'un module de couplage
CPFMOF	Comportement de l'axe asservi lors de la désactivation du couplage
CPFMON	Comportement de l'axe asservi lors de l'activation du couplage
CPFMSON	Mode de synchronisation lors du couplage
CPFPOS	Position synchrone de l'axe asservi lors de l'activation
CPFRS	Système de référence pour le module de couplage de l'axe asservi
CPLA ²⁾	Affectation d'un axe en tant qu'axe pilote à un axe asservi
CPLCTID	Numéro de la table de courbes pour le couplage de l'axe asservi
CPLDEF	Définition de la référence : Axe pilote / axe asservi
CPLDEL	Suppression de la référence : Axe pilote / axe asservi
CPLDEN	Facteur de couplage : Compteur
CPLNUM	Facteur de couplage : Dénominateur
CPLDYPRIO	Priorité de l'axe pilote en cas de limitation dynamique
CPLDYVLL	Limitation du déplacement forcé de l'axe pilote : Limite inférieure
CPLDYVLU	Limitation du déplacement forcé de l'axe pilote : Limite supérieure
CPLINSC	Facteur de normalisation pour la valeur d'entrée de l'axe pilote
CPLINTR	Valeur de décalage pour la valeur d'entrée de l'axe pilote
CPLOF	Couplage axe pilote / axe asservi : Arrêt
CPLON	Couplage axe pilote / axe asservi : Mise en marche
CPLOUTSC	Normalisation de la valeur de départ
CPLOUTTR	Décalage de la valeur de départ
CPLPOS	Position synchrone de l'axe pilote lors de l'activation
CPLSETVAL	Type de couplage de l'axe asservi avec l'axe pilote
CPMALARM	Définition du comportement aux alarmes
CPMBRAKE ²⁾	Définition de la réaction aux signaux et commandes d'arrêt
CPMPRT	Définition du comportement au démarrage lors du test de programme
CPMRESET	Définition du comportement reset
CPMSTART	Définition du comportement au démarrage
CPMVDI	Définition du comportement relatif aux signaux d'interface CN/AP
CPOF	Désactivation du couplage pour tous les axes pilotes définis
CPON	Activation du couplage pour tous les axes pilotes définis
CPRES ²⁾	Activation des paramètres de couplage définis dans les paramètres machine
CPSETTYPE	Définition des propriétés de couplage fondamentales
CPSYNCOF	Synchronisation de position "grossière"
CPSYNCOF2	Synchronisation de position 2 "grossière"
CPSYNFIP	Synchronisation de position "fine"
CPSYNFIP2	Synchronisation de position 2 "fine"
CPSYNCOV	Synchronisation de vitesse "grossière"

4.5 Éléments de langage pour les actions synchrones et les cycles technologiques

Adresses réglables : Couplages > Couplage générique ¹⁾	
CPSYNFIV	Synchronisation de vitesse "fine"
1) Chapitre : "Couplages (CP..., LEAD..., TRAIL..., CTAB...) (Page 113)"	
2) Actuellement indisponible dans les actions synchrones	

Fonctions G : définition du système de mesure ¹⁾	
G70	Système de mesure anglo-saxon
G71	Système de mesure métrique
G700	Système de mesure anglo-saxon
G710	Système de mesure métrique
1) Chapitre : "Réglage du système de mesure (G70, G71, G700, G710) (Page 90)"	

Sous-programmes prédéfinis : divers	
POLFA	Position de retrait axiale pour axe individuel
POLFC	Position de retrait axiale pour axes de canal
STOPREOF	Annulation de l'arrêt du prétraitement des blocs (Page 84)
RDISABLE	Blocage de lecture programmé (Page 83)
DELDTG	Effacement de la distance restant à parcourir (Page 85)
LOCK	Blocage de l'action synchrone
UNLOCK	Déblocage de l'action synchrone
ICYCON	Cycle technologique : un bloc par période d'appel de l'interpolateur
ICYCOF	Cycle technologique : tous les blocs dans une période d'appel de l'interpolateur
SYNFCT	Évaluation du polynôme (Page 75)
FTOC	Correction fine d'outil (Page 81)
SOFTENDSA	Fin de course logiciel
PROTA	Modification de l'état d'une zone de protection
SETM	Définition d'un repère pour la coordination de canaux (Page 122)
CLEARM	Suppression d'un repère pour la coordination de canaux (Page 122)
RET	Retour du sous-programme au programme principal
GET	Demande d'un axe (Page 94)
RELEASE	Libération de l'axe (Page 94)
AXTOCHAN	Transfert d'un axe à un autre canal (Page 94)
SETAL	Réaction aux erreurs spécifique à l'utilisateur (Page 123)
IPOBRKA	Critère de changement de bloc : Rampe de freinage
ADISPOSA	Fenêtre de tolérance pour le critère de fin de déplacement

Sous-programmes prédéfinis : Couplage > Déplacements conjugués ¹⁾	
TRAILON	Déplacements conjugués actifs
TRAILOF	Déplacements conjugués inactifs
1) Chapitre : "Couplages (CP..., LEAD..., TRAIL..., CTAB...) (Page 113)"	

4.5 Éléments de langage pour les actions synchrones et les cycles technologiques

Sous-programmes prédéfinis : Couplages > Couplage par valeur pilote ¹⁾	
LEADON	Activation du couplage par valeur pilote
LEADOF	Désactivation du couplage par valeur pilote
1) Chapitre : "Couplages (CP..., LEAD..., TRAIL..., CTAB...)" (Page 113)"	

Sous-programmes prédéfinis : Couplages > Couplage du couple de rotation (couplage principal/ subordonné)	
MASLON	Activation du couplage
MASLOF	Désactivation du couplage
MASLDEF	Définition du couplage
MASLDEL	Suppression du couplage
MASLOFS	Désactivation du couplage broche asservie

Fonctions prédéfinies : Couplage > Tables de courbes ¹⁾	
CTAB	Calcule la position de l'axe asservi avec la table de courbes à l'aide de la position de l'axe pilote
CTABINV	Calcule la position de l'axe pilote avec la table de courbes à l'aide de la position de l'axe asservi
CTABID	Détermine le numéro de la table de courbes
CTABLOCK	Blocage de la table de courbes
CTABUNLOCK	Déblocage de la table de courbes
CTABISLOCK	Détermine l'état de blocage de la table de courbes
CTABEXISTS	Vérifie l'existence de la table de courbes
CTABMEMTYP	Détermine le lieu de stockage de la table de courbes (mémoire statique/ dynamique)
CTABPERIOD	Détermine la périodicité de la table de courbes
CTABNO	Détermine le nombre de tables de courbes
CTABNOMEM	Détermine le nombre de tables de courbes existantes en fonction du lieu de stockage
CTABSEG	Détermine le nombre de segments de courbes déjà utilisés en fonction du lieu de stockage
CTABSEGID	Détermine le nombre de segments de courbes déjà utilisés en fonction des tables
CTABFSEG	Détermine le nombre de segments de courbes encore possibles en fonction des tables
CTABMSEG	Détermine le nombre de segments de courbes au maximum possibles en fonction du lieu de stockage
CTABPOL	Détermine le nombre de polynômes déjà utilisés en fonction du lieu de stockage
CTABPOLID	Détermine le nombre de polynômes déjà utilisés en fonction des tables
CTABFPOL	Détermine le nombre de polynômes encore possibles en fonction des tables
CTABMPOL	Détermine le nombre de polynômes au maximum possibles en fonction du lieu de stockage

Fonctions prédéfinies : Couplage > Tables de courbes ¹⁾	
CTABTSV	Détermine la valeur asservie au début de la table
CTABTEV	Détermine la valeur asservie à la fin de la table
CTABTSP	Détermine la valeur pilote au début de la table
CTABTEP	Détermine la valeur pilote à la fin de la table
CTABTMIN	Détermine la valeur pilote minimale de la table
CTABTMAX	Détermine la valeur pilote maximale de la table
CTABFNO	Détermine le nombre de tables de courbes encore possibles en fonction du lieu de stockage
CTABSSV	Détermine la valeur de début d'un segment de table pour les axes asservis
CTABSEV	Détermine la valeur de fin d'un segment de table pour les axes asservis
1) Chapitre : "Couplages (CP..., LEAD..., TRAIL..., CTAB...) (Page 113)"	

Fonctions prédéfinies : arithmétique	
SIN	Sinus
ASIN	Arc sinus
COS	Cosinus
ACOS	Arc cosinus
TAN	Tangente
ATAN2	Arc tangente 2
SQRT	racine carrée
POT	Puissance 2 (carré)
TRUNC	Partie entière
ROUND	Arrondi à l'entier suivant
ROUNDUP	Arrondi d'une valeur entrée à l'entier suivant
ABS	Valeur absolue
LN	Logarithme naturel
EXP	Fonction exponentielle
MINVAL	Valeur la plus petite de deux valeurs
MAXVAL	Valeur la plus grande de deux valeurs
BOUND	Vérification de la plage de valeurs définie

Fonctions prédéfinies : Valeurs actuelles des paramètres machine	
GETMDACT	Détermine la valeur actuelle du paramètre machine
GETMDPEAK	Détermine la valeur maximale du paramètre machine depuis le dernier RESETPEAK
GETMDLIM	Détermine la valeur limite minimale ou maximale du paramètre machine
RESETPEAK	Réinitialise la valeur maximale pour GETMDPEAK

4.6 Éléments de langage pour cycles technologiques uniquement

Fonctions prédéfinies : conversions de format	
ITOR	INT → REAL
RTOI	REAL → INT
RTOB	REAL → BOOL
BTOR	BOOL → REAL
ITOB	INT → BOOL
BTOI	BOOL → INT

Fonctions prédéfinies : divers	
POSRANGE	Position dans la zone de référence prescrite (Page 92)
PRESETON	Forçage de valeur réelle avec perte de l'état de référencement (Page 101)
PRESETONS	Forçage de valeur réelle sans perte de l'état de référencement (Page 107)

Procédures prédéfinies : divers	
CANCELSUB	Interruption du niveau de sous-programme actuel (Page 124)

Plus d'informations

Une description détaillée des éléments de langage CN figure dans le Manuel de programmation CN.

4.6 Éléments de langage pour cycles technologiques uniquement

Les éléments de langage suivants peuvent être utilisés uniquement dans les cycles technologiques :

Instructions de saut	
IF	Branchement
GOTO	Saut vers étiquette, recherche vers l'avant, puis vers l'arrière
GOTOF	Saut vers étiquette, recherche vers l'avant
GOTOB	Saut vers étiquette, recherche vers l'arrière

Fin du programme	
M02	Fin du programme
M17	Fin du programme
M30	Fin du programme
RET	Fin du programme

4.7 Actions synchrones

4.7.1 Sortie des fonctions auxiliaires M, S et H en direction de l'AP

Instant de sortie

La sortie de fonctions auxiliaires de type M, S et H est possible à partir des actions synchrones. La sortie en direction de l'AP intervient immédiatement, c'est-à-dire directement au cours de la période d'appel de l'interpolateur durant laquelle l'action est exécutée.

Les instants de sortie éventuellement réglés dans les paramètres machine pour les fonctions auxiliaires ne sont pas effectifs pour la sortie à partir des actions synchrones :

- PM11110 \$MN_AUXFU_GROUP_SPEC (spécification des groupes de fonctions auxiliaires)
- PM22200 \$MC_AUXFU_M_SYNC_TYPE (instant de sortie des fonctions M)
- PM22210 \$MC_AUXFU_S_SYNC_TYPE (instant de sortie des fonctions S)
- PM22230 \$MC_AUXFU_H_SYNC_TYPE (instant de sortie des fonctions H)

Nombre maximal

Généralités

La sortie simultanée, c.-à-d. dans un cycle OB40 de l'AP, de 10 fonctions auxiliaires au maximum est possible à partir du programme pièce et des actions synchrones actives d'un canal.

Spécifique aux actions synchrones

Le nombre maximal admissible de fonctions auxiliaires dans la partie action d'une action synchrone est de :

- Fonctions M : 5
- Fonctions S : 3
- Fonctions H : 3

Actions synchrones à effet non modal

Dans les actions synchrones à effet non modal (sans indication d'ID ou IDS), la sortie des fonctions auxiliaires n'est possible qu'en association avec l'indication de fréquence WHEN ou EVERY.

Fonctions M prédéfinies

La sortie des fonctions M prédéfinies n'est généralement pas autorisée dans les actions synchrones.

Exceptions : M3, M4, M5, M40, M41, M42, M43, M44, M45, M70 et M17

Voir aussi

Fréquence (WHENEVER, FROM, WHEN, EVERY) (Page 21)

4.7.2 Lecture et écriture de variables système

Les variables système de la CN sont énumérées dans les Tables de paramètres "Variables système" avec leurs propriétés respectives. Les variables système qui peuvent être lues ou écrites dans la partie action des actions synchrones sont signalées par un "X" dans la ligne correspondante (Read ou Write) de la colonne "AS" (action synchrone).

Remarque

Les variables système qui sont utilisées dans les actions synchrones sont toujours implicitement lues et écrites de façon synchrone à l'exécution.

Informations complémentaires

Tables de paramètres Variables système

4.7.3 Evaluation du polynôme (SYNFCT)**Application**

La fonction SYNFCT permet de lire une variable pendant l'exécution, de l'évaluer via un polynôme et d'écrire le résultat dans une autre variable. Exemples d'application :

- Avance en fonction de la charge de l'entraînement
- Position en fonction d'un signal de capteur
- Puissance du laser en fonction de la vitesse tangentielle

Syntaxe

SYNFCT (<Poly_Nr>, <SysVar_Out>, <SysVar_In>)

Signification

Paramètres	Signification
<Poly_Nr> :	Numéro du polynôme défini avec FCTDEF : $f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3$
<SysVar_Out> :	Variable système, sortie : $\text{<SysVar_Out>} = f(x)$
<SysVar_In> :	Variable système, entrée : $x = \text{<SysVar_In>}$
Pour FCTDEF, voir chapitre "Coefficients/paramètres de polynôme (\$AC_FCT...) (Page 49)"	

Exemple : Superposition additive de l'avance tangentielle

Une valeur de superposition est ajoutée à l'avance programmée (mot F) :

$F_{\text{actif}} = F_{\text{programmé}} + F_{\text{AC}}$

<SysVar_Out>	Signification
\$AC_VC	Correction d'avance additive
\$AA_VC[axe]	Correction d'avance axiale additive

La valeur d'entrée est la valeur réelle de courant \$AA_CURR de l'axe X.

Le point de fonctionnement est fixé à 5 A.

L'avance peut être modifiée de ±100 mm/min, l'écart du courant axial doit alors s'élever à ±1 A.

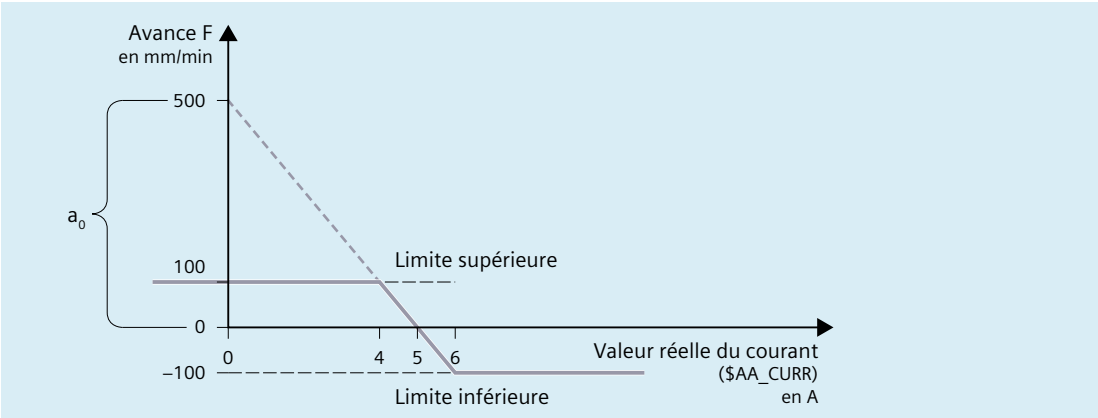


Figure 4-3 Exemple : Influence additive de l'avance tangentielle

Détermination des paramètres de la fonction FCTDEF :

FCTDEF(<Poly_Nr>,<Lo_Limit>,<Up_Limit>,<a_0>,<a_1>,<a_2>,<a_3>)

<Poly_Nr> : = 1 (exemple)

<Lo_Limit> : = -100

<Up_Limit> : = 100

Polynôme : $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$

a_0 : $1 / 100 = 5 / a_0 \Rightarrow a_0 = 500$

a_1 : $100 \text{ mm/min} / -1 \text{ A} = -100 \text{ [mm/min / A]}$

a_2 : = 0 (pas de terme quadratique)

a_3 : = 0 (pas de terme cubique)

Calcul de la valeur de superposition :

SYNFACT(<Poly_Nr>,<SysVar_Out>,<SysVar_In>)

<Poly_Nr> : = 1

<SysVar_Out> : \$AC_VC (correction d'avance additive)

<SysVar_In> : \$AA_CURR (valeur actuelle du courant de l'entraînement)

Programmation :

Code de programme
N100 FCTDEF(1, -100, 100, 500, -100)
N110 ID=1 DO SYNFACT(1, \$AC_VC[X], \$AA_CURR[X])

Exemple : Superposition multiplicative de l'avance tangentielle

L'avance programmée est multipliée par un facteur en pourcentage (correction supplémentaire) :

$$F_{\text{actif}} = F_{\text{programmé}} * \text{Facteur}_{AC}$$

<SysVar_Out>	Signification
\$AC_OVR	Correction tangentielle spécifiable par action synchrone

La valeur d'entrée est la charge d'entraînement en pourcentage \$AA_LOAD de l'axe X.

Le point de fonctionnement est fixé à 100 % pour une charge d'entraînement de 30 %.

Avec une charge de 80 %, l'axe doit être immobilisé.

Un dépassement de +20 % de la vitesse programmée est autorisé.

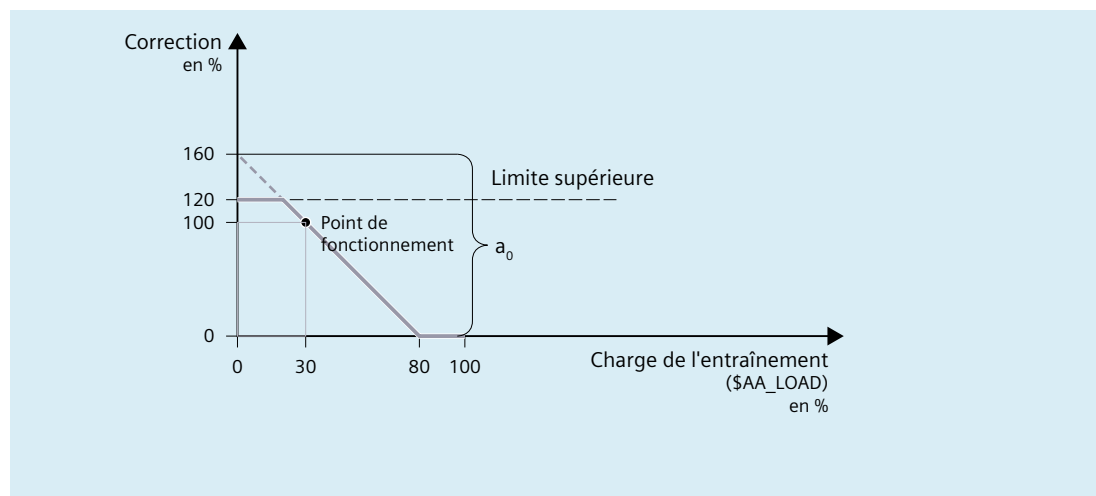


Figure 4-4 Exemple : Influence multiplicative

Détermination des paramètres de la fonction FCTDEF :

FCTDEF(<Poly_Nr>, <Lo_Limit>, <Up_Limit>, a_0, a_1, a_2, a_3)

<Poly_Nr> : = 2 (exemple)

<Lo_Limit> : = 0

<Up_Limit> : = 120

Polynôme : $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$

a_0 : $50 / 100 = 80 / a_0 \Rightarrow a_0 = 160$

a_1 : $= 100 \% / -50 \% = -2$

$a_2 = 0$ (pas de terme quadratique)

$a_3 = 0$ (pas de terme cubique)

Calcul de la valeur de superposition :

`SYNFCT(<Poly_Nr>, <SysVar_Out>, <SysVar_In>)`

`<Poly_Nr> : = 2`

`<SysVar_Out> : $AC_OVR (Correction tangentielle spécifiable par synchronisation)`

`<SysVar_In> : $AA_LOAD (charge d'entraînement)`

Programmation :

Code de programme

N100 FCTDEF(2, 0, 120, 160, -2)

N110 ID=1 DO SYNFCT(2, \$AC_OVR[X], \$AA_LOAD[X])

Exemple : Régulation de la distance

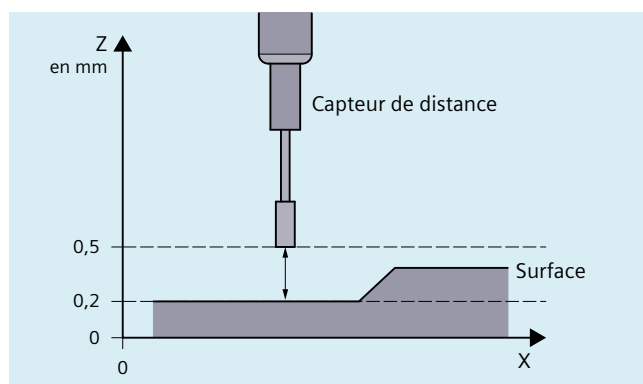


Figure 4-5 Régulation de la distance : Principe

La régulation de distance de l'axe de pénétration Z est réalisée via les fonctions `FCTDEF` et `SYNFCT` ainsi que par les variables système `$AA_OFF` et `$A_INA`.

Conditions supplémentaires :

- La tension analogique du capteur de distance est raccordée via l'entrée analogique \$A_INA[3].
- Les écarts de position sont ajoutés (intégrés) dans \$AA_OFF :
MD36750 \$MA_AA_OFF_MODE, Bit 0 = 1
- En cas de dépassement de 1 mm de la valeur limite supérieure de l'axe Z, l'axe X est arrêté :
SD43350 \$SA_AA_OFF_LIMIT[Z] = 1
Voir également le chapitre "Déplacements forcés (\$AA_OFF) (Page 51)".

Remarque

\$AA_OFF agit dans le système de coordonnées de base (SCB)

La correction s'applique avant la transformation cinématique dans le système de coordonnées de base (SCB). L'exemple ne peut donc **pas** être utilisé pour une régulation de distance dans la direction d'orientation de l'outil (système de coordonnées pièce SCP).

Au sujet de la régulation de distance avec demandes dynamiques élevées ou de la régulation de distance en 3D, voir :

Plus d'informations

Description fonctionnelle Technologies

Réactions spécifiques à l'utilisateur

Lorsque la valeur limite SD43350 \$SA_AA_OFF_LIMIT est atteinte, des réactions spécifiques à l'utilisateur peuvent être déclenchées, par exemple :

- Chapitre "Correction (\$A...OVR) (Page 41)"
- Chapitre "Réaction aux erreurs spécifique à l'utilisateur (SETAL) (Page 123)"

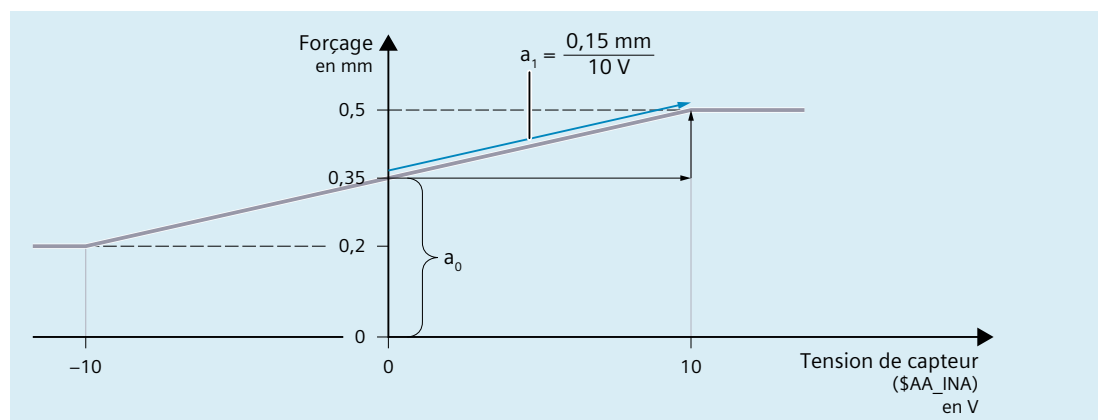


Figure 4-6 Régulation de la distance

Détermination des paramètres de la fonction FCTDEF :

FCTDEF(<Poly_Nr>, <Lo_Limit>, <Up_Limit>, a_0 , a_1 , a_2 , a_3)

<Poly_Nr> : = 1 (exemple)

<Lo_Limit> : = 0,2

<Up_Limit> : = 0,5

Polynôme : $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$

4.7 Actions synchrones

$a_0 :$
 a_1
 a_2
 a_3

$10 / x = 20 / 0,3 \Rightarrow a_0 = x + 0,2 = 0,15 + 0,2 = 0,35$
 $= 0,15 \text{ mm} / 10 \text{ V} = 1,5 * 10^{-2} \text{ mm/V}$
 $= 0 \text{ (pas de terme quadratique)}$
 $= 0 \text{ (pas de terme cubique)}$

Calcul de la valeur de superposition :

```
SYNFCT(<Poly_Nr>,<SysVar_Out>,<SysVar_In>)
<Poly_Nr> :           = 1
<SysVar_Out> :       $AA_OFF (déplacement forcé d'un axe)
<SysVar_In> :        $A_INA (entrée analogique)
```

Programmation :

Code de programme : %_N_AON_SPF	Commentaire
PROC AON	; Régulation de distance "MARCHE"
FCTDEF(1, 0.2, 0.5, 0.35, 1.5 EX-2)	; Définition du polynôme
ID=1 DO SYNFCT(1, \$AA_OFF[Z], \$A_INA[3])	; Régulation de distance
ID=2 WHENEVER \$AA_OFF_LIMIT[Z]<>0 DO \$AA_OVR[X] = 0	; Contrôle des valeurs limites
RET	
ENDPROC	

Code de programme : %_N_AOFF_SPF	Commentaire
PROC AOFF	; Régulation de distance "ARRET"
CANCEL(1)	; Supprimer la régulation de distance
CANCEL(2)	; Supprimer le contrôle des valeurs limites
RET	
ENDPROC	

Code de programme : %_N_MAIN_MPF	Commentaire
N100 \$SA_AA_OFF_LIMIT[Z]=1	
N110 AON	; Régulation de distance "MARCHE"
...	
N200 G1 X100 F1000	
N210 AOFF	; Régulation de distance "ARRET"
M30	

Voir aussi

Correction d'outil en ligne (FTOC) (Page 81)

4.7.4 Correction d'outil en ligne (FTOC)

La fonction FTOC permet, pour corriger un outil en ligne, de réaliser le déplacement forcé d'un axe géométrique en fonction d'une valeur de référence, par exemple la valeur réelle de n'importe quel axe. La valeur de correction est alors calculée à l'aide d'un polynôme défini par `FCTDEF` (voir chapitre "Coefficients/paramètres de polynôme (`$AC_FCT...`) (Page 49)"). Le coefficient a_0 indiqué lors de la définition du polynôme est également évalué par `FTOC`.

Exemple : Usinage et dressage dans la technologie "Rectification"

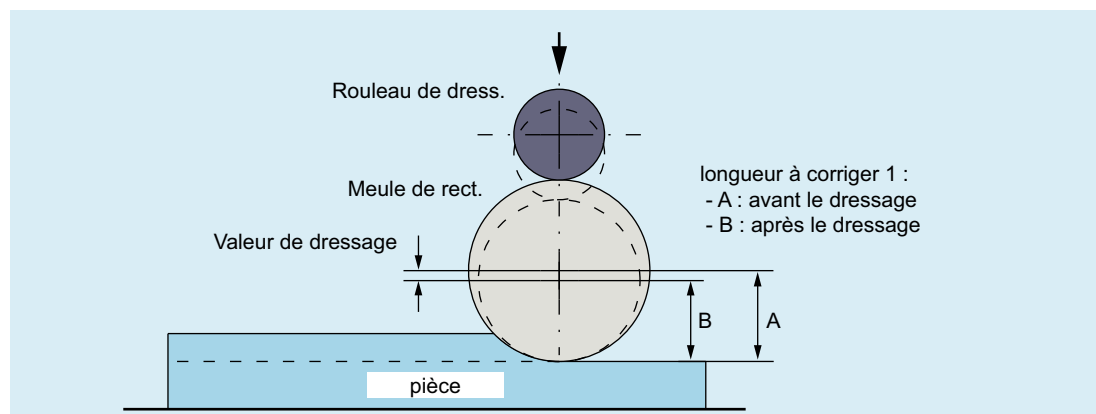


Figure 4-7 Dressage pendant l'usinage avec une meule de dressage

Plus d'informations

Description fonctionnelle Outils

Syntaxe

`FTOC(<Poly_Nr>, <Systemvar>, <Usure>[, <n°_canal>, <n°_broche>])`

Signification

Paramètres	Signification
<code><Poly_Nr></code> :	Numéro du polynôme défini avec <code>FCTDEF</code>
<code><Systemvar></code> :	Toute variable système de type REAL qui peut être utilisée dans les actions synchrones.
<code><Usure></code> :	Paramètre d'usure (longueur 1, 2 ou 3) dans lequel la valeur de correction est ajoutée .

Paramètres	Signification
<n°_canal> :	Canal de destination dans lequel la correction doit s'appliquer. Ainsi, le dressage simultané à partir d'un canal parallèle est possible. Dans le canal de destination de la correction, la correction en ligne doit être activée avec FTOCON. Si aucun numéro de canal n'est programmé, la correction s'applique dans le canal actif.
<n°_broche> :	Le numéro de broche est programmé lorsqu'une meule non active doit être dressée. Condition : une des fonctions suivantes est active "Vitesse de meule constante" "Surveillance d'outil" Si aucun numéro de broche n'est programmé, l'outil actif est corrigé.

Exemple

Correction de la longueur d'une meule active

Code de programme	Commentaire
FCTDEF(1, -1000, 1000, -\$AA_IW[V], 1)	
; FTOC :	
; N° de polynôme : 1	
; Variable système : \$AA_IW[V] (valeur réelle axiale de l'axe V)	
; Paramètre d'usure : Longueur 3	
; Canal de destination : Canal 1	
ID=1 DO FTOC(1, \$AA_IW[V], 3, 1)	
WAITM(1,1,2)	; Synchronisation avec le canal d'usinage
G1 V-0.05 F0.01 G91	; Déplacement de l'axe V
...	
CANCEL(1)	; Désactiver la correction en ligne
...	

Remarque

Aucune fréquence et aucune condition n'étant indiquées dans l'action synchrone, la partie action est exécutée à chaque période d'appel de l'interpolateur.

4.7.5 Blocage de lecture programmé (RDISABLE)

Fonction

L'instruction `RDISABLE` dans la partie action entraîne l'arrêt du traitement du bloc lorsque la condition associée est remplie. Seules les actions synchrones de déplacement programmées continuent d'être traitées. Si la condition de l'instruction `RDISABLE` n'est plus remplie, le blocage de lecture est annulé.

A la fin du bloc avec `RDISABLE`, un arrêt précis est déclenché, que le blocage de lecture soit actif ou non. L'arrêt précis est également déclenché lorsque la commande se trouve en contournage (G64, G641 ... G645).

`RDISABLE` peut être programmé de manière modale ou non modale (`ID=`, `IDS=`).

Application

`RDISABLE` permet par exemple de démarrer le programme dans la période d'appel de l'interpolateur en fonction d'entrées externes.

Exemple

Code de programme	Commentaire
WHENEVER \$A_INA[2]<7000 DO RDISABLE	; Lorsque la tension est inférieure à 7 V à l'entrée 2, le programme est arrêté (hypothèse : la valeur 1000 cor- respond à 1 V).
...	
N10 G01 X10	; A la fin de N10, RDISABLE est actif si la condition est remplie pendant son traitement.
N20 Y20	
...	

Autres conditions à prendre en compte

Blocage de lecture RDISABLE en liaison avec une permutation d'axe

Si le blocage de lecture `RDISABLE` et la permutation d'axe (par exemple axe à interpolation → axe de positionnement) s'appliquent ensemble dans un bloc via les actions synchrones, `RDISABLE` ne s'applique pas au bloc, mais au bloc de réaccostage `REPOSA` généré implicitement par la permutation d'axe :

Code de programme	Commentaire
N100 G0 G60 X300 Y300	
N105 WHEN TRUE DO POS[X]=20 FA[X]=20000	; Action synchrone → REORG → REPOSA
N110 WHENEVER \$AA_IM[X]<>20 DO RDISABLE	; RDISABLE s'applique à REPOSA
N115 G0 Y20	; 1. Axe X, 2è axe Y
N120 Y-20	
N125 M30	

4.7 Actions synchrones

L'action synchrone du bloc N105 permet de transformer l'axe à interpolation X en axe de positionnement. REORG est donc exécuté avec REPOSA dans le canal. RDISABLE dans N110 ne s'applique donc pas au bloc N115, mais au bloc REPOSA interne. Ainsi, l'axe de positionnement X est d'abord déplacé à la position programmée, suivi par l'axe Y du bloc N115.

Un déblocage explicite de l'axe à interpolation X avant son déplacement en tant qu'axe de positionnement (action synchrone dans N105) avec RELEASE (X) évite le processus de REORG et les axes X et Y se déplacent ensemble dans le bloc N115.

Code de programme	Commentaire
N100 G0 G60 X300 Y300	
N101 RELEASE (X)	; Déblocage explicite
N105 WHEN TRUE DO POS[X]=20 FA[X]=20000	
...	

4.7.6 Annulation de l'arrêt du prétraitement (STOPREOF)

L'instruction STOPREOF permet d'annuler un arrêt existant du prétraitement à partir d'une action synchrone.

Remarque

L'instruction STOPREOF ne doit être programmée que dans des actions synchrones à effet non modal (sans indication d'ID ou IDS) et uniquement avec une indication de fréquence WHEN.

Exemple

- N10 : Action synchrone non modale
Si la distance restant à parcourir \$AC_DTEB est inférieure à 5 mm, l'arrêt du prétraitement défini est annulé en raison de la lecture de l'entrée analogique \$A_INA.
- N20 : Bloc de déplacement dont la distance trajectoire restant à parcourir est évaluée via \$AC_DTEB.
- N30 : Branchement qui se déclenche en raison de la lecture de l'arrêt du prétraitement \$A_INA.

En raison de l'action synchrone, l'entrée \$A_INA n'est pas évaluée à partir de la fin du bloc N20 seulement, mais déjà 5 mm avant la fin du bloc. Si la tension est alors supérieure à 5 V à l'entrée \$A_INA, un saut sera effectué vers "REPERE_1".

Code de programme
N10 WHEN \$AC_DTEB < 5 DO STOPREOF
N20 G01 X100
N30 IF \$A_INA[7] > 5000 GOTOF REPERE_1

4.7.7 Suppression de la distance restant à parcourir (DELDTG)

L'instruction `DELDTG` permet de supprimer la distance trajectoire restant à parcourir dans l'action synchrone et la fonction `DELDTG ( . . . )` les distances axiales restant à parcourir.

Après la suppression des distances restant à parcourir, la valeur de la distance restante supprimée peut être lue via une variable système :

- Distance trajectoire restant à parcourir : `$AC_DELT`
- Distance axiale restant à parcourir : `$AA_DELT`

Syntaxe

```
DELDTG  
DELDTG (<Axe 1>[, <Axe 2>, ...])
```

Signification

Paramètre	Signification
<code>DELDTG</code>	Suppression de la distance trajectoire restant à parcourir
<code>DELDTG (. . .)</code>	Suppression de la distance axiale restant à parcourir des axes de canal indiqués
<code><axe n> :</code>	Axe de canal

Autres conditions à prendre en compte

Suppression des distances trajectoire et axiales restant à parcourir

La suppression des distances trajectoire et axiales restant à parcourir ne peut être exécutée que dans une action synchrone à effet **non modal** (sans indication d'ID ou IDS).

Suppression de la distance trajectoire restant à parcourir

- La suppression de la distance trajectoire restant à parcourir ne peut être exécutée que dans une action synchrone à effet non modal (sans indication d'ID ou IDS).
- La suppression de la distance trajectoire restant à parcourir ne peut **pas** être utilisée si la correction de rayon d'outil est active.

Suppression de la distance axiale restant à parcourir

Suppression de la distance restant à parcourir pour axes indexés :

- **sans** denture Hirth : L'axe est freiné immédiatement
- **avec** denture Hirth : L'axe continue jusqu'à la position indexée suivante

Exemples

Suppression de la distance trajectoire restant à parcourir

Si l'entrée \$A_IN est réglée pendant le bloc de déplacement N20, la distance trajectoire restant à parcourir est supprimée.

Code de programme

```
N10 WHEN $A_IN[1]==1 DO DELDTG
N20 G01 X100 Y100 F1000
```

Suppression de la distance axiale restant à parcourir

N10 : Si l'entrée 1 est réglée à un moment quelconque du programme pièce, l'axe V est démarré comme axe de positionnement dans la direction de déplacement positive.

N100 : Action synchrone à effet non modal pour la suppression de la distance restant à parcourir de l'axe V en fonction de l'entrée TOR 2.

N110 : Action synchrone à effet non modal pour la suppression de la distance restant à parcourir de l'axe X1 en fonction de l'entrée TOR 3.

N120 : L'axe X1 est positionné de manière modale. Déplacement des axes Y et Z en tant qu'axes à interpolation. Les actions synchrones à effet non modal issues de N100 et N110 sont exécutées avec N120. A la fin du bloc N120, les actions synchrones à effet non modal se terminent également.

La distance restant à parcourir des axes X1 et V ne peut ainsi être supprimée que tant que N120 est actif.

Code de programme

```
N10 ID=1 WHEN $A_IN[1]==1 DO MOV[V]=1 FA[V]=700
...
N100 WHEN $A_IN[2]==1 DO DELDTG(V)
N110 WHEN $A_IN[3]==1 DO DELDTG(X1)
N120 POSA[X1]=100 FA[X1]=10 G1 Y100 Z100 F1000
```

4.7.8 Déplacement d'axes en position (POS)

L'instruction POS permet de déplacer un axe avec une action synchrone. L'axe est alors appelé "axe de commande". Un déplacement alterné de l'axe avec le programme pièce et une action synchrone est possible. Si un axe de commande déplacé avec des actions synchrones est ensuite déplacé avec le programme pièce, un arrêt du prétraitement des blocs avec réorganisation (STOPRE) est exécuté dans le canal du programme pièce.

Exemple 1 : Déplacement alterné avec le programme pièce et une action synchrone

Code de programme**Commentaire**

```
N10 G01 X100 Y200 F1000 ; Déplacement avec le programme pièce.
...
; Déplacement avec une action synchrone statique lorsque l'entrée 1 est mise à 1 :
N20 ID=1 WHEN $A_IN[1]==1 DO POS[X]=150 FA[X]=200
```

Code de programme	Commentaire
...	
CANCEL(1)	; Désactiver l'action synchrone.
...	
; Nouveau déplacement avec le programme pièce => arrêt implicite du prétraitement des blocs	
; avec réorganisation si l'axe X a été déplacé entre-temps	
; avec l'action synchrone.	
N100 G01 X240 Y200 F1000	

Exemple 2 : Déplacement alterné de l'axe X avec deux actions synchrones

Si le déplacement d'une action synchrone est encore actif lorsque le déplacement de l'autre action synchrone est lancé, le deuxième déplacement prend le relais du premier.

Code de programme

```
; 1er déplacement :
ID=1 EVERY $A_IN[1]>=1 DO POS[V]=100 FA[V]=560
; 2e déplacement :
ID=2 EVERY $A_IN[2]>=1 DO POS[V]=$AA_IM[V] FA[V]=790
```

Cotes : absolues / relatives

Les instructions G90 / G91 pour la définition des cotes (absolues / relatives) ne peuvent pas être programmées dans les actions synchrones. Par conséquent, dans une action synchrone, les cotes qui étaient effectives dans le programme pièce au moment de l'exécution de l'action synchrone s'appliquent par défaut.

Pour définir les cotes dans une action synchrone, les instructions suivantes peuvent être programmées dans la partie action :

Instruction	Signification
IC (...)	Relatif
AC (...)	Absolu
DC (...)	Direct (positionner l'axe rotatif par le chemin le plus court)
ACN (...)	Positionnement absolu de l'axe rotatif à valeur modulo dans le sens de déplacement négatif
ACP (...)	Programmation absolue de l'axe rotatif à valeur modulo dans le sens de déplacement positif
CAC (...)	Déplacement de l'axe en position codée, absolu
CIC (...)	Déplacement de l'axe en position codée, relatif
CDC (...)	Déplacement de l'axe rotatif en position codée par le chemin le plus court
CACN (...)	Déplacement de l'axe rotatif à valeur modulo en position codée dans le sens négatif
CACP (...)	Déplacement de l'axe rotatif à valeur modulo en position codée dans le sens positif

Exemples :

Code de programme

```
; Déplacement relatif de 10 mm :
ID=1 EVERY G710 $AA_IM[B]>75 DO POS[X]=IC(10)
```

4.7 Actions synchrones

Code de programme

```
...
; Déplacement absolu :
ID=1 EVERY G710 $AA_IM[B]>75 DO POS[X]=AC($AA_MW[V]-$AA_IM[W]+13.5)
```

Comportement en présence de frames actifs spécifiques à l'axe

La prise en compte ou non dans les actions synchrones des frames spécifiques à l'axe et des corrections de longueur d'outil programmables et réglables dépend du réglage PM suivant :

MD32074 \$MA_FRAME_OR_CORRPOS_NOTALLOWED

Bit	Valeur	Signification
9	0 (par défaut)	Le frame spécifique à l'axe et/ou la correction de longueur d'outil actif dans le programme pièce au moment de l'exécution est effectif dans l'action synchrone exécutée parallèlement au programme pièce.
	1	Les frames spécifiques à l'axe et la correction de longueur d'outil ne sont pas pris en compte pour les axes de commande.

Exemple 1 : Déplacement avec frames / corrections de longueur d'outil **actifs** (bit 9 == 0) :

Code de programme	Commentaire
N100 TRANS X20	; Décalage d'origine en X : 20 mm.
; Action synchrone : l'axe X se déplace sur la position 60 mm.	
IDS=1 EVERY G710 \$A_IN==1 DO POS[X]=40	
...	
; Décalage d'origine en X : -10 mm =>	
; Action synchrone : l'axe X se déplace maintenant sur la position 30 mm	
N130 TRANS X-10	
...	

Exemple 2 : Déplacement avec frames / corrections de longueur d'outil **désactivés** (bit 9 == 1) :

Code de programme	Commentaire
N100 TRANS X=0,001	; Décalage d'origine en X : 0,001 degré
N120 POS[X]=270	; X se déplace sur la position 270,001 degrés
...	
; Avec \$A_IN=1, X se déplace sur la position 180,000 degrés	
IDS=1 EVERY G710 \$A_IN==1 DO POS[X]=180	
...	
; X se déplace sur la position 90,001 degrés	
N130 POS[X]=90	
...	
; Position codée 1 = 100 degrés => X se déplace sur 100,001 degrés	
N140 POS[X]=CAC(1)	
...	
; Position codée 2 = 200 degrés => X se déplace sur 200,000 degrés	
N150 POS[X]=CIC(1)	

Remarque

Si un axe de commande est déplacé de manière relative sur des positions d'indexage, les frames spécifiques à l'axe sont **sans effet** pour cet axe de commande.

Inhibition bloc par bloc du frame actif avec G153

Si PM32074 est réglé de manière à ce que les frames spécifiques à l'axe et corrections d'outil actifs pour les axes de commande soient toujours pris en compte (bit 9 == 0), le frame actif peut être inhibé si nécessaire dans une action synchrone non modale avec G153. Pour cela, le bloc avec l'instruction G153 doit être rendu exécutable par le code de programme G4 F0.1. Il convient de noter que l'arrêt temporisé est alors d'au moins 0,1. Cela permet de garantir un temps de traitement suffisant dans l'interpolateur.

Exemples :

```
...  
WHEN TRUE DO POS[Z]=401 FA[Z]=$MA_MAX_AX_VELO[Z]  
G153 G4 F0.1  
...
```

```
...  
WHILE $AA_IM[Z]<400  
ENDWHILE  
WHEN TRUE DO POS[Z]=400  
G153 G4 F0.1  
...
```

Prise de contrôle d'un axe de commande par l'AP

Un axe de commande qui a été démarré avec une action synchrone statique (IDS) devient indépendant de l'état du programme pièce qui contient l'action synchrone lorsque le contrôle de l'axe de commande est pris par l'AP :

<Axis>.plcCtrl.out.requestPlcCtrl (demander la commande par l'AP pour les opérations à axe unique autonomes)

Bibliographie

Description fonctionnelle Axes et broches

État de l'axe paramétrable

Le paramètre machine suivant permet de paramétrer le comportement relatif à l'état de l'axe après la fin du programme pièce et un Reset de la CN :

PM30450 \$MA_IS_CONCURRENT_POS_AX[<axe>] = <valeur>

<Valeur>	État de l'axe avant fin PP / RESET CN ¹⁾	État de l'axe après fin PP / RESET CN ¹⁾
0	Axe de canal	Axe de canal
0	Axe de commande	Axe de canal
1	Axe de canal	Axe de commande
1	Axe de commande	Axe de commande

¹⁾ Fin PP : fin du programme pièce

Signaux AP

Basic Program Plus	Basic Program	
<Axis>.plcCtrl.out.requestPlcCtrl	LBP_Axis*.A_PLCAxis	DB31,DBX28.7

Voir aussi

Cycles technologiques (Page 125)

4.7.9 Réglage du système de mesure (G70, G71, G700, G710)

Si un système de mesure (anglo-saxon / métrique) précis n'est pas spécifié explicitement avec G70, G71, G700 ou G710 dans une action synchrone, le système de mesure actif dans le programme pièce au moment de l'exécution de l'action synchrone est effectif :

- G70/G71 actif dans le programme pièce :
 - Toutes les valeurs de position **programmées** sont interprétées dans le système de mesure **programmé**.
 - Toutes les valeurs de position **lues** sont interprétées dans le **système de base paramétré**.
- G700/G710 actif dans le programme pièce :
 - Toutes les valeurs de position **programmées** sont interprétées dans le système de mesure **programmé**.
 - Toutes les valeurs de position **lues** sont interprétées dans le **système de base paramétré**.

Lors de la définition du système de mesure dans l'action synchrone, les règles suivantes s'appliquent :

- Si un système de mesure est programmé dans la partie condition, celui-ci est également effectif dans la partie action si aucun système de mesure propre n'est programmé dans celle-ci.
- Si un système de mesure est programmé uniquement dans la partie action, le système de mesure actuellement actif dans le programme pièce est effectif dans la partie condition.

- Des systèmes de mesure différents peuvent être programmés dans la partie condition et dans la partie action.
- Le système de mesure programmé dans l'action synchrone n'a aucune influence sur le programme pièce.

Exemple

Code de programme	Commentaire		
N10 ID=1 EVERY \$AA_IM[Z]>200 DO POS[Z2]=10	;	\$AA_IM :	#
	;	200 :	#
	;	10 :	#
N20 ID=2 EVERY \$AA_IM[Z]>200 DO G70 POS[Z2]=10	;	\$AA_IM :	#
	;	200 :	#
	;	10 :	inch
N30 ID=3 EVERY G71 \$AA_IM[Z]>200 DO POS[Z2]=10	;	\$AA_IM :	#
	;	200 :	mm
	;	10 :	mm
N40 ID=4 EVERY G71 \$AA_IM[Z]>200 DO G70 POS[Z2]=10	;	\$AA_IM :	#
	;	200 :	mm
	;	10 :	inch
N50 ID=5 EVERY \$AA_IM[Z]>200 DO G700 POS[Z2]=10	;	\$AA_IM :	#
	;	200 :	#
	;	10 :	inch
N60 ID=6 EVERY G710 \$AA_IM[Z]>200 DO POS[Z2]=10	;	\$AA_IM :	mm
	;	200 :	mm
	;	10 :	mm
N70 ID=7 EVERY G710 \$AA_IM[Z]>200 DO G700 POS[Z2]=10	;	\$AA_IM :	mm
	;	200 :	mm
	;	10 :	inch
#: l'unité dépend du système de base paramétré (PM10240 \$MN_SCALING_SYSTEM_IS_METRIC) et du système de mesure programmé dans le programme pièce			

Remarque

Système de mesure et cycles technologiques

Au lieu de spécifier le système de mesure dans la partie action de l'action synchrone, le système de mesure peut également être programmé dans le cycle technologique lorsque celui-ci est utilisé.

4.7.10 Position dans la zone de référence prescrite (POSRANGE)

Fonction

La fonction POSRANGE permet de déterminer si la position actuelle d'un axe se trouve dans une plage de tolérance autour d'une position de référence prescrite.

Remarque

La correction modulo est prise en compte pour les axes modulo.

Syntaxe

<Etat> POSRANGE (<Axe>, <PosRéf>, <Tolérance>, [<SysCoord>])

Signification

<Etat>	Valeur de retour de la fonction Type : BOOL TRUE : La position actuelle de l'axe se trouve dans la plage de tolérance. FALSE : La position actuelle de l'axe ne se trouve pas dans la plage de tolérance.
<Axe>	Nom de l'axe de canal Type : AXIS
<PosRéf>	Position de référence Type : REAL
<Tolérance>	Tolérance autorisée autour de la position de référence Type : REAL La tolérance est indiquée en tant que valeur. La plage de tolérance résulte de : Position de référence +/- tolérance
<SysCoord>	Facultatif : Système de coordonnées Type : INT Plage de valeurs : 0 = SCM (système de coordonnées machine) 1 = SCB (système de coordonnées de base) 2 = SPR (système de coordonnées-pièce réglable) 3 = SCP (système de coordonnées pièce)

4.7.11 Déplacement d'axes, sans fin (MOV)

Fonction

L'instruction `MOV` permet de déplacer sans fin un axe, c'est-à-dire sans indiquer de position de fin, dans une direction donnée. L'axe se déplace dans la direction prescrite jusqu'à ce qu'il soit arrêté ou qu'une autre direction soit prescrite par `MOV`.

Exemple d'application : axe rotatif tournant en continu

Syntaxe

`MOV[<Axe>] = <Direction>`

Signification

<code>MOV</code>	Instruction de déplacement pour un axe de commande
<code><Axe></code>	Nom d'axe de canal Type : AXIS
<code><Direction></code>	Sens de déplacement Type : INT Plage de valeurs : > 0: direction de déplacement positive (par défaut : +1) < 0: direction de déplacement négative (par défaut : -1) = 0: Pause

Remarque

Axe indexé

Si un axe indexé est arrêté avec `MOV[<axe indexé>] = 0`, il s'arrête à la position indexée suivante.

Cycle technologique

L'instruction `MOV` ne doit **pas** être utilisée dans des cycles technologiques.

Voir aussi

Avance axiale (FA) (Page 93)

4.7.12 Avance axiale (FA)

L'instruction `FA` permet de spécifier une avance axiale dans une action synchrone. L'avance axiale est à effet modal.

Exemples

Valeur d'avance constante :

Code de programme

```
ID=1 EVERY $AA_IM[B] > 75 DO POS[U]=100 FA[U]=990
```

Valeur d'avance variable :

Code de programme

```
ID=1 EVERY $AA_IM[B] > 75 DO POS[U]=100 FA[U]=$AA_VACTM[W]+100
IDS=2 WHENEVER $A_IN[1] == 1 DO POS[X]=100 FA[X]=$R1
```

Remarques

- La valeur standard pour l'avance des axes de positionnement se règle avec le paramètre machine axial :
MD32060 \$MA_POS_AX_VELO (position initiale pour vitesse d'axe de positionnement)
- L'avance axiale peut être spécifiée en tant qu'avance linéaire ou par tour.
Le type d'avance est réglable avec la donnée de réglage :
SD43300 \$SA_ASSIGN_FEED_PER_REV_SOURCE (avance par tour pour axes de positionnement / broches)
- Le type d'avance peut être basculé de manière synchrone sur le programme pièce avec les instructions FPRAON et FPRAOF. Voir :

Informations complémentaires

Description fonctionnelle Axes et broches

Remarque

Pour que des cycles technologiques s'exécutant en parallèle ne se bloquent pas mutuellement, l'avance axiale à partir des actions synchrones n'est pas transmise en tant que fonction auxiliaire à l'interface CN/AP.

Voir aussi

Déplacement d'axes, sans fin (MOV) (Page 93)

4.7.13 Permutation d'axe (GET, RELEASE, AXTOCHAN)

Les axes de commande peuvent être permutés entre les canaux avec les instructions GET, RELEASE et AXTOCHAN.

Condition

L'axe de commande qui est permuté entre les canaux doit être connu ou paramétré en tant qu'axe de canal dans le canal correspondant.

Programmation

Syntaxe

```
GET(<axe 1> [{, <axe n>}])  
RELEASE((<axe 1> [{, <axe n> }]))  
AXTOCHAN(<axe 1>, <numéro de canal 1> [{, <axe n>, <numéro de canal  
n> }])
```

Signification

GET :	Demande de permutation d'un axe dans le canal propre
RELEASE :	Déblocage d'un axe pour la permutation
AXTOCHAN :	Demande d'un axe pour la permutation dans le canal indiqué
<axe n> :	Nom d'axe machine
	Type : AXIS
	Plage de valeurs : noms d'axe machine définis dans le canal
<numéro de canal n> :	Numéro de canal
	Type : INTEGER
	Plage de valeurs : 1 ... numéro de canal maximal

Type d'axe et état de l'axe concernant la permutation

Le type d'axe et l'état de l'axe actuellement valides au moment de l'activation de l'action synchrone peuvent être interrogés avec les variables système \$AA_AXCHANGE_TYP et \$AA_AXCHANGE_STAT. En fonction du canal qui dispose du droit d'interpolation de cet axe, et de l'état pour la permutation d'axe autorisée, le déroulement à partir de l'action synchrone est différent.

À partir d'une action synchrone, un axe peut être demandé avec GET dans les conditions suivantes :

- Un autre canal dispose du droit d'écriture ou du droit d'interpolation pour l'axe.
- L'axe demandé est déjà affecté au canal demandé.
- L'axe à l'état d'axe banalisé est contrôlé par l'AP.
- L'axe est un axe de commande, un axe d'oscillation ou un axe AP concurrent.
- L'axe est déjà affecté au programme CN du canal.

Remarque

Autre condition à prendre en compte : un "axe exclusivement contrôlé par l'AP" ou un "axe AP affecté de manière fixe" ne peut pas être affecté au programme CN.

À partir d'une action synchrone, un axe peut être débloqué avec RELEASE dans les conditions suivantes :

- L'axe est jusqu'à présent affecté au programme CN du canal.
- L'axe se trouve déjà à l'état d'axe banalisé.
- Un autre canal dispose déjà du droit d'interpolation de cet axe.

Demander un axe à partir d'un autre canal

Si, au moment de l'activation de l'action `GET`, un **autre canal** dispose du droit d'interpolation pour l'axe `$AA_AXCHANGE_TYP[axe] == 2`, l'axe est alors demandé par ce canal par le biais d'une permutation d'axe `$AA_AXCHANGE_TYP[axe] == 6` et affecté dès que possible au canal qui a fait la demande. Son état devient alors celui d'un **axe banalisé** `$AA_AXCHANGE_TYP[axe] == 3`.

Le passage à l'état d'axe banalisé n'entraîne **aucune** réorganisation dans le canal qui fait la demande.

L'axe demandé a déjà été demandé en tant qu'axe banalisé :

`$AA_AXCHANGE_TYP[<axe>] == 6`, l'axe est alors demandé pour le programme CN
`$AA_AXCHANGE_TYP[axe] == 5` et affecté dès que possible au programme CN du canal
`$AA_AXCHANGE_TYP[axe] == 0`.

Remarque

Cette affectation **entraîne** une réorganisation.

L'axe est déjà affecté au canal demandé

Si, au moment de l'activation, l'axe demandé est **déjà affecté à ce canal** et qu'il se trouve à l'état d'axe banalisé non contrôlé par l'AP `$AA_AXCHANGE_TYP[axe] == 3`, cet axe est alors affecté au programme CN `$AA_AXCHANGE_TYP[axe] == 0`.

Cela entraîne **une** opération de réorganisation.

Axe à l'état d'axe banalisé contrôlé par l'AP

Si l'axe à l'état d'axe banalisé est **contrôlé par l'AP** `$AA_AXCHANGE_TYP[axe] == 4`, il est demandé en tant qu'axe banalisé `$AA_AXCHANGE_TYP[axe] == 8`. L'axe est alors bloqué pour une permutation d'axe automatique entre les canaux (bit 0 == 0), en fonction du bit 0 dans le paramètre machine :

MD10722 `$MN_AXCHANGE_MASK` (paramétrage du comportement de permutation d'axe)

Cela correspond à `$AA_AXCHANGE_STAT[axe] == 1`.

L'axe est actif en tant qu'axe de commande / affecté à l'AP

Si l'axe est actif en tant qu'axe de commande ou axe d'oscillation, ou bien en tant qu'axe de positionnement concurrent (axe AP) (`$AA_AXCHANGE_TYP[<axe>] == 1`), il est demandé en tant qu'axe banalisé (`$AA_AXCHANGE_TYP[<axe>] == 8`). L'axe est alors bloqué pour une permutation d'axe automatique entre les canaux, en fonction du réglage dans le paramètre machine suivant :

MD10722 `$MN_AXCHANGE_MASK` (paramétrage du comportement de permutation d'axe)

Cela correspond à `$AA_AXCHANGE_STAT[<axe>] == 1`.

Avec une nouvelle demande `GET`, l'axe est ensuite demandé pour le programme CN $\Rightarrow$ `$AA_AXCHANGE_TYP[axe] == 7`.

Axe déjà affecté au programme CN du canal

Si l'axe est déjà affecté au programme CN du canal ($\$AA_AXCHANGE_TYP[<axe>] == 0$) ou si cette affectation est demandée, p. ex. permutation d'axe déclenchée par le programme CN ($\$AA_AXCHANGE_TYP[<axe>] == 5$ ou $\$AA_AXCHANGE_TYP[<axe>] == 7$), **aucun** changement d'état ne se produit.

Débloquer l'axe pour la permutation d'axe

Si, au moment du déblocage, l'axe est affecté au programme CN ($\$AA_AXCHANGE_TYP[<axe>] == 0$), il est mis à l'état d'axe banalisé ($\$AA_AXCHANGE_TYP[<axe>] == 3$) et débloquent le cas échéant pour la permutation d'axe dans un autre canal.

Cela entraîne **une** opération de réorganisation.

L'axe à débloquent est déjà un axe banalisé :

Si l'axe est déjà à l'état d'axe banalisé ($\$AA_AXCHANGE_TYP[<axe>] == 3$) ou s'il est actif en tant qu'axe de commande ou axe d'oscillation ou bien affecté à l'AP en tant qu'axe de positionnement concurrent ($\$AA_AXCHANGE_TYP[<axe>] == 1$), il est débloquent pour une permutation d'axe automatique entre les canaux.

S'il n'y a pas d'autre raison d'associer l'axe au canal, $\$AA_AXCHANGE_STAT[<axe>]$ est réinitialisé de 1 à 0. Une telle association de l'axe est par exemple présente dans les cas suivants :

- Couplage d'axes actif
- Retrait rapide actif
- Transformation active
- Demande JOG
- Frame en rotation avec déplacement d'axe AP, de commande ou d'oscillation

Un autre canal dispose déjà du droit d'interpolation

Si un autre canal dispose déjà du droit d'interpolation ($\$AA_AXCHANGE_TYP[<axe>] == 2$), aucun changement d'état ne se produit. Cela signifie également que l'attente d'un axe, déclenchée par le programme CN ($\$AA_AXCHANGE_TYP[<axe>] == 5$), ou une demande `GET` précédente à partir d'une action synchrone ($\$AA_AXCHANGE_TYP[<axe>] == 6$) ne peut pas être interrompue par un débloquent `RELEASE` à partir d'une action synchrone.

Autres conditions à prendre en compte

- Si plusieurs demandes `GET` et débloques `RELEASE` sont programmés pour le même axe dans la partie action d'une action synchrone ou dans un bloc d'un cycle technologique, ces ordres s'annulent mutuellement dans certains cas et seul le dernier ordre est exécuté.
Exemple :

Programmation : `GET (X, Y)   RELEASE (Y, Z)   GET (Z)`

Exécution : `GET(X) RELEASE(Y) GET(Z)`

4.7 Actions synchrones

- Si, dans la partie action d'une action synchrone, d'autres instructions sont programmées en plus de GET / RELEASE, le système n'attendra pas la fin de la demande GET / RELEASE avant d'exécuter ces instructions. Cela peut entraîner une erreur si, par exemple, un axe demandé avec GET n'est pas encore disponible pour le mouvement de positionnement :
GET[<axe>] POS[<axe>]

Exemple 1 : GET et RELEASE en tant qu'action dans des actions synchrones dans deux canaux

Prérequis : l'axe Z doit être l'axe de canal dans le 1er et le 2ème canal

1. Déroulement du programme dans le 1er canal :

Code de programme	Commentaire
WHEN TRUE DO RELEASE(Z)	; L'axe Z est banalisé
; Blocage de la lecture tant que l'axe Z est un axe du programme	
WHENEVER \$AA_TYP[Z] == 1 DO RDISABLE	
N110 G4 F0.1	
...	
; L'axe Z redevient un axe du programme CN	
WHEN TRUE DO GET(Z)	
; Blocage de la lecture jusqu'à ce que l'axe Z soit un axe du programme	
WHENEVER \$AA_TYP[Z] <> 1 DO RDISABLE	
N120 G4 F0.1	
...	
WHEN TRUE DO RELEASE(Z)	; L'axe Z est banalisé
; Blocage de la lecture tant que l'axe Z est un axe du programme	
WHENEVER \$AA_TYP[Z] == 1 DO RDISABLE	
N130 G4 F0.1	
...	
N140 START(2)	; démarrer 2ème canal
N150 ; voir ci-dessous : "3. Suite : déroulement du programme dans le 1er canal"	

2. Déroulement du programme dans le 2ème canal :

Code de programme	Commentaire
WHEN TRUE DO GET(Z)	; Faire passer l'axe Z dans le 2ème canal (banalisé)
; Blocage de la lecture tant que l'axe Z est dans un autre canal	
WHENEVER \$AA_TYP[Z] == 0 DO RDISABLE	
N210 G4 F0.1	
...	
WHEN TRUE DO GET(Z)	; L'axe Z devient un axe du programme CN
; Blocage de la lecture jusqu'à ce que l'axe Z soit un axe du programme	
WHENEVER \$AA_TYP[Z] <> 1 DO RDISABLE	
N220 G4 F0.1	
...	
WHEN TRUE DO RELEASE(Z)	; Axe Z banalisé dans le 2ème canal

Code de programme	Commentaire
; Blocage de la lecture tant que l'axe Z est un axe du programme	
WHENEVER \$AA_TYP[Z] == 1 DO RDISABLE	
N230 G4 F0.1	
...	
N250 WAITM(10,1,2)	; Synchroniser avec le canal 1
N999 M30	

3. Suite : déroulement du programme dans le 1er canal :

Code de programme	Commentaire
N150 WAITM(10,1,2)	; Synchroniser avec le canal 2
...	
WHEN TRUE DO GET(Z)	; Faire passer l'axe Z dans ce canal
; Blocage de la lecture tant que l'axe Z est dans un autre canal	
WHENEVER \$AA_TYP[Z] == 0 DO RDISABLE	
N160 G4 F0.1	
...	
N199 WAITE(2)	; En attente de la fin du programme dans le canal 2
N999 M30	

Transmettre un axe à un autre canal (AXTOCHAN)

À partir d'une action synchrone, un axe peut être demandé pour un canal quelconque avec l'instruction AXTOCHAN.

Si l'axe est déjà affecté au programme CN du canal (\$AA_AXCHANGE_TYP[<axe>] == 0), **aucun** changement d'état ne se produit.

Lorsqu'un axe est demandé pour le canal propre à partir d'une action synchrone, AXTOCHAN est reproduit dans l'instruction GET.

- À la **première** demande pour le canal propre, l'axe devient un axe banalisé.
- À la **deuxième** demande, l'axe est affecté au programme CN.

Conditions complémentaires

Un "axe contrôlé par l'AP" correspond à un "axe de positionnement concurrent" pour lequel d'autres conditions particulières doivent être prises en compte. Voir à ce sujet :

Informations complémentaires

Description fonctionnelle Axes et broches

Remarque

Un axe AP ne peut pas changer de canal.

Un axe exclusivement contrôlé par l'AP ne peut pas être affecté au programme CN.

4.7.14 Déplacement de broches (M, S, SPOS)

Les broches peuvent être démarrées, positionnées et arrêtées avec des actions synchrones. La programmation se déroule dans la partie action de l'action synchrone avec la même syntaxe que dans le programme pièce. Sans extension numérique, les instructions sont valables pour la broche maître. Chaque broche peut être programmée en saisissant une extension numérique :

Code de programme	Commentaire
ID = 1 EVERY \$A_IN[1]==1 DO M3 S1000	; Broche maître
ID = 2 EVERY \$A_IN[2]==1 DO SPOS=270	; Broche maître
ID = 1 EVERY \$A_IN[1]==1 DO M1=3 S1=1000 SPOS[2]=90	

Si des instructions concurrentes sont données pour une broche par des actions synchrones activées parallèlement, l'ordre chronologique sera pris en compte pour l'activation.

Déblocage de la broche spécifique à l'utilisateur

Le démarrage des déplacements de broche à des moments définis peut être obtenu à l'aide d'actions synchrones en bloquant le déplacement programmé dans le programme pièce.

Exemple :

La broche est programmée dans un programme pièce et ne doit pas démarrer au début du bloc, mais seulement lorsque l'entrée 1 est activée. La correction de vitesse de rotation de broche est maintenue à 0 % à l'aide d'une action synchrone jusqu'au déblocage par l'entrée 1. Voir à ce propos le chapitre "Correction (\$A...OVR) (Page 41)".

Code de programme
; Tant que l'entrée 1 n'est pas active => correction de vitesse de rotation de broche = 0 %
ID=1 WHENEVER \$A_IN[1]==0 DO \$AA_OVR[S1]=0
...
; Le démarrage de la broche est déclenché
; Le déblocage a lieu lorsque l'entrée 1 est activée
G01 X100 F1000 M3 S1=1000

Transition entre axe de commande et broche

Comme plusieurs actions synchrones peuvent être actives simultanément, il est possible de démarrer un déplacement de broche alors que la broche est déjà active. Dans ce cas, le dernier déplacement activé est réalisé. Si le sens de déplacement est inversé, la broche est d'abord freinée, puis se déplace dans le sens opposé.

Le sens de rotation, la vitesse et la position peuvent également être modifiés pendant le déplacement.

Exemples

Code de programme	Commentaire
ID=1 EVERY \$AC_TIMER[1] >= 5 DO M3 S300	; Sens de rotation et vitesse
ID=2 EVERY \$AC_TIMER[1] >= 7 DO M4 S500	; Sens de rotation et vitesse
ID=3 EVERY \$A_IN[1]==1 DO S1000	; Vitesse

Code de programme	Commentaire
ID=4 EVERY (\$A_IN[4]==1) AND (\$A_IN[1]==0) DO SPOS=0	; Positionnement de la broche

Transitions entre axe et broche

À l'état ↓	Après →	POS	MOV<>0	MOV=0	SPOS	M3/M4	M5	LEADON	TRAIL-ON
pendant le déplacement									
Axe		x	x	x	x	x	x	x	x
Broche en asservissement de position		x	x	x	x	x	x	-	-
Broche en asservissement de vitesse		-	-	-	x	x	x	-	-
en mouvement									
Axe		x	x	x	-	-	-	x	x
Broche en asservissement de position		-	-	-	-	-	-	-	-
Broche en asservissement de vitesse		-	-	-	x	x	x	-	-
Les transitions repérées par x sont autorisées. Les transitions repérées par - sont rejetées avec alarme.									

4.7.15 Forçage de valeurs réelles avec perte de l'état de référencement (PRESETON)

Fonction

À partir d'actions synchrones, la procédure **PRESETON** () définit de nouvelles valeurs réelles pour **un** axe dans le système de coordonnées machine (SCM). Cela correspond à un décalage d'origine du SCM de l'axe. Ainsi, l'axe ne se déplace pas.

À partir d'actions synchrones, **PRESETON** peut être mis en œuvre **uniquement sur des axes de commande**, c'est-à-dire des axes qui ont été démarrés à partir d'une action synchrone. L'axe doit ensuite être affecté au canal qui doit disposer du droit d'interpolation pour cet axe. L'axe n'est **pas** demandé par permutation d'axe par un autre canal.

État de référencement

La définition d'une nouvelle valeur réelle dans le système de coordonnées machine remet à zéro l'état de référencement de l'axe machine :

`<Axis>.basic.in.enc1Synchronized / <Axis>.basic.in.enc2Synchronized = 0`
(référéncé / synchronisé système de mesure 1 / 2)

Il est recommandé de n'utiliser **PRESETON** qu'avec des axes sans point de référence obligatoire.

Pour rétablir le système de coordonnées machine d'origine, le système de mesure de l'axe machine doit être de nouveau référencé, par exemple par accostage du point de référence depuis le programme pièce (G74).

 PRUDENCE
Perte de l'état de référencement La définition d'une nouvelle valeur réelle dans le système de coordonnées machine avec PRESETON remet l'état de référencement de l'axe machine sur "non référencé / synchronisé".

Programmation

Syntaxe

WHEN | EVERY ... DO PRESETON(<axe>, <valeur>)

Signification

WHEN, EVERY : Pour la fréquence (Page 21), seuls WHEN et EVERY peuvent être utilisés.

PRESETON : Forçage de valeur réelle avec perte de l'état de référencement

<axe> : Nom de l'axe machine
Type : AXIS
Plage de valeurs : noms d'axe machine définis dans le canal

<valeur> : Nouvelle valeur réelle de l'axe machine dans le système de coordonnées machine (SCM)
La saisie s'effectue dans le système de mesure actuellement valide (anglo-saxon/métrique)
Une programmation active de diamètre (DIAMON) est prise en compte
Type : REAL

Variable système

\$AC_PRESET

La variable système spécifique à l'axe \$AC_PRESET fournit le vecteur du point d'origine du SCM décalé actuel au point d'origine du SCM₀ d'origine après référencement de l'axe machine.

\$AC_PRESET<axe> = \$AC_PRESET<axe> + "Position réelle actuelle de l'axe dans le SCM" - "PRESETON-position réelle"

Le décalage d'origine peut être à nouveau annulé à l'aide des variables système :

PRESETON(<axe>, \$VA_IM + \$AC_PRESET[<axe>]) ; "Valeur réelle actuelle de l'axe dans le SCM" + "Décalages"

Exemple

Code de programme

```
N10 G1 X=10 F5000
; Déplacement de l'axe X comme axe de commande en position 200
N20 WHEN TRUE DO G71 POS[X]=200
; IF position de consigne de l'axe X dans le SCM ($AA_IM[X]) >= 80
; THEN "Position réelle de l'axe X dans le SCM" = "Position de consigne de
l'axe X dans le SCM" + "Décalage"
;                                     = 80 + 70 = 150
; "Progr. position finale de l'axe X" = "Progr. position finale de
l'axe X" + "Décalage"
;                                     = 200 + 70 = 270
; $AC_PRESET = $AC_PRESET - 70
N30 WHEN G71 $AA_IM[X] >= 80 DO PRESETON(X, $AA_IM[X]+70)
N40 G4 F3
```

Autres conditions à prendre en compte

Axes pour lesquels PRESETON ne peut pas être mis en œuvre

- Axes de commande se déplaçant en mode broche
- Axes de positionnement concurrents se déplaçant (SINU_TraversePosAxis (Basic Program FC18))
- Axes qui participent à une transformation
- Axes à interpolation se déplaçant
- Axes d'oscillation
- Axes pour lesquels une ou plusieurs des fonctions de sécurité (Safety Integrated) suivantes sont actives :
 - Déblocage "Fins de course logiciels sûrs"
PM36901 \$MA_SAFE_FUNCTION_ENABLE[<SafeAxis>], bit 1 = 1
 - Déblocage "Came logicielle sûre", paire 1 ... 4, came +/-
PM36901 \$MA_SAFE_FUNCTION_ENABLE[<SafeAxis>], bit 8 ... 15 = 1
ou
Déblocage "Piste de cames sûre", came 1 ... 30
PM36903 \$MA_SAFE_CAM_ENABLE[<SafeAxis>], bit 0 ... 29 = 1
- Axes Hirth
- Axes synchrones d'un groupe Gantry
- Axes pour lesquels l'accostage du point de référence du programme pièce (G74) est actif
- Axe subordonné d'un couplage vitesse/couple de rotation (couplage principal/subordonné)

Axes géométriques

- PRESETON peut être mis en œuvre sur un axe géométrique à l'arrêt si aucun autre axe géométrique ne se déplace dans le canal en même temps.
- PRESETON peut être mis en œuvre sur un axe géométrique à l'arrêt même si un autre axe géométrique se déplace dans le canal en même temps, si cet axe se trouve à l'état "Axe neutre" ou se déplace comme axe de commande.

Exemple : Un autre axe géométrique (X) se déplace en même temps à l'état "**Axe neutre**"

Code de programme	Commentaire
N10 G0 X0 Y0	; X, Y : axes géométriques
N15 RELEASE(Y) ¹⁾	; Axe neutre
N20 ID=1 WHEN 20.0 < \$AA_IM[X] DO PRESETON(Y,20)	; \$AA_IM : position de consigne dans le SCM
N30 G0 X40	; L'axe géométrique X se déplace
N40 M30	
1) Remarque Si l'axe est débloquenté dans la partie action de l'action synchrone, il n'y a aucune garantie que le déblocage aura lieu à temps. N20 ID=1 WHEN 20.0 < \$AA_IM[X] DO RELEASE(Y) PRESETON(Y,20) ; NON recommandé !	

Exemple : Un autre axe géométrique (X) se déplace en même temps que l'**axe de commande**

Code de programme	Commentaire
N10 G0 X0 Y0	; X, Y : axes géométriques
N20 ID=1 WHEN TRUE DO POS[X]=40 FA[X]=1000	; Axe de commande X
N30 ID=2 WHEN 20.0 < \$AA_IM[X] DO PRESETON(Y,20)	; \$AA_IM : position de consigne dans le SCM
N40 M30	

Axes contrôlés par l'AP

- PRESETON peut être mis en œuvre sur des axes contrôlés par l'AP en fonction de leur type actuel.

États de la broche

Le tableau suivant indique les réactions qui se déclenchent quand PRESETON est mis en œuvre dans une action synchrone sur une broche :

PRESETON dans une action synchrone			
Mode de fonctionnement de la broche	État de déplacement	Affecté au programme CN	Axe d'exécution
Mode de régulation de vitesse	En déplacement	Alarme 17601	Alarme 17601
	Arrêté	+	+
Mode Positionnement SPOS	Déplacement	Alarme 17601	Alarme 17601
	Arrêté	+	+
Positionnement au-delà des limites de bloc SPOSA	En déplacement	Alarme 17601	-

PRESETON dans une action synchrone			
Mode de fonctionnement de la broche	État de déplacement	Affecté au programme CN	Axe d'exécution
Mode axe	En déplacement	Alarme 17601	+
	Arrêté	+	+
+: possible -: impossible			

PRESETON dans le programme CN			
Mode de fonctionnement de la broche	État de déplacement	Affecté au programme CN	Axe d'exécution
Mode de régulation de vitesse	En déplacement	Alarme 22324	Alarme 22324
	Arrêté	+	+
Mode Positionnement SPOS	Déplacement	-	+
	Arrêté	+	+
Positionnement au-delà des limites de bloc SPOSA	En déplacement	Alarme 10610	-
Mode axe	En déplacement	-	+
	Arrêté	+	+
+: possible -: impossible			

Couplages d'axes

- Axes pilotes : la modification par échelon de la position de l'axe pilote provoquée par PRESETON n'est pas exécutée dans les axes asservis. Le couplage reste inchangé.
- Axes asservis : seule la composante position forcée de l'axe asservi est affectée par PRESETON.

Groupe Gantry

- Si PRESETON est mis en œuvre sur l'axe directeur d'un groupe gantry, le décalage d'origine est également exécuté dans tous les axes synchrones du groupe gantry.

Axes indexés

- PRESETON peut être mis en œuvre sur des axes indexés.

Fin de course logiciel, limitation de la zone de travail, zones de protection

- Si, après un décalage d'origine par PRESETON, la position de l'axe se trouve en dehors d'une des limites mentionnées, une alarme n'est affichée que si une tentative de déplacement de l'axe a lieu.

Recherche de bloc avec calcul

Les instructions PRESETON sont collectées pendant la recherche de bloc et exécutées avec le départ CN pour la poursuite du programme CN.

Signaux d'interface CN/AP dépendant de la position

- L'état des signaux d'interface dépendant de la position CN/AP est déterminé à nouveau en raison de la nouvelle position réelle.
Exemple : positions de points fixes
 - Positions de points fixes paramétrées : PM30600 \$MA_FIX_POINT_POS[0...3] = <position du point fixe 1...4>
 - Signaux d'interface CN/AP : <Axis>.basic.in.jogFixedPointReachedNumber (accostage d'un point fixe JOG : atteint)
- Si l'axe se trouve dans la limite d'arrêt précis sur une position de point fixe, le signal d'interface CN/AP correspondant est activé. Lorsque la valeur réelle est réglée par PRESETON sur une autre valeur en dehors de la limite d'arrêt précis au niveau de la position du point fixe, le signal d'interface est réinitialisé.

Décalage DRF

- PRESETON supprime le décalage DRF de l'axe.

Déplacement forcé \$AA_OFF

- Un déplacement forcé (\$AA_OFF) (Page 51) n'est pas affecté par PRESETON.

Correction d'outil en ligne FTOC

- Une correction d'outil en ligne (FTOC) (Page 81) active reste également active après PRESETON.

Compensations spécifiques à un axe

Les compensations spécifiques à un axe restent actives après PRESETON.

Mode de fonctionnement JOG

- PRESETON ne peut être mis en œuvre que sur un axe à l'arrêt.

Mode JOG, fonction machine REF

- PRESETON ne doit **pas** être mis en œuvre.

Signaux AP

CN → AP

Basic Program Plus	Basic Program	
<Axis>.basic.in.enc1Synchronized	LBP_Axis*.E_RefSyn1	DB31,DBX60.4
<Axis>.basic.in.enc2Synchronized	LBP_AxisX*.E_RefSyn2	DB31,DBX60.5
<Axis>.basic.in.jogFixedPointReachedNumber	LBP_Axis*.E_JogFixPPos0 ... 2	DB31,DBX75.3..5

Voir aussi

Tronçonnage au vol (Page 155)

4.7.16 Forçage de valeurs réelles sans perte de l'état de référencement (PRESETONS)

Fonction

À partir d'actions synchrones, la procédure `PRESETONS` () définit de nouvelles valeurs réelles pour **un** axe dans le système de coordonnées machine (SCM). Cela correspond à un décalage d'origine du SCM de l'axe. Ainsi, l'axe ne se déplace pas.

À partir d'actions synchrones, `PRESETONS` peut être mis en œuvre **uniquement sur des axes de commande**, c'est-à-dire des axes qui ont été démarrés à partir d'une action synchrone. L'axe doit ensuite être affecté au canal qui doit disposer du droit d'interpolation pour cet axe. L'axe n'est **pas** demandé par permutation d'axe par un autre canal.

État de référencement

La définition d'une nouvelle valeur réelle dans le système de coordonnées machine (SCM) avec `PRESETONS` ne modifie **pas** l'état de référencement de l'axe machine.

Conditions

- **Type de codeur**

`PRESETONS` n'est possible que pour les types de codeurs suivants du système de mesure actif :

- PM30240 \$MA_ENC_TYPE[<système de mesure>] = 0 (codeur simulé)
- PM30240 \$MA_ENC_TYPE[<système de mesure>] = 1 (codeur de signal brut)

- **Mode de référencement**

`PRESETONS` n'est possible que pour les modes de référencement suivants du système de mesure actif :

- PM34200 \$MA_ENC_REFP_MODE[<système de mesure>] = 0 (aucun accostage du point de référence possible)
- PM34200 \$MA_ENC_REFP_MODE[<système de mesure>] = 1 (référencement de systèmes de mesure relatifs, rotatifs ou linéaires : impulsion top zéro sur la voie du codeur)

Mise en service

Paramètres machine spécifiques à un axe

Le forçage de valeurs réelles sans perte de l'état de référencement (`PRESETONS`) doit être activé pour chaque axe :

MD30460 \$MA_BASE_FUNCTION_MASK, Bit 9 = 1

Remarque

PRESETON désactivé

L'activation de la fonction "Forçage de valeurs réelles sans perte de l'état de référencement `PRESETONS`" entraîne la désactivation de la fonction "Forçage de valeurs réelles avec perte de l'état de référencement (`PRESETON`)". Les deux fonctions s'excluent mutuellement.

Programmation

Syntaxe

<fréquence> ... DO PRESETONS (<axe>, <valeur>)

Signification

<fréquence> : Pour la fréquence (Page 21), seuls WHEN et EVERY peuvent être utilisés.
 PRESETONS : Forçage de valeur réelle sans perte de l'état de référencement
 <axe> : Nom de l'axe machine
 Type : AXIS
 Plage de valeurs : noms d'axe machine définis dans le canal
 <valeur> : Nouvelle valeur réelle courante de l'axe machine dans le système de coordonnées machine (SCM)
 La saisie s'effectue dans le système d'unités actif (anglo-saxon/métrique)
 Une programmation active de diamètre (DIAMON) est prise en compte
 Type : REAL

Variable système

\$AC_PRESET

La variable système spécifique à l'axe \$AC_PRESET fournit le vecteur du point d'origine du SCM décalé actuel au point d'origine du SCM_0 d'origine après référencement de l'axe machine.

$\$AC_PRESET<axe> = \$AC_PRESET<axe> + \text{"Position réelle actuelle de l'axe dans le SCM"} - \text{"PRESETONS-position réelle"}$

Le décalage d'origine peut être à nouveau annulé à l'aide des variables système :

$PRESETONS(<axe>, \$VA_IM + \$AC_PRESET[<axe>]) ; \text{"Valeur réelle actuelle de l'axe dans le SCM"} + \text{"Décalages"}$

Exemple

Décalage d'origine du SCM de l'axe X de 70 unités.

La position de fin programmée de l'axe X (axe de commande) est transformée avec PRESETONS dans le nouveau SCM.

Code de programme

```
N10 G1 X=10 F5000
; Déplacement de l'axe X comme axe de commande en position 200
N20 WHEN TRUE DO G71 POS[X]=200
; IF position de consigne de l'axe X dans le SCM ($AA_IM[X]) >= 80
; THEN "Position réelle de l'axe X dans le SCM" = "Position de consigne de
l'axe X dans le SCM" + "Décalage"
;                                     = 80 + 70 = 150
; "Progr. position finale de l'axe X" = "Progr. position finale de
l'axe X" + "Décalage"
;                                     = 200 + 70 = 270
; $AC_PRESET = $AC_PRESET - 70
```

Code de programme

```
N30 WHEN G71 $AA_IM[X] >= 80 DO PRESETONS(X, $AA_IM[X]+70)
N40 G4 F3
```

Autres conditions à prendre en compte

Axes pour lesquels **PRESETONS** ne peut pas être mis en œuvre

- Axes de commande se déplaçant en mode broche
- Axes de positionnement concurrents se déplaçant (SINU_TraversePosAxis (Basic Program FC18))
- Axes qui participent à une transformation
- Axes à interpolation se déplaçant
- Axes d'oscillation
- Axes pour lesquels une ou plusieurs des fonctions de sécurité (Safety Integrated) suivantes sont actives :
 - Déblocage "Fins de course logiciels sûrs"
PM36901 \$MA_SAFE_FUNCTION_ENABLE[<SafeAxis>], bit 1 = 1
 - Déblocage "Came logicielle sûre", paire 1 ... 4, came +/-
PM36901 \$MA_SAFE_FUNCTION_ENABLE[<SafeAxis>], bit 8 ... 15 = 1
ou
Déblocage "Piste de cames sûre", came 1 ... 30
PM36903 \$MA_SAFE_CAM_ENABLE[<SafeAxis>], bit 0 ... 29 = 1
- Axes Hirth
- Axes synchrones d'un groupe Gantry
- Axes pour lesquels l'accostage du point de référence du programme pièce (G74) est actif
- Axe subordonné d'un couplage vitesse/couple de rotation (couplage principal/subordonné)

Axes géométriques

- PRESETONS peut être mis en œuvre sur un axe géométrique à l'arrêt si aucun autre axe géométrique ne se déplace dans le canal en même temps.
- PRESETONS peut être mis en œuvre sur un axe géométrique à l'arrêt même si un autre axe géométrique se déplace dans le canal en même temps, si cet axe se trouve à l'état "Axe neutre" ou se déplace comme axe de commande.
Exemple : Un autre axe géométrique (X) se déplace en même temps à l'état "**Axe neutre**"

Code de programme	Commentaire
N10 G0 X0 Y0	; X, Y : axes géométriques
N15 RELEASE(Y) ¹⁾	; Axe neutre
N20 ID=1 WHEN 20.0 < \$AA_IM[X] DO PRESETONS(Y,20)	; \$AA_IM : position de consigne dans le SCM
N30 G0 X40	; L'axe géométrique X se déplace
N40 M30	
1) Remarque Si l'axe est débloquenté dans la partie action de l'action synchrone, il n'y a aucune garantie que le déblocage aura lieu à temps. N20 ID=1 WHEN 20.0 < \$AA_IM[X] DO RELEASE(Y) PRESETONS(Y,20) ; NON recommandé !	

Exemple : Un autre axe géométrique (X) se déplace en même temps que l'**axe de commande**

Code de programme	Commentaire
N10 G0 X0 Y0	; X, Y : axes géométriques
N20 ID=1 WHEN TRUE DO POS[X]=40 FA[X]=1000	; Axe de commande X
N30 ID=2 WHEN 20.0 < \$AA_IM[X] DO PRESETONS(Y,20)	; \$AA_IM : position de consigne dans le SCM
N40 M30	

Axes contrôlés par l'AP

- PRESETONS peut être mis en œuvre sur des axes contrôlés par l'AP en fonction de leur type actuel.

États de la broche

Le tableau suivant indique les réactions qui se déclenchent quand PRESETONS est mis en œuvre dans une action synchrone sur une broche :

PRESETONS dans une action synchrone			
Mode de fonctionnement de la broche	État de déplacement	Affecté au programme CN	Axe d'exécution
Mode de régulation de vitesse	En déplacement	Alarme 17601	Alarme 17601
	Arrêté	+	+
Mode Positionnement SPOS	Déplacement	Alarme 17601	Alarme 17601
	Arrêté	+	+
Positionnement au-delà des limites de bloc SPOSA	En déplacement	Alarme 17601	-

PRESETONS dans une action synchrone			
Mode de fonctionnement de la broche	État de déplacement	Affecté au programme CN	Axe d'exécution
Mode axe	En déplacement	Alarme 17601	+
	Arrêté	+	+
+: possible -: impossible			

PRESETONS dans le programme CN			
Mode de fonctionnement de la broche	État de déplacement	Affecté au programme CN	Axe d'exécution
Mode de régulation de vitesse	En déplacement	Alarme 22324	Alarme 22324
	Arrêté	+	+
Mode Positionnement SPOS	Déplacement	-	+
	Arrêté	+	+
Positionnement au-delà des limites de bloc SPOSA	En déplacement	Alarme 10610	-
Mode axe	En déplacement	-	+
	Arrêté	+	+
+: possible -: impossible			

Couplages d'axes

- Axes pilotes : La modification par échelon de la position de l'axe pilote provoquée par PRESETONS n'est pas exécutée dans les axes asservis. Le couplage reste inchangé.
- Axes asservis : Seule la composante position forcée de l'axe asservi est affectée par PRESETONS.

Groupe Gantry

- Si PRESETONS est mis en œuvre sur l'axe directeur d'un groupe gantry, le décalage d'origine est également exécuté dans tous les axes synchrones du groupe gantry.

Axes indexés

- PRESETONS peut être mis en œuvre sur des axes indexés.

Fin de course logiciel, limitation de la zone de travail, zones de protection

- Si, après un décalage d'origine par PRESETONS, la position de l'axe se trouve en dehors d'une des limites mentionnées, une alarme n'est affichée que si une tentative de déplacement de l'axe a lieu.

Recherche de bloc avec calcul

Les instructions PRESETONS sont collectées pendant la recherche de bloc et exécutées avec le départ CN pour la poursuite du programme CN.

Signaux d'interface CN/AP dépendant de la position

- L'état des signaux d'interface dépendant de la position CN/AP est déterminé à nouveau en raison de la nouvelle position réelle.
Exemple : positions de points fixes
 - Positions de points fixes paramétrées : PM30600 \$MA_FIX_POINT_POS[0...3] = <position du point fixe 1...4>
 - Signaux d'interface CN/AP : <Axis>.basic.in.jogFixedPointReachedNumber (accostage d'un point fixe JOG : atteint)

Si l'axe se trouve dans la limite d'arrêt précis sur une position de point fixe, le signal d'interface CN/AP correspondant est activé. Lorsque la valeur réelle est réglée par PRESETONS sur une autre valeur en dehors de la limite d'arrêt précis au niveau de la position du point fixe, le signal d'interface est réinitialisé.

Décalage DRF

- PRESETONS supprime le décalage DRF de l'axe.

Déplacement forcé \$AA_OFF

- Un déplacement forcé (\$AA_OFF) (Page 51) n'est pas affecté par PRESETONS.

Correction d'outil en ligne FTOC

- Une correction d'outil en ligne (FTOC) (Page 81) active reste également active après PRESETONS.

Compensations spécifiques à un axe

Les compensations spécifiques à un axe restent actives après PRESETONS.

Mode de fonctionnement JOG

- PRESETONS ne peut être mis en œuvre que sur un axe à l'arrêt.

Mode JOG, fonction machine REF

- PRESETONS ne doit **pas** être mis en œuvre.

Signaux AP

Basic Program Plus	Basic Program	
<Axis>.basic.in.jogFixedPointReachedNumber	LBP_Axis*.E_JogFixPPos0 ... 2	DB31,DBX75.3..5

4.7.17 Couplages (CP..., LEAD..., TRAIL..., CTAB...)

Dans les actions synchrones, il est possible pour les fonctions Déplacements conjugués (TRAIL...), Tables de courbes (CTAB...), Couplage par valeur pilote (LEAD...) et Couplage générique (CP...) de programmer les instructions présentées au chapitre "Éléments de langage pour les actions synchrones et les cycles technologiques (Page 68)".

Remarque

Couplage générique

Il est à noter que les instructions de "Couplage générique" CP... dans les actions synchrones sont toujours traitées dans l'ordre de programmation de gauche à droite. Autrement dit, contrairement à la programmation dans le programme pièce, l'effet des différentes instructions dépend de l'ordre dans l'action synchrone.

Tables de courbes

Les instructions CTAB et CTABINV peuvent être utilisées dans la condition et dans l'action.

Plus d'informations

Pour des informations détaillées sur les instructions de couplage, voir :

- Déplacements conjugués, tables de courbes, couplage par valeur pilote
Manuel de programmation Programmation CN
- Couplage générique
Description fonctionnelle Axes et broches

Déplacements conjugués

Quand le couplage est activé à partir de l'action synchrone, l'axe pilote peut être en cours de déplacement. Dans ce cas, l'axe asservi subit une accélération pour atteindre la vitesse de consigne. La position de l'axe pilote au moment de la synchronisation des vitesses constitue la position de départ des déplacements conjugués.

Couplage par valeur pilote

Syntaxe

```
... DO LEADON (<FA>, <LA>, <NR>, <OVW>)
```

Signification

<FA> :	Nom de l'axe asservi Type : AXIS
<LA> :	Nom de l'axe pilote Type : AXIS

<NR> : Numéro de la table de courbes
Type : INT

<OVW> : État de l'autorisation d'écrasement
Type : BOOL
0 : L'écrasement de la table n'est **pas** autorisé
1 : L'écrasement de la table est autorisé

- La table de courbes de référence peut également être modifiée sans resynchronisation pendant un couplage par valeur pilote actif au moyen d'actions synchrones. L'axe asservi tente alors au plus vite de suivre la valeur de position indiquée par la nouvelle table de courbes.
- Pour pouvoir programmer un axe à coupler au moyen d'actions synchrones, l'axe doit tout d'abord être débloquenté avec RELEASE.
Exemple :

Code de programme

```
...
N60 RELEASE(X)
N50 ID=1 EVERY SR1==1 DO LEADON(C, X, 1)
```

Exemple : Tronçonnage au vol

Un tube, qui traverse à vitesse continue la zone de travail d'une scie, doit être découpé en tronçons de même longueur.

- Axe X : axe dans lequel se déplace le tube (SCP)
- Axe X1 : axe machine du tube (SCM
)
- Axe Y : axe dans lequel la scie "se déplace avec" le tube

On suppose que le déplacement de la scie et sa commande sont pilotés par le programme AP utilisateur. Pour vérifier le synchronisme entre le tube et la scie, il est possible d'exploiter les signaux de l'interface AP.

Code de programme

```
N100 R3=1500 ; longueur d'une pièce à tronçonner
N200 R2=100000 R13=R2/300
N300 R4=100000
N400 R6=30 ; position de départ de l'axe Y
N500 R1=1 ; condition de départ pour l'axe de la bande
N600 LEADOF(Y,X) ; suppression du couplage
N700 CTABDEF(Y,X,1,0) ; définition de table
N800 X=30 Y=30 ; paires de valeurs
N900 X=R13 Y=R13
N1000 X=2*R13 Y=30
N1100 CTABEND ; fin de la définition de table
N1200 PRESETON(X1,0) ; PRESET au début
N1300 Y=R6 G0 ; position de départ de l'axe Y, l'axe est linéaire
```

Code de programme

```
; PRESET après longueur R3, nouveau départ après
tronçonnage
N1400 ID=1 WHENEVER $AA_IW[X]>$R3 DO PRESETON(X1,0)
N1500 RELEASE(Y)
; Couplage d'Y à X par la table 1 lorsque X < 10
N1800 ID=6 EVERY $AA_IM[X]<10 DO LEADON(Y,X,1)
; Séparation du couplage à > 30 avant la longueur de tronçonnage parcourue
N1900 ID=10 EVERY $AA_IM[X]>$R3-30 DO LEADOF(Y,X)
N2000 WAITP(X)
; Mise en mouvement continue de l'axe du tube
N2100 ID=7 WHEN $R1==1 DO MOV[X]=1 FA[X]=$R4
N2200 M30
```

Couplage générique

- Lors de l'activation d'un module de couplage dans une action synchrone, l'axe asservi **doit** déjà être actif dans le canal et se trouver à l'état "Axe neutre" ou "Axe déjà affecté au programme pièce du canal". Si nécessaire, l'état de l'axe correspondant peut être généré par la programmation de GET[<AxeAsservi>] dans l'action synchrone.
- Les instructions du couplage générique CP... sont traitées directement par le module de couplage dans les actions synchrones. Ainsi, l'instruction est effective immédiatement.
- La programmation d'un facteur de couplage (CPLNUM, CPLDEN) ou d'un numéro de table (CPLCTID) désactive un rapport de couplage non linéaire précédemment activé, par exemple une table de courbes.

Couplage générique : utilisation du type de couplage TRAIL, LEAD, EG ou COUP.

Si, dans le cadre du couplage générique, le comportement souhaité correspond à l'un des types de couplage connus "Déplacements conjugués", "Couplage par valeur pilote", "Réducteur électronique" ou "Broche synchrone", l'instruction CPSETTYPE est en outre possible dans les actions synchrones lors de la création ou de la définition du module de couplage :

CPSETTYPE[FAx] = <type de couplage>

<type de couplage>	Signification
CP	Libre programmabilité
TRAIL	Type de couplage "Déplacements conjugués"
LEAD	Type de couplage "Couplage par valeur pilote"
EG	Type de couplage "Réducteur électronique"
COUP	Type de couplage "Broche synchrone"

Autres conditions à prendre en compte**État de synchronisme d'un axe asservi**

La variable système \$AA_SYNC[<axe>] permet de lire l'état de synchronisme d'un axe asservi dans le programme pièce ou l'action synchrone.

Permutation d'axe en cas de couplage dans tous les canaux

Lors de la permutation d'axe, les axes asservis et pilotes doivent être connus du canal appelant. La permutation d'axes pilotes est possible indépendamment de l'état du couplage. Un couplage défini ou actif n'entraîne aucune autre condition à prendre en compte.

Remarque

Lorsque le couplage est activé, l'axe asservi devient l'axe d'exécution des blocs et n'est pas disponible pour une permutation d'axe. L'axe asservi est alors supprimé du canal. Un déplacement forcé n'est donc pas possible avec ce type de couplage.

Voir aussi le chapitre "Permutation d'axe (GET, RELEASE, AXTOCHAN) (Page 94)".

Gestion des conflits lors du passage d'un axe asservi à un axe de canal

Pour pouvoir à nouveau déplacer comme axe de canal un axe asservi qui a été déplacé par des actions synchrones, il faut veiller à ce que le couplage soit désactivé avant que le canal ne demande l'axe en question.

L'exemple ci-dessous montre un **cas d'erreur** :

Code de programme

```
...
N50 WHEN TRUE DO TRAILOF(Y, X)
N60 Y100
```

L'axe Y n'est pas débloquent à temps dans N50, car TRAILOF est seulement actif avec N60 en raison de l'action synchrone non modale.

Exemple corrigé :

Code de programme

```
...
N50 WHEN TRUE DO TRAILOF(Y, X)
N55 WAITP(Y)          ; Attente de la fin du déplacement de l'axe de
positionnement
N60 Y100
```

Exemples

Définition du couplage : Y = axe asservi, X = axe pilote

Code de programme

```
... DO CPLDEF[Y]=X CPLNUM[Y,X]=1.5
```

Activation du couplage et définition du rapport de couplage

- N10 avec ordre correct : d'abord CPLON, puis CPLNUM
- N20 avec ordre **erroné** : d'abord CPLNUM, puis CPLON

Code de programme

```
N10 ... DO CPLON[Y]=X CPLNUM[X,Y]=1.5  
N20 ... DO CPLNUM[X,Y]=2 CPLON[Y]=X ; défaut
```

Activation du couplage, activation/désactivation avec resynchronisation implicite

Code de programme

```
N10 ... DO CPLON[X]=Y CPLNUM[X,Y]=3  
N20 Y100 F100  
N30 ... DO CPLOF=X CPLON[X]=Y CPLNUM[X,Y]=3
```

Activation du couplage, désactivation et déplacement en tant qu'axe de commande

Code de programme

```
N10 ... DO CPLON[X]=Y CPLNUM[X,Y]=3  
N20 Y100 F100  
N30 ... DO CPLOF=X MOV[X]=10
```

4.7.18 Mesure (MEAWA, MEAC)

Les instructions suivantes peuvent être utilisées pour la mesure dans les actions synchrones :

- MEAWA (Mesure sans effacement de la distance restant à parcourir)
- MEAC (Mesure continue sans effacement de la distance restant à parcourir)

Tandis que la fonction de mesure est limitée à un seul bloc de déplacement dans le programme pièce, elle peut être activée et désactivée à volonté à partir d'actions synchrones.

Remarque

Les actions synchrones statiques IDS . . . permettent également de mesurer en mode de fonctionnement JOG.

Informations complémentaires

Pour obtenir des informations détaillées sur les instructions de mesure, voir :

- Déplacements conjugués, tables de courbes, couplage par valeur pilote
Manuel de programmation Programmation CN
- Couplage générique
Description fonctionnelle Axes et broches

Tâches de mesure et modifications des états

Si la tâche de mesure a été effectuée à partir d'actions synchrones, la commande affiche le comportement suivant :

État	Comportement
Changement de mode de fonctionnement	Une tâche de mesure qui a été activée par une action synchrone modale n'est pas affectée par le changement de mode de fonctionnement. Elle reste effective au-delà des limites de bloc.
Reset	La tâche de mesure est interrompue.
Recherche de bloc	Les tâches de mesure sont collectées et activées uniquement lorsque la condition programmée est remplie.
REPOS	Les tâches de mesure activées ne sont pas affectées.
Fin du programme	Les tâches de mesure qui ont été lancées à partir d'actions synchrones statiques sont conservées.

Remarques

Variable système

Les variables système suivantes peuvent être utilisées en relation avec des actions synchrones :

- \$AA_MEAACT (Mesure axiale active)
- \$A_PROBE (État du palpeur)
- \$AA_MM1 ... 4 (Position du palpeur du 1er au 4e déclencheur (SCM))

La variable système suivante ne peut **pas** être utilisée en relation avec des actions synchrones :

- \$AC_MEA (Le palpeur s'est déclenché)

Tâche de mesure

Une seule tâche de mesure peut être active par axe.

Priorité en cas de mesures multiples

Une nouvelle tâche de mesure pour le même axe entraîne la réactivation des événements déclencheurs et la réinitialisation des résultats de mesure.

Les tâches de mesure lancées à partir du programme pièce ne peuvent pas être influencées par des actions synchrones. Si une tâche de mesure est lancée à partir d'une action synchrone pour un axe pour lequel une tâche de mesure est déjà active à partir du programme pièce, une alarme s'affiche.

Si une tâche de mesure est active à partir d'une action synchrone, aucune mesure ne peut plus être lancée à partir du programme pièce.

Enregistrement des résultats de mesure

Afin d'enregistrer les résultats de mesure, une mémoire FIFO est configurée dans les variables système \$AC_FIFO. Voir à ce propos le chapitre "Variables FIFO (\$AC_FIFO) (Page 36)".

Exemples

Pour les exemples suivants, 2 mémoires FIFO sont configurées avec des paramètres machine :

- MD28050 \$MC_MM_NUM_R_PARAM = 300
- MD28258 \$MC_MM_NUM_AC_TIMER = 1
- MD28260 \$MC_NUM_AC_FIFO = 1 (configurer la mémoire FIFO)
- MD28262 \$MC_START_AC_FIFO = 100 (la mémoire FIFO commence à R100)
- MD28264 \$MC_LEN_AC_FIFO = 28 (22 variables + 6 données de gestion)
- MD28266 \$MC_MODE_AC_FIFO = 0 (pas de calcul de la somme)

Exemple 1

Pour l'axe X, tous les flancs montants du palpeur 1 doivent être détectés entre 0 et 100 mm. On suppose que pas plus de 22 fronts de mesure peuvent se produire.

Code de programme	Commentaire
DEF INT NOMBRE	; Nombre de valeurs de mesure actuelles
DEF INT INDICE_R	; Indice de rectification
N10 GO X0	; Accostage du point de départ de la mesure
; Mesure : Mode=1 (simultanément), mémoire FIFO=1,	
; Événement déclencheur=1 (front montant du palpeur 1)	
N20 MEAC[X]=(1, 1, 1) POS[X]=100	
N30 STOPRE	; Arrêt du prétraitement
N40 MEAC[X]=(0)	; Annulation de la mesure
N50 ANZAHL=\$AC_FIFO1[4]	; Nombre de valeurs de mesure enregistrées
N60 NOMBRE = NOMBRE - 1	
N70 FOR INDICE_R=0 TO NOMBRE	
N80 R[INDICE_R]=\$AC_FIFO1[0]	; Enregistrement de la valeur de mesure dans par. R
N90 ENDFOR	

Exemple 2

Pour l'axe X, tous les flancs montants et descendants du palpeur 1 doivent être détectés entre 0 et 100 mm. Le nombre de mesures n'est pas connu. Par conséquent, les valeurs de mesure doivent être collectées parallèlement à la mesure et enregistrées dans l'ordre croissant à partir de \$R1. Le nombre de valeurs de mesure enregistrées est entré dans \$R0.

Code de programme	Commentaire
\$AC_MARKER[1]=1	; Initialisation de l'indice pour le paramètre R
N10 GO X0	; Accostage du point de départ de la mesure
; Si une valeur de mesure est présente dans la mémoire FIFO, la valeur la plus ancienne est lue et	
; enregistrée dans le paramètre R actuel [\$AC_MARKER[1]].	
; L'indice du paramètre R est ensuite incrémenté.	
N20 ID=1 WHENEVER \$AC_FIFO1[4] >= 1 DO \$R[\$AC_MARKER[1]] = \$AC_FIFO1[0]	
	\$AC_MARKER[1] = \$AC_MARKER[1] + 1

4.7 Actions synchrones

Code de programme	Commentaire
; Mesure continue : Mode=1 (simultanément), mémoire FIFO=1,	
; Événement déclencheur 1=1 (front montant du palpeur 1),	
; Événement déclencheur 2=-1 (front descendant du palpeur 1)	
N30 MEAC[X]=(1, 1, 1, -1) POS[X]=100	
N40 MEAC[X]=(0)	; Désactivation de la mesure
N50 STOPRE	; Arrêt du prétraitement
N60 R0 = \$AC_MARKER[1]	; Nombre de valeurs de mesure collectées

Exemple 3

Pour l'axe X, les flancs montants et descendants du palpeur 1 doivent être détectés entre 0 et 500 mm. Le nombre de mesures est limité à 10.

La distance restant à parcourir de l'axe X est ensuite supprimée.

Code de programme	Commentaire
N10 G0 X0	; Accostage du point de départ de la mesure
; Condition d'abandon : désélection de la mesure continue après 10 mesures ou plus	
; et exécution de "Effacer la distance restant à parcourir"	
N10 WHEN \$AC_FIFO1[4] >= 10 DO MEAC[X]=(0) DELDTG(X)	
; Mesure continue : Mode=1 (simultanément), mémoire FIFO=1,	
; Événement déclencheur 1=1 (front montant du palpeur 1),	
; Événement déclencheur 2=-1 (front descendant du palpeur 1)	
N20 MEAC[X]=(1, 1, 1, -1) G01 X100 F500	
N30 MEAC[X]=(0)	; Désactivation de la mesure
N40 R0 = \$AC_FIFO1[4]	; Nombre de valeurs de mesure collectées

4.7.19 Accostage d'une butée (FXS, FXST, FXSW, FOCON, FOCOF, FOC)

Fonction

Accostage de butée

Les instructions FXS, FXST et FXSW permettent de commander la fonction "Accostage de butée" via des actions synchrones.

L'activation peut également être effectuée sans déplacement de l'axe correspondant. Le couple est immédiatement limité. La butée est surveillée dès que l'axe est déplacé.

Déplacement avec réduction de couple/force

Les instructions FOCON, FOCOF et FOC permettent de commander le déplacement avec réduction de couple/force via des actions synchrones.

Syntaxe

FXS [<Axe>]=<Demande>

FXST[<Axe>]=<Couple de blocage>
FXSW[<Axe>]=<Largeur de fenêtre>
FOCON[<Axe>]
FOCOF[<Axe>]
FOC[<Axe>]

Signification

Paramètre	Signification
FXS :	Accostage de butée
<Demande> :	Demande à la fonction "Accostage de butée" : 0 = Désactivation 1 = Activation
FXST :	Réglage du couple de blocage
<Couple de blocage> :	Couple de blocage en % du couple maximal de l'entraînement
FXSW :	Réglage de la fenêtre de surveillance
<Largeur de fenêtre> :	Largeur de la fenêtre de tolérance autour de la butée Unité : mm, inch ou degré
FOCON :	Activation de la limitation de couple/force à effet modal
FOCOF :	Désactivation de la limitation de couple/force à effet modal
FOC :	Limitation de couple/force non modale
<axe> :	Nom de l'axe de canal auquel l'instruction est appliquée

Commentaires

Eviter les activations multiples

La fonction "Accostage de butée" ne doit être activée qu'une fois par axe. En cas d'erreur, l'alarme 20092 s'affiche et la réaction correspondante est activée.

Pour éviter les sélections multiples, il est recommandé d'utiliser un memento de sélection dans l'action synchrone.

Exemple :

Code de programme	Commentaire
N10 R1=0	; Initialisation du memento de sélection
...	
N20 IDS=1 WHENEVER (\$R1==0 AND \$AA_IW[AX3] > 7) DO \$R1=1 FXS[AX1]=1	

Activation pendant l'accostage

Une action synchrone non modale permet d'activer "Accostage de butée" même pendant l'accostage.

Exemple :

Code de programme	Commentaire
N10 G0 G90 X0 Y0	; Accostage de la position de base

4.7 Actions synchrones

Code de programme	Commentaire
...	
; "Accostage de butée" est activé pour l'axe X,	
, dès que la position de consigne est > 20 mm dans le SCP	
; exécution de l'action synchrone à effet non modal : avec N30	
N20 WHEN G71 \$AA_IW[X] > 20 DO FXS[X]=1	
N30 G1 F200 X100	; Bloc de déplacement de l'axe X

Exemple : accostage de butée entièrement via les actions synchrones

Code de programme	Commentaire
; SI demande d'activation \$R1==1 ET état de l'axe Y == "pas en butée"	
; ALORS : Pour l'axe Y :	
; - activer FXS	
; - Déplacement à la position 150 mm	
; - Réduire la couple d'entraînement à 10 %	
IDS=1 WHENEVER G71 ((\$R1==1) AND \$AA_FXS[Y]==0)) DO \$R1=0 FXS[Y]=1 FXST[Y]=10	
	FA[Y]=200 POS[Y]=150
...	
; SI état de l'axe Y == "butée détectée"	
; ALORS : Augmenter le couple d'entraînement à 30 %	
IDS=2 WHENEVER (\$AA_FXS[Y]==4) DO FXST[Y]=30	
...	
; SI état de l'axe Y == "butée accostée avec succès"	
; ALORS : Régler le couple d'entraînement selon la définition \$R0	
IDS=3 WHENEVER (\$AA_FXS[Y]==1) DO FXST[Y]=\$R0	
...	
; Désactivation en fonction de R3 et retour.	
IDS=4 WHENEVER ((\$R3==1) AND \$AA_FXS[Y]==1)) DO FXS[Y]=0 FA[Y]=1000 POS[Y]=0	
...	
N10 R1=0 FXS[Y]=0 G0 G90 Y0	; Initialisation
N30 RELEASE(Y)	; Débloquer l'axe Y pour le déplacement dans les actions syn-
	chrones
N50 ...	
N60 GET(Y)	; Réintégrer l'axe Y dans le groupe d'interpolation

4.7.20 Synchronisation de canal (SETM, CLEARM)

Les instructions SETM et CLEARM permettent de définir et de supprimer des marques de synchronisation dans le canal dans lequel l'action synchrone s'exécute.

Syntaxe

SETM(<n°_marqueur 1> [, <n°_marqueur 2> {, ... <n°_marqueur n>}])

```
CLEARM(<n°_marqueur 1> [, <n°_marqueur 2> {, ... <n°_marqueur n>} ] )
```

Signification

Une description détaillée des instructions SETM et CLEARM est disponible dans :

Informations complémentaires

Manuel de programmation Programmation CN

4.7.21 Réaction aux erreurs spécifique à l'utilisateur (SETAL)

Via les actions synchrones, il est possible d'obtenir des réactions spécifiques à l'utilisateur pour des états d'erreur spécifiques à l'application. Les réactions possibles sont :

- Axe avec arrêt via correction = 0 %
- Affichage d'une alarme spécifique à l'utilisateur
- Mise à 1 de sortie TOR

Afficher alarme

Syntaxe

```
SETAL(<n°_alarme>[, "Texte alarme"])
```

Signification

Paramètre	Signification
<n°_alarme> :	Numéro d'alarme dans la plage : 65000 - 69999

Une description complète de la configuration des alarmes utilisateur est fournie dans :

Informations complémentaires

Manuel de mise en service Logiciel de base

Exemples

```
; Si la distance entre les axes X1 et X2 et inférieure à 5 mm =>  
; Arrêt de l'axe X2  
ID=1 WHENEVER G71 ($AA_IM[X1]-$AA_IM[X2])<5.0 DO $AA_OVR[X2]=0  
  
; Si la distance entre les axes X1 et X2 et inférieure à 5 mm =>  
; Affichage de l'alarme 65000  
ID=1 WHENEVER G71 ($AA_IM[X1]-$AA_IM[X2])<5.0 DO SETAL(65000)
```

4.7.22 Interruption du niveau de sous-programme actuel (CANCELSUB)

À l'aide de CANCELSUB, le programme CN actif dans le niveau de sous-programme actuel est interrompu dans le canal dans lequel l'action synchrone est exécutée et il est renvoyé dans le programme appelant du niveau de programme supérieur suivant. Le déroulement du programme s'y poursuit normalement.

Propriétés

- Seul le niveau de sous-programme actuel est interrompu pour chaque appel.
- Après une interruption, le niveau de sous-programme supérieur suivant ne peut être interrompu que lorsque le retour du niveau de sous-programme interrompu a été effectué.
- Le niveau de programme principal ne peut pas être interrompu.

Syntaxe

CANCELSUB

Signification

CANCELSUB :	Interruption du niveau de sous-programme actuel
-------------	---

Voir aussi

L'interruption du niveau de sous-programme actuel peut également être effectuée de façon équivalente par le programme AP utilisateur avec le signal d'interface CN/AP spécifique au canal :
<Chan>.basic.out.abortProgLevel (Interruption de niveau de programme)

Signaux AP

AP → CN

Basic Program Plus	Basic Program	
<Chan>.basic.out.abortProgLevel	LBP_Chان*.A_ProgAbort	DB21,DBX6.4

4.8 Cycles technologiques

4.8.1 Généralités

Définition

Un cycle technologique désigne un programme CN appelé dans la partie action d'une action synchrone. Tous les éléments de langage et toutes les variables système pouvant être utilisés dans la partie action d'une action synchrone peuvent être également utilisés dans un cycle technologique. Il existe en outre les éléments de langage supplémentaires suivants qui peuvent uniquement être utilisés dans un cycle technologique :

- Chapitre "Variables définies par l'utilisateur pour actions synchrones (Page 66)"
- Chapitre "Éléments de langage pour les actions synchrones et les cycles technologiques (Page 68)"
- Chapitre "Eléments de langage pour cycles technologiques uniquement (Page 73)"
- Chapitre "Variable système pour actions synchrones (Page 25)"
- Chapitre "Actions synchrones (Page 74)"

Fin du programme

Les instructions suivantes sont autorisées comme fin de programme : M02, M17, M30, RET

Chemin de recherche

Lors de l'appel d'un cycle technologique, c'est le même chemin de recherche que pour les sous-programmes et les cycles qui est utilisé.

Informations complémentaires

Manuel de programmation Programmation CN

Appels multiples

Si une condition est de nouveau remplie pendant l'exécution du cycle technologique, celui-ci n'est **pas** démarré une nouvelle fois.

Si un cycle technologique est démarré parce que la condition `WHENEVER` est remplie et si cette condition est toujours remplie à la fin du cycle technologique, celui-ci est démarré une nouvelle fois.

Comportement avec des actions synchrones non modales

Une action synchrone non modale est toujours liée au bloc principal suivant. Si le temps d'exécution du cycle technologique est supérieur au temps de traitement du bloc principal correspondant, le cycle technologique est interrompu au changement de bloc.

Séquence d'exécution des cycles technologiques

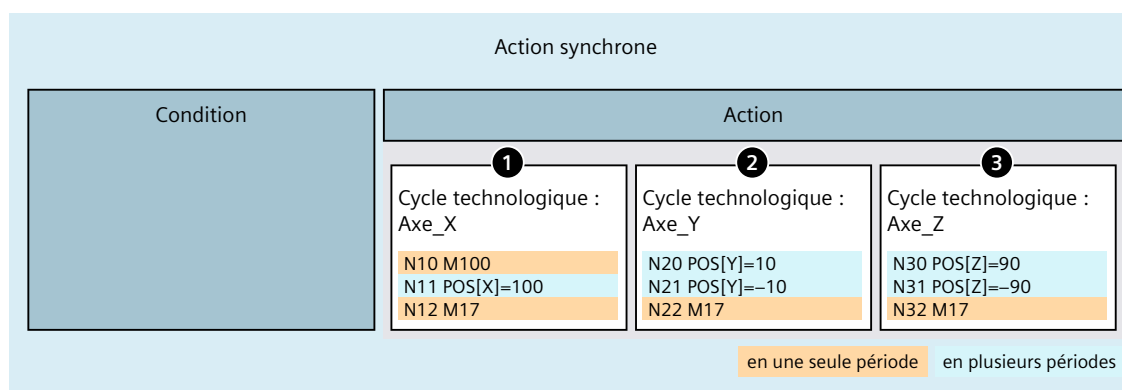
Si plusieurs cycles technologiques sont programmés dans la partie action d'une action synchrone, ils sont exécutés dans l'ordre programmé, de gauche à droite.

Exemple

Appel de trois cycles technologiques dans la partie action d'une action synchrone.

Code de programme

```
ID=1 <partie condition> DO AXE_X AXE_Y AXE_Z
```



Séquence d'exécution des blocs des cycles technologiques : N10, N11, N12, N20, N21, N22, N30, N31, N32

Remarque

Autres conditions

- Dans la partie action d'une action synchrone, 8 cycles technologiques peuvent être appelés au maximum.
- Dans la partie action d'une action synchrone dans laquelle un cycle technologique est appelé, aucune autre action que l'appel d'autres cycles technologiques ne peut être programmée.

4.8.2 Mode d'exécution (ICYCON, ICYCOF)

Fonction

Les instructions **ICYCOF** et **ICYCON** permettent de commander le mode d'exécution des actions dans des cycles technologiques.

Par défaut, le mode d'exécution **ICYCON** est actif.

Mode d'exécution : ICYCON

Avec le mode d'exécution **ICYCON**, un cycle technologique est exécuté de manière non modale. L'exécution de toutes les actions programmées dans un bloc est déclenchée dans la même période d'appel de l'interpolateur. Dès que toutes les actions déclenchées sont terminées, le bloc suivant est exécuté dans la période d'appel de l'interpolateur suivante.

On différencie les actions nécessitant une ou plusieurs périodes d'appel. Par exemple :

- Actions à une période d'appel : sortie de fonctions auxiliaires, affectations de valeurs
- Actions à plusieurs périodes d'appel : déplacements d'axes et de broches

Chaque bloc d'un cycle technologique nécessite au moins une période d'appel de l'interpolateur.

Mode d'exécution : ICYCOF

En mode d'exécution **ICYCOF**, toutes les actions de tous les blocs d'un cycle technologique sont déclenchées simultanément en parallèle.

Programme CN en tant que programme pièce

Si un programme CN est exécuté en tant que programme pièce, les instructions **ICYCOF** et **ICYCON** n'ont pas d'effet.

Syntaxe

Dans la partie action d'une action synchrone

```
ID=1 <Partie_condition> DO [ICYCOF] <Cycle_technologique_1> [ICYCOF
| ICYCON] <Cycle_technologique_2> ...
```

En tant qu'attribut d'un programme CN

```
PROC <Nom> [ICYCOF | ICYCON]
```

Au sein d'un programme CN

```
PROC <Nom>
  N10 ...
  N20 [ICYCOF | ICYCON]
  N90 ...
  N100 [ICYCOF | ICYCON]
  N110 ...
RET
```

Exemple

Code de programme	Mode d'exécution actif	Cycle d'interpolation
PROC TECHNOCYC	ICYCON	
\$R1=1	ICYCON	1
POS[X]=100	ICYCON	2 ... 25
ICYCOF	ICYCOF	26
\$R1=2	ICYCOF	26
\$R2=\$R1+1	ICYCOF	26
POS[X]=110	ICYCOF	26
\$R3=3	ICYCOF	26
RET	ICYCOF	26

4.8.3 Définitions (DEF, DEFINE)

Si un programme CN qui contient des instructions de définition d'une variable (`DEF`) et/ou d'une macro (`DEFINE`) est utilisé comme cycle technologique, celles-ci n'ont **pas d'effet** lors de l'exécution du cycle technologique.

Bien que les définitions de variables et de macros dans un cycle technologique n'aient pas d'effet, elles doivent être correctes du point de vue de la syntaxe. En cas d'erreur, l'exécution du cycle technologique est interrompue et une alarme s'affiche.

Comme les variables et les macros du cycle technologique ne sont pas disponibles, il est parfois nécessaire de prendre des mesures particulières dans le code du programme. Voir chapitre "Variable de contexte (`$P_TECCYCLE`) (Page 128)".

4.8.4 Transfert de paramètres

Dans un sous-programme utilisé comme cycle technologique, seul le transfert de paramètre Call-by-Value est possible.

L'utilisation de paramètres Call-by-Reference n'est **pas** autorisée et déclenche une alarme spécifique.

Informations complémentaires

Pour une description détaillée du transfert de paramètres et de la définition de paramètres pour les sous-programmes, voir :

Manuel de programmation Programmation CN

4.8.5 Variable de contexte (`$P_TECCYCLE`)

Fonction

Si un programme CN est utilisé aussi bien en tant que programme pièce qu'en tant que cycle technologique, des sections de programme spécifiques au contexte peuvent être définies via la variable système `$P_TECCYCLE` :

- `$P_TECCYCLE == TRUE` ⇒ Le programme CN est actuellement exécuté en tant que cycle technologique
- `$P_TECCYCLE == FALSE` ⇒ Le programme CN est actuellement exécuté en tant que programme pièce

Application

Dans les cycles technologiques, les définitions de variables (`DEF`) et de macros (`DEFINE`) n'ont pas d'effet. Si un programme CN est exécuté en tant que cycle technologique contenant des définitions correspondantes, il est nécessaire que le code du programme contienne une distinction de cas d'utilisation spécifique au contexte, car les variables et les macros ne sont alors pas disponibles.

Exemple

Paramètre de déplacement via des variables utilisateur dans le programme pièce et le paramètre R dans le cycle technologique

Code de programme	Commentaire : Utilisation dans
PROC UP_1	
DEF REAL POS_X=100.0	Programme pièce
DEF REAL F_X=250.0	Programme pièce
IF \$P_TECCYCLE==TRUE	
\$R1=100.0	Cycle technologique
\$R2=250.0	Cycle technologique
ENDIF	
IF \$P_TECCYCLE==TRUE	
N100 POS[X]=\$R1 FA[X]=\$R2	Cycle technologique
ELSE	
N200 POS[X]=POS_X FA[X]=F_X	Programme pièce
ENDIF	
RET	

Voir aussi

Définitions (DEF, DEFINE) (Page 128)

4.9 Actions synchrones protégées

Chaque action synchrone est clairement identifiée par son numéro d'identification ID.

Les paramètres machine suivants permettent de définir globalement, pour toute la CN, ou spécifiquement pour un canal une plage de numéros d'identification permettant de protéger une action synchrone afin qu'elle ne puisse pas être écrasée, supprimée (CANCEL (ID)) ni bloquée (LOCK (ID)) :

- Globalement pour la CN
MD11500 \$MN_PREVENT_SYNACT_LOCK (actions synchrones protégées)
- Spécifiquement pour un canal :
MD21240 \$MC_PREVENT_SYNACT_LOCK_CHAN (actions synchrones protégées)

Dans les deux cas, le comportement est le même.

Les actions synchrones protégées ne peuvent pas être bloquées via l'interface CN/AP et sont affichées comme ne pouvant pas être bloquées :

- <Chan>.synActs[1].out.lock (Bloquer des actions synchrones)
- <Chan>.synActs[1].in.lockable (Actions synchrones blocables)

Application

Les actions synchrones définies par le constructeur de la machine pour la réaction à certains états de la machine ne doivent plus pouvoir être influencées après la mise en service.

Remarque

Il est recommandé de ne pas activer la protection des actions synchrones pendant la phase de mise en service afin d'éviter de devoir effectuer un Power On-Reset à chaque modification de l'action synchrone.

Exemple

Dans un système à 2 canaux, les actions synchrones des plages de numéros d'identification suivantes doivent être protégées :

Canal 1 : 20 ... 30

Canal 2 : 25 ... 35

Configuration des paramètres machine

Zone de protection globale de la CN :

- MD11500 \$MN_PREVENT_SYNACT_LOCK[0] = 25
- MD11500 \$MN_PREVENT_SYNACT_LOCK[1] = 35

Zone de protection spécifique du canal 1 :

- MD21240 \$MC_PREVENT_SYNACT_LOCK_CHAN[0] = 20
- MD21240 \$MC_PREVENT_SYNACT_LOCK_CHAN1] = 30

Zone de protection spécifique du canal 2 :

- MD21241 \$MC_PREVENT_SYNACT_LOCK_CHAN[0] = -1
- MD21241 \$MC_PREVENT_SYNACT_LOCK_CHAN[1] = -1

Aucune zone de protection spécifique n'a été définie dans le canal 2, la zone de protection globale de la CN étant donc valable.

Signaux AP

AP → CN

Basic Program Plus	Basic Program	
<Chan>.synActs[1].out.lock	LBP_Chan*.A_SynDisabl[1] - [64]	DB21,DBX300.0..307.7

CN → AP

Basic Program Plus	Basic Program	
<Chan>.synActs[1].in.lockable	LBP_Chan*.E_SynDisabled[1] - [64]	DB21,DBX308.0..315.7

4.10 Coordination par programme pièce et action synchrone (LOCK, UNLOCK, CANCEL)

Chaque action synchrone modale et statique doit se voir attribuer un numéro d'identification univoque lors de la définition :

Code de programme
ID=<Numéro> Partie condition DO Partie action
IDS=<Numéro> Partie condition DO Partie action

Les instructions suivantes permettent de coordonner des actions synchrones à partir de programmes pièce et d'actions synchrones en indiquant le numéro d'identification :

Mot-clé	Signification	PP ¹⁾	AS ²⁾
LOCK (<numéro>) :	Blocage de l'action synchrone Tout positionnement actif est interrompu.	-	x
UNLOCK (<numéro>) :	Reprendre une action synchrone interrompue Tout positionnement interrompu est repris.	-	x
CANCEL (<numéro>) :	Effacer l'action synchrone Tout positionnement actif est terminé.	x	-
¹⁾ Programmable dans le programme pièce			
²⁾ Programmable dans une action synchrone / un cycle technologique			

4.11 Coordination via AP

En ce qui concerne leur exécution par la CN, les actions synchrones non protégées peuvent être bloquées. Il est possible de bloquer toutes les actions synchrones du canal en même temps ou des actions synchrones individuelles dans la plage ID/IDS 1 - 64.

Blocage global, spécifique au canal

Bloquer toutes les actions synchrones du canal :

```
<Chan>.basic.out.lockAllSynActs = 1 (Bloquer toutes les actions synchrones)
```

Blocage individuel, spécifique au canal

Actions synchrones blocables

Les actions synchrones ID/IDS blocables s'affichent via :

```
<Chan>.synActs[1].in.lockable == 1 (Actions synchrones ID/IDS blocables)
```

Bloquer des actions synchrones

Pour chaque action synchrone ID/IDS devant être bloquée dans le canal, le signal de blocage correspondant doit être activé par le programme AP utilisateur :

```
<Chan>.synActs[1].out.lock = 1 (Bloquer l'action synchrone ID/IDS)
```

Signaux AP

AP → CN

Basic Program Plus	Basic Program	
<Chan>.basic.out.lockAllSynActs	LBP_Chan*.A_SynOff	DB21,DBX1.2
<Chan>.synActs[1].out.lock	LBP_Chan*.A_SynDisabl[1] - [64]	DB21,DBX300.0..307.7

CN → AP

Basic Program Plus	Basic Program	
<Chan>.synActs[1].in.lockable	LBP_Chan*.E_SynDisabled[1] - [64]	DB21,DBX308..315.7

4.12 Configuration

Nombre d'éléments d'action synchrone

Le nombre d'éléments d'action synchrone devant être mis à disposition par canal se règle avec le paramètre machine :

MD28250 \$MC_MM_NUM_SYNC_ELEMENTS (nombre d'éléments pour les expressions dans les actions synchrones)

Chaque action synchrone a besoin d'au moins 4 éléments d'action synchrone. Des éléments d'action synchrone supplémentaires sont nécessaires dans les cas suivants :

Instruction	Nombre d'éléments nécessaires
Opérateur dans la condition	1
Action	>= 1
Affectation	2
Opérandes supplémentaires dans les expressions complexes	1

Le nombre d'actions synchrones programmables dépend donc du nombre d'éléments d'action synchrone disponibles et de la complexité des actions synchrones.

Capacité mémoire

L'affichage d'état des actions synchrones (voir chapitre "Diagnostic (Page 139)") permet de suivre la capacité de la mémoire d'action synchrone.

Le nombre d'éléments d'action synchrone libres peut également être lu avec la variable système \$AC_SYNA_MEM.

Si le nombre d'éléments d'action synchrone nécessaires augmente pendant le fonctionnement et dépasse le nombre disponible, l'alarme "14751 Trop peu de ressources pour les actions synchrones au déplacement" s'affiche.

4.13 Comportement de la commande dans des états de fonctionnement donnés

Nombre d'éléments FCTDEF

Le nombre d'éléments `FCTDEF` par canal se règle avec le paramètre machine :

MD28252 `$MC_MM_NUM_FCTDEF_ELEMENTS` (nombre d'éléments FCTDEF)

Actions synchrones et fréquence d'interpolation

Si de nombreuses actions synchrones sont activées simultanément, il peut être nécessaire d'augmenter la fréquence d'interpolation :

PM10070 `$MN_IPO_SYSCLOCK_TIME_RATIO`

Tableau 4-1 Temps nécessaire pour différentes opérations

Instructions d'action synchrone	Temps nécessaire ¹⁾	
	Total	Partie en gras
Charge de base d'une action synchrone si la condition n'est pas remplie : WHENEVER FALSE DO \$AC_MARKER[0]=0	10 µs	10 µs
Lecture de variable : WHENEVER \$AA_IM[Y]>10 DO \$AC_MARKER[0]=1	11 µs	1 µs
Écriture de variable : DO \$R2=1	11-12 µs	1-2 µs
Lecture/écriture de donnée de réglage : DO \$\$\$SN_SW_CAM_MINUS_POS_TAB_1[0]=20	24 µs	14 µs
Opérations de base, par exemple multiplication : DO \$R2=\$R2*2	22 µs	12 µs
Fonctions trigonométriques (par exemple cos) : DO \$R2=COS(\$R2)	23 µs	13 µs
Démarrage de l'axe de positionnement : WHEN TRUE DO POS[z]=10	83 µs	73 µs
1) mesuré avec SINUMERIK 840D sl avec NCU 7x0.3 PN		

4.13 Comportement de la commande dans des états de fonctionnement donnés

4.13.1 Power On

Pendant le démarrage de la CN (Power On), aucune action synchrone n'est activée.

Les actions synchrones devant être activées immédiatement après le démarrage de la CN (Power On) doivent être activées comme actions synchrones statiques par un événement à l'intérieur d'un ASUP, ou par le programme AP utilisateur.

Informations complémentaires

Pour plus d'informations sur l'activation d'actions synchrones après le démarrage de la CN (Power On), voir :

Programme AP utilisateur

Description fonctionnelle AP

Déclenché par événement

Description fonctionnelle Fonctions de base

4.13.2 Reset CN

État après un reset CN :

De :	Action synchrone modale et non modale (ID)	Action synchrone statique (IDS)
Action synchrone	abandonnée ou désactivée	activée
Déplacement	Les déplacements sont interrompus.	
Broche en asservissement de vitesse	MD35040 \$MA_SPIND_ACTIVE_AFTER_RESET = <valeur> TRUE ⇒ la broche reste activée. FALSE ⇒ la broche s'arrête.	
Couplage par valeur pilote	MD20110 \$MC_RESET_MODE_MASK, bit13 = <valeur> 1 ⇒ le couplage reste activé. 0 ⇒ le couplage est supprimé.	
Mesure	abandonnée	

4.13.3 Arrêt CN

Actions synchrones modales et non modales (ID)

Après un arrêt CN, les déplacements des actions synchrones modales et non modales sont suspendus.

Pendant que l'état du canal est "Interrompu" (`<Chan>.basic.in.stateInterrupted == 1`), l'action synchrone modale ou non modale reste activée. Si, pendant ce temps, la condition est remplie, les actions (excepté les déplacements) sont exécutées.

Après un démarrage CN, les déplacements suspendus se poursuivent.

Actions synchrones statiques (IDS)

La suspension par arrêt CN de déplacements d'actions synchrones statiques dépend du composant qui contrôle l'axe de commande :

- La **CN** contrôle l'axe (`<Chan>.basic.in.stateInterrupted == 0`) :
Après un arrêt CN, les déplacements des actions synchrones statiques sont suspendus.
- L'**AP** contrôle l'axe (`<Chan>.basic.in.stateInterrupted == 1`) :
Après un arrêt CN, les déplacements des actions synchrones statiques ne sont **pas** suspendus.

Signaux AP

AP → CN

Basic Program Plus	Basic Program	
<Axis>.plcCtrl.out.requestPlcCtrl	LBP_Axis*.A_PLCAxis	DB31,DBX28.7

CN → AP

Basic Program Plus	Basic Program	
<Chan>.basic.in.stateInterrupted	LBP_Chan*.E_ChanInterrupt	DB21,DBX35.6

4.13.4 Changement de mode de fonctionnement

État après changement du mode de fonctionnement :

De :	Action synchrone modale et non modale (ID)	Action synchrone statique (IDS)
Action synchrone	abandonnée ou désactivée ¹⁾	activée
Déplacement	abandonnée ²⁾	activée
Broche en asservissement de vitesse	activée	activée
Couplage par valeur pilote	MD20110 \$MC_RESET_MODE_MASK, bit13 = <valeur> 1 ⇒ le couplage reste activé. 0 ⇒ le couplage est supprimé.	activée
Mesure	abandonnée	activée
1) Après un nouveau passage au mode de fonctionnement AUTOMATIQUE, la réactivation, ou non, de l'action synchrone dépend des instructions de l'action synchrone (voir exemple).		
2) La fin de programme M30 est temporisé jusqu'à ce que l'axe soit immobilisé.		

Exemple

Cas 1 :

```
ID=1 Do POS[X]=100 FA[X]=100
```

```
G4 F999
```

ID=1 Do signifie : l'action synchrone démarre dans chaque cycle IPO du bloc suivant même si elle a déjà atteint la position "100".

Cas 2 :

Modification de l'action synchrone du cas 1 selon :

```
ID=1 When true Do POS[X]=100 FA[X]=100
```

```
G4 F999
```

ID=1 When true Do signifie : l'action synchrone démarre "une fois" lors du changement de bloc. L'axe se déplace.

Effectuer les manipulations :

4.13 Comportement de la commande dans des états de fonctionnement donnés

Après un arrêt CN et une correction "0", l'axe s'arrête dans les deux cas sans avoir atteint la position finale.

Dans le cas 2, l'action synchrone est terminée après le premier cycle IPO. Dans ce cas, l'axe ne démarre plus lors d'un démarrage CN.

L'axe n'est pas en position finale.

4.13.5 Fin du programme

État après la fin de programme :

De :	Action synchrone modale et non modale (ID)	Action synchrone statique (IDS)
Action synchrone	abandonnée ou désactivée	activée
Déplacement	abandonnée 1)	activée
Broche en asservissement de vitesse	MD35040 \$MA_SPIND_ACTIVE_AFTER_RESET = <valeur> TRUE ⇒ la broche reste activée. FALSE ⇒ la broche s'arrête.	activée
Couplage par valeur pilote	MD20110 \$MC_RESET_MODE_MASK, bit13 = <valeur> 1 ⇒ le couplage reste activé. 0 ⇒ le couplage est supprimé.	activée
Mesure	abandonnée	activée

1) La fin de programme M30 est temporisé jusqu'à ce que l'axe soit immobilisé.

4.13.6 Recherche de bloc

Les actions synchrones sont collectées pendant la recherche de bloc, mais pas activées. Cela signifie que les conditions ne sont pas évaluées et que les actions ne sont pas exécutées.

Les actions synchrones ne sont activées qu'après un démarrage CN. Cela signifie que les conditions sont évaluées et que les actions sont exécutées le cas échéant.

Actions synchrones statiques (IDS)

Les actions synchrones statiques déjà activées restent activées pendant la recherche de bloc. Les actions étant toutefois en plus collectées, il est possible que le nombre d'éléments d'action synchrone nécessaires soit plus grand pendant la recherche de bloc que le nombre disponible pour le canal selon le paramétrage de MD28250 \$MC_MM_NUM_SYNC_ELEMENTS. Cela déclencherait l'alarme 14751 "Trop peu de ressources pour les actions synchrones au déplacement" (voir "Configuration (Page 132)").

4.13 Comportement de la commande dans des états de fonctionnement donnés

Afin d'éviter une surcharge de la mémoire pendant la recherche de bloc, la procédure suivante est recommandée :

1. Activer le démarrage ASUP automatique après la recherche de bloc :
MD11450 \$MN_SEARCH_RUN_MODE bit1 = 1
⇒ Après le démarrage CN et le passage au dernier bloc d'action, l'ASUP de recherche de bloc indiqué dans MD11620 \$MN_PROG_EVENT_NAME est activé.
2. Dans l'ASUP de recherche de bloc, à l'intérieur de la demande IF (\$P_PROG_EVENT==5) (l'ASUP a été implicitement activé après la sortie du dernier bloc d'action, après la recherche de bloc), activer les actions synchrones statiques (voir exemple).
3. Dans le programme pièce, mettre les actions synchrones statiques entre parenthèses avec IF (\$P_SEARCH==0) (recherche de bloc non activée) afin que cette section ne soit pas exécutée dans la recherche de bloc.

Exemple :

Sous-programme _N_PROG_EVENT_SP (extrait)

```

DEF INT II, SIDS

IF ($P_PROG_EVENT==5)

FOR II=SIDS TO SIDS+8
CANCEL (II)
ENDFOR
STOPRE
IDS=SIDS DO G710 $AC_MARKER[SACM+2]=(( $AC_MARKER[SACM+2]+1) MOD 2) * (( $A_DBW[72]
B_AND 16) /16)
SIDS=SIDS+1
IDS=SIDS WHENEVER (( $AC_PARAM[SACP+6]) >=ZPOS_CO[1])
DO $AC_PARAM[SACP+4]=SIN((( $AC_PARAM[SACP+6]-ZPOS_CO[0]) / (ZPD_CO)) *90)

... ; Autres actions synchrones

ENDIF

```

4.13.7 Interruption du programme par un ASUP

Actions synchrones modales et non modales (ID)

Les actions synchrones modales activées restent activées pendant l'ASUP.

Les déplacements démarrés par des actions synchrones modales et non modales sont suspendus. Si, à la fin de l'ASUP, la position revient au point d'interruption du programme pièce (REPOS), les déplacements suspendus se poursuivent.

Actions synchrones statiques (IDS)

Les actions synchrones statiques restent activées pendant l'ASUP.

4.13 Comportement de la commande dans des états de fonctionnement donnés

Les déplacements démarrés par des actions synchrones statiques ne sont pas suspendus par l'ASUP.

Actions synchrones de l'ASUP

Si l'ASUP n'est pas poursuivi avec `REPOS`, les actions synchrones modales et statiques de l'ASUP restent actives dans le programme pièce.

4.13.8 REPOS

Dans le bloc restant, les actions synchrones restent actives comme dans le bloc d'interruption.

Les modifications apportées aux actions synchrones modales dans le sous-programme asynchrone ne sont pas actives dans le programme interrompu.

Les coefficients du polynôme programmés avec `FCTDEF` ne sont pas influencés par `ASUP` et `REPOS`.

Dans le sous-programme asynchrone, les coefficients du programme appelant sont actifs. Dans le programme appelant, les coefficients du sous-programme asynchrone sont toujours actifs.

Si, lors d'un changement de mode de fonctionnement ou du démarrage du programme d'interruption, des mouvements de positionnement d'actions synchrones ont été suspendus, ceux-ci sont poursuivis avec `REPOS`.

4.13.9 Comportement en cas d'alarmes

- Si une action d'une action synchrone déclenche une alarme, cette action est abandonnée. Les autres actions de l'action synchrone sont traitées.
- Si une action synchrone modale déclenche une alarme, elle est désactivée à partir de l'instant de l'alarme.
- Si un cycle technologique émet une alarme avec arrêt de déplacement, il est abandonné et son traitement ne se poursuit pas.
- Après une alarme avec arrêt de déplacement, tous les déplacements d'axe et de broche démarrés par des actions synchrones s'arrêtent. Les actions sans déplacements se poursuivent.
- Après une alarme avec arrêt d'interpréteur, l'arrêt d'interpréteur n'agit pour les actions synchrones qu'après l'exécution complète des blocs prédécodés.

4.14 Diagnostic

4.14.1 Afficher l'état des actions synchrones

Pour le diagnostic d'actions synchrones, les informations suivantes s'affichent dans la fenêtre "Actions synchrones", dans le groupe fonctionnel "Machine" :

- Liste des actions synchrones programmées
- Validité et numéro d'identification (seulement pour les actions synchrones modales)
Voir chapitre "Validité, numéros d'identification (ID, IDS) (Page 20)"
- État de l'action synchrone

État	Signification
en attente	La condition est vérifiée dans le cycle IPO.
active	La partie action de l'action synchrone est exécutée. Si l'action est constituée d'un cycle technologique, le numéro de bloc actuel est affiché dans celui-ci.
bloquée	L'action synchrone est bloquée. Voir chapitre : • Coordination par programme pièce et action synchrone (LOCK, UNLOCK, CANCEL) (Page 131) • Coordination via AP (Page 131)

Remarque

Les actions synchrones non modales ne sont identifiées que par l'affichage de leur état. Elles ne sont affichées qu'au cours de l'exécution.

4.14 Diagnostic

Exemples

5.1 Exemples de conditions d'actions synchrones

Condition	Programmation
Distance restante de la trajectoire ≤ 10 mm (SCP)	... WHEN $\$AC_DTEW \leq 10$ DO ...
Distance restant à parcourir par l'axe X ≤ 10 mm (SCP)	... WHEN $\$AA_DTEW[X] \leq 10$ DO ...
Distance de trajectoire par rapport au début de bloc ≥ 20 mm (SCB)	... WHEN $\$AC_PLTBB \geq 20$ DO ...
Valeur réelle de l'axe Y (SCM) $> 10 * \sin(R10)$	... WHEN $\$AA_IM[y] > 10 * \sin(R10)$ DO...
Passage de 0 à 1 de l'entrée 1	... EVERY $\$A_IN[1] == 1$ DO ...
Entrée 1 == 1	... WHENEVER $\$A_IN[1] == 0$ DO ...

5.2 Écriture et lecture de SD/MD d'actions synchrones

Pénétration et oscillation lors de la rectification

Les données de réglage dont les valeurs ne changent pas pendant l'usinage sont traitées comme dans le programme pièce Noms.

Exemple : Oscillation d'actions synchrones

Code de programme
<pre> N610 ID=1 WHENEVER $\\$AA_IM[Z] > \\$SA_OSCILL_REVERSE_POS1[Z]$ DO $\\$AC_MARKER[1]=0$... ; À CHAQUE FOIS que la position actuelle de l'axe d'oscillation est inférieure au début de la zone d'inversion de sens 2 dans le SCM, ; ALORS mettre la correction de l'axe de pénétration à 0 %. N620 ID=2 WHENEVER $\\$AA_IM[Z] < \\$SA_OSCILL_REVERSE_POS2[Z] - 6$ DO $\\$AA_OVR[X]=0$ $\\$AC_MARKER[0]=0$... ; À CHAQUE FOIS que la position actuelle de l'axe d'oscillation est égale à la position d'inversion de sens 1 dans le SCM, ; ALORS mettre la correction de l'axe d'oscillation à 0 %, et la correction de l'axe de pénétration à 100 %. ; Cela supprime l'action synchrone précédente ! N630 ID=3 WHENEVER $\\$AA_IM[Z] == \\$SA_OSCILL_REVERSE_POS1[Z]$ DO $\\$AA_OVR[Z]=0$ $\\$AA_OVR[X]=100$... ; À CHAQUE FOIS que la distance restant à parcourir pour la pénétration partielle est égale à 0, ; ALORS mettre la correction de l'axe d'oscillation à 100 %. ; Cela supprime l'action synchrone précédente ! </pre>

5.2 Écriture et lecture de SD/MD d'actions synchrones

Code de programme

```

N640 ID=4 WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]=100 $AC_MARKER[0]=1 $AC_MARKER[1]=1
N650 ID=5 WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0
N660 ID=6 WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
...
; SI la position actuelle de l'axe d'oscillation est égale à la position d'inversion de sens 1 dans le
SCP,
; ALORS mettre la correction de l'axe d'oscillation à 100 %, et la correction de l'axe de pénétration
à 0 %.
; Cela supprime une fois la deuxième action synchrone !
N670 ID=7 WHEN $AA_IM[Z] == $$SA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[Z]=100 $AA_OVR[X]=0
...
; Les valeurs de réglage dont la valeur change pendant l'usinage (par exemple
; en raison d'une commande ou d'une action synchrone) doivent être programmées avec $$$... :
; Exemple : Oscillations d'actions synchrones avec modification de la position d'oscillation depuis
; l'interface utilisateur
N610 ID=1 WHENEVER $AA_IM[Z] > $$SA_OSCILL_REVERSE_POS1[Z] DO $AC_MARKER[1]=0
...
; À CHAQUE FOIS que la position actuelle de l'axe d'oscillation est inférieure au début de la zone
d'inversion de sens 2 dans le SCM,
; ALORS mettre la correction de l'axe de pénétration à 0 %.
N620 ID=2 WHENEVER $AA_IM[Z] < $$SA_OSCILL_REVERSE_POS2[Z]-6 DO
    $AA_OVR[X]=0 $AC_MARKER[0]=0
...
; À CHAQUE FOIS que la position actuelle de l'axe d'oscillation est égale à la position d'inversion de
sens 1 dans le SCM,
; ALORS mettre la correction de l'axe d'oscillation à 0 %, et la correction de l'axe de pénétration à
100 %.
; Cela supprime l'action synchrone précédente !
N630 ID=3 WHENEVER $AA_IM[Z]==$$SA_OSCILL_REVERSE_POS1[Z] DO
    $AA_OVR[Z]=0 $AA_OVR[X]=100
...
; À CHAQUE FOIS que la distance restant à parcourir pour la pénétration partielle est égale à 0,
; ALORS mettre la correction de l'axe d'oscillation à 100 %.
; Cela supprime l'action synchrone précédente !
N640 ID=4 WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]=100 $AC_MARKER[0]=1 $AC_MARKER[1]=1
N650 ID=5 WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0
N660 ID=6 WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
...
; SI la position actuelle de l'axe d'oscillation est égale à la position d'inversion de sens 1 dans le
SCP,
; ALORS mettre la correction de l'axe d'oscillation à 100 %, et la correction de l'axe de pénétration
à 0 %.
; Cela supprime une fois la deuxième action synchrone !
N670 ID=7 WHEN $AA_IM[Z]==$$SA_OSCILL_REVERSE_POS1[Z]
DO $AA_OVR[Z]=100 $AA_OVR[X]=0

```

5.3 Exemples de régulation AC

Procédure générale

Les exemples suivants utilisent la fonction d'évaluation polynomiale `SYNFCT()`.

1. Représentation de la corrélation entre la valeur d'entrée et la valeur de sortie (variables d'exécution respectives)
2. Définition de ce rapport comme polynôme avec limitations
3. Pour offset de position : définition des MD et SD
 - MD36750 `$MA_AA_OFF_MODE` (effet de l'affectation de valeur pour la superposition axiale d'actions synchrones)
 - SD43350 `$SA_AA_OFF_LIMIT` (facultatif) (limite supérieure de la valeur de correction `$AA_OFF` pour la régulation de la distance)
4. Activation de la régulation dans une action synchrone

5.3.1 Régulation de la distance avec limite supérieure variable

Exemple de polynôme avec limite supérieure dynamique

Pour une régulation de la distance, la limite supérieure de la sortie (`$AA_OFF`, valeur de superposition dans l'axe V) change en fonction de la correction de la broche (entrée analogique 1). Cette limitation supérieure du polynôme 1 change de manière dynamique en fonction de l'entrée analogique 2.

Le polynôme 1 est défini directement avec les variables système :

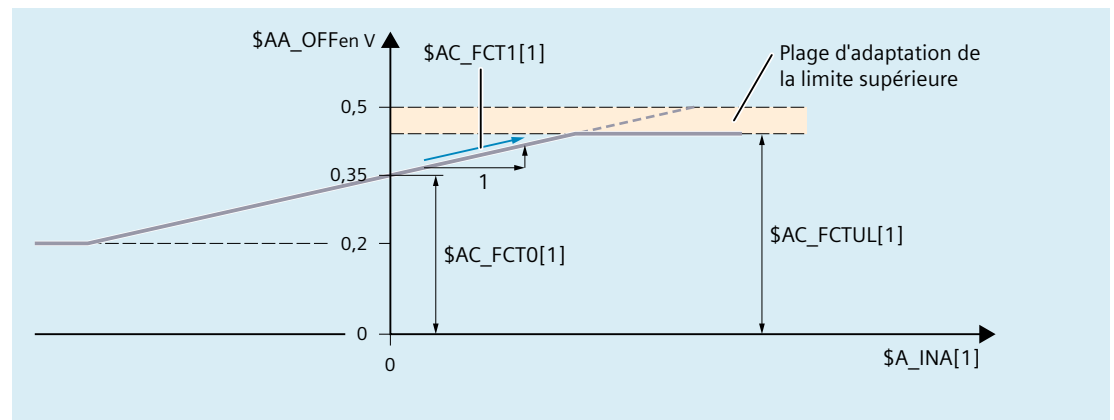


Figure 5-1 Régulation de la distance avec limite supérieure variable

<code>\$AC_FCTL[1]=0.2</code>	; Limitation inférieure
<code>\$AC_FCTUL[1]=0.5</code>	; Début valeur limitation supérieure
<code>\$AC_FCT0[1]=0.35</code>	; Passage par zéro a_0
<code>\$AC_FCT1[1]=1.5 EX-5</code>	; Pas a_1

```

STOPRE                                ; Voir remarque ci-dessous
...
STOPRE                                ; Voir remarque ci-dessous
ID=1 DO $AC_FCTUL[1]=$A_INA[2]*0.1+0.35 ; Adapter la limitation supérieure de
                                         manière dynamique via l'entrée ana-
                                         ; logique 2,
                                         pas de condition
                                         ;
ID=2 DO SYNFACT(1, $AA_OFF[V], $A_INA[1]) ; Régulation de la distance par super-
                                         position, pas de condition
                                         ;
...

```

Remarque

Lorsque des variables système sont utilisées dans le programme pièce, la programmation de **STOPRE** doit assurer la synchronisation de l'écriture avec les blocs. La notation suivante est égale à la notation ci-dessus pour la définition du polynôme :

```
FCTDEF(1,0.2, 0.5, 0.35, 1.5EX-5).
```

5.3.2 Régulation de l'avance

Exemple de régulation AC avec une tension d'entrée analogique

Une grandeur process (mesurée avec $\$A_INA[1]$) doit être réglée sur 2 V par influence additive, par correction de l'avance tangentielle (ou axiale). L'avance doit être corrigée dans des limites de ± 100 [mm/minute].

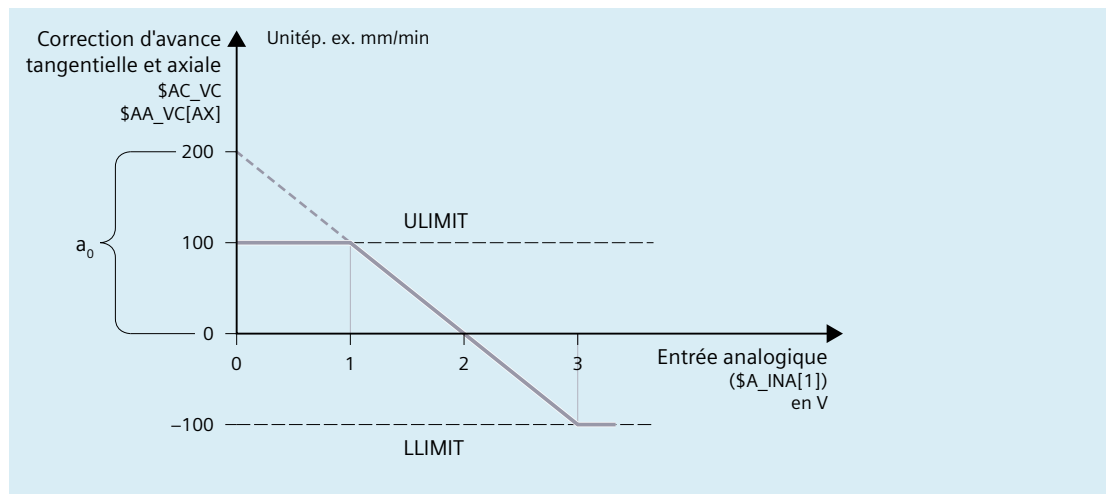


Figure 5-2 Diagramme de régulation AC

Détermination des coefficients :

$$y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$$

$$a_1 = -100 \text{ mm} / (1 \text{ min} * 1 \text{ V})$$

$$a_1 = -100 \% \text{ constante de régulation, pas}$$

$$a_0 = -(-100) * 2 = 200$$

$$a_2 = 0 \text{ (pas de terme quadratique)}$$

$$a_3 = 0 \text{ (pas de terme cubique)}$$

$$\text{limite supérieure} = 100$$

$$\text{limite inférieure} = -100$$

```

FCTDEF(      N° de polynôme,
              LLIMIT,
              ULIMIT,
              a0,                ; y pour x = 0
              a1,                ; pas
              a2,                ; pas de terme quadratique
              a3 )              ; pas de terme cubique

```

Avec les valeurs déterminées ci-dessus, la définition du polynôme est la suivante :

```
FCTDEF(1, -100, 100, 200, -100, 0, 0)
```

La régulation AC peut être activée avec les actions synchrones suivantes :

Pour l'avance paraxiale :

```
ID = 1 DO SYNFACT (1, $AA_VC[X], $A_INA[1])
```

Ou pour l'avance tangentielle :

```
ID = 2 DO SYNFACT(1, $AC_VC, $A_INA[1])
```

5.3.3 Réguler la vitesse en fonction de la trajectoire normalisée

Adaptation multiplicative

La trajectoire normalisée est utilisée comme grandeur d'entrée : \$AC_PATHN.

0 : en début de bloc

1 : en fin de bloc

La grandeur de modification \$AC_OVR doit être régulée en fonction de \$AC_PATHN, selon un polynôme de 3e ordre. La correction doit être réduite de 100 à 1 % pendant le mouvement.

5.4 Surveillance d'une distance de sécurité entre deux axes

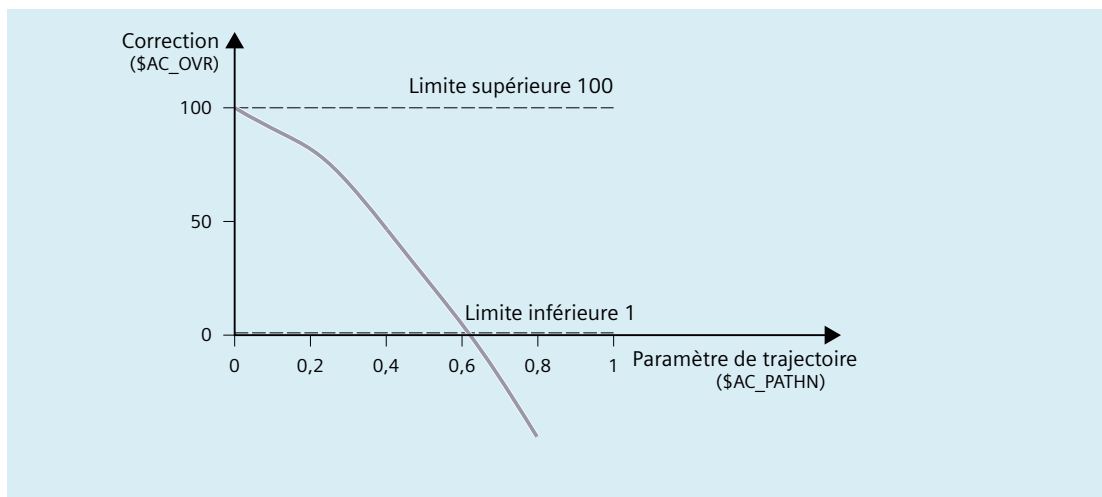


Figure 5-3 Réguler la vitesse en continu

Polynôme 2 :

Limite inférieure : 1

Limite supérieure : 100

a_0 : 100

a_1 : -100

a_2 : -100

a_3 : inutilisé

Avec ces valeurs, la définition du polynôme est la suivante :

```
FCTDEF(2, 1, 100, 100, -100, -100)
```

; Activation de la correction variable en fonction de la trajectoire :

```
ID= 1 DO SYNFACT (2, $AC_OVR, $AC_PATHN)
```

```
G01 X100 Y100 F1000
```

5.4 Surveillance d'une distance de sécurité entre deux axes

Tâche

Les axes X1 et X2 actionnent deux dispositifs de manutention à commandes indépendantes pour le chargement et le déchargement de pièces.

Pour éviter les collisions, une distance de sécurité doit être maintenue entre les deux axes.

Lorsque la distance de sécurité n'est pas maintenue, l'axe X2 freine. Le verrouillage est valable jusqu'à ce que l'axe X1 se trouve à nouveau au-delà de la distance de sécurité.

5.5 Enregistrer les temps d'exécution dans le paramètre R

Si l'axe X1 continue de s'approcher de l'axe X2 et franchit la barrière photoélectrique de sécurité rapprochée, il se place dans une position sûre.

Langage CN	Commentaire
ID=1 WHENEVER \$AA_IM[X2] - \$AA_IM[X1] < 30 DO \$AA_OVR[X2]=0	; Barrière photoélectrique de sécurité
ID=2 EVERY \$AA_IM[X2] - \$AA_IM[X1] < 15 DO POS[X1]=0	; Position sûre

5.5 Enregistrer les temps d'exécution dans le paramètre R

Tâche

Enregistrer le temps d'exécution à partir du paramètre R 10 pour les blocs de programme pièce.

Programme	Commentaire
	; Sans programmation symbolique, l'exemple se ; présente comme suit :
IDS=1 EVERY \$AC_TIMEC==0 DO \$AC_MAR- KER[0] = \$AC_MARKER[0] + 1	; Lors d'un changement de bloc, incrémenter ; l'indicateur de paramètre R
IDS=2 DO \$R[10+\$AC_MARKER[0]] = \$AC_TIME	; Écrire le temps actuel du début ; de bloc dans le paramètre R
	; Avec une programmation symbolique, l'exemple ; se présente comme suit :
DEFINE INDEX AS \$AC_MARKER[0]	; Conventions de programmation ; symbolique
IDS=1 EVERY \$AC_TIMEC==0 DO INDEX = INDEX + 1	; Lors d'un changement de bloc, incrémenter ; l'indicateur de paramètre R
IDS=2 DO \$R[10+INDEX] = \$AC_TIME	; Écrire le temps actuel du début ; de bloc dans le paramètre R

5.6 "Centrer" avec mesure continue

Introduction

Les interstices de roue dentée sont mesurés successivement. La somme des interstices et le nombre de dents permet de calculer la cote d'interstice. La position centrale recherchée pour poursuivre l'usinage correspond à la position du premier point de mesure plus 1/2 de la taille moyenne des interstices. Pour la mesure, la vitesse de rotation est choisie de manière à permettre l'enregistrement fiable d'une valeur de mesure par cycle IPO.

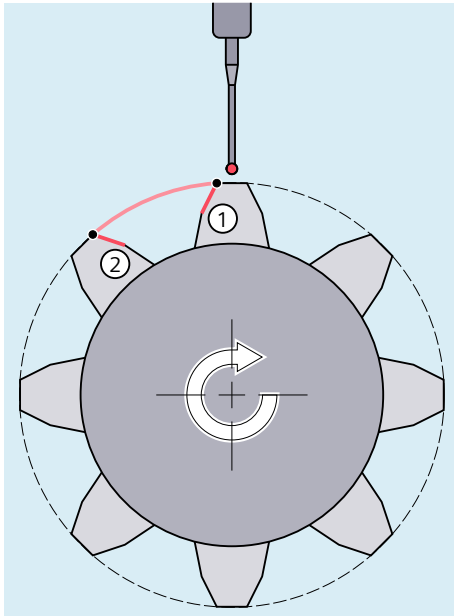


Figure 5-4 Schéma de mesure des interstices de roue dentée

%_N_MEAC_MITTEN_MPF

```
; Mesurer avec l'axe rotatif B (BACH), avec affichage de la différence
; entre les valeurs de mesure
;*** Définir les variables utilisateur locales ***
N1 DEF INT ZAEHNEZahl          ; Saisie du nombre de dents de la roue dentée
N5 DEF REAL HYS_POS_Flanke     ; Hystérésis du front montant du palpeur
N6 DEF REAL HYS_NEG_Flanke     ; Hystérésis du front descendant du palpeur
;*** Définir les noms abrégés des mémentos d'action synchrone ***
define M_ZAEHNE as $AC_Marker[1] ; ID memento de calcul : front descendant/
montant de chaque dent
define Z_MW as $AC_Marker[2]     ; ID lire compteur valeur moyenne FIFO
define Z_RW as $AC_Marker[3]     ; ID compteur valeur moyenne calcul intersti-
ces de dent
;*** Valeurs d'entrée pour la MESURE DE ROUE DENTÉE ***
N50 ZAEHNEZahl=26                ; Saisie du nombre de dents de roue dentée de-
vant être mesurées
N70 HYS_POS_Flanke = 0.160       ; Hystérésis du front montant du palpeur
N80 HYS_NEG_Flanke = 0.140       ; Hystérésis du front descendant du palpeur
```

```

Début : ; *** Affecter les variables ***
R1=0 ; ID2 résultat de calcul cote d'interstice
R2=0 ; ID2 résultat de calcul addition de tous les
interstices
R3=0 ; Contenu du premier élément lu
R4=0 ; R4 correspond à un écart de dent
R5=0 ; Position d'interstice calculée, résultat fi-
nal
R6=1 ; ID 3 activer BACH avec MOV
R7=1 ; ID 5 activer MEAC
M_ZAEHNE=ZAEHNEZ AHL*2 ; ID calculer front descendant/montant de cha-
que dent
Z_MW=0 ; ID lire ID lire compteur valeur moyenne FIFO
jusqu'au nombre de dents
Z_RW=2 ; ID compteur calcul différence interstice de
dent
R13=HYS_POS_FLANKE ; Hystérésis dans registre de calcul
R14=HYS_NEG_FLANKE ; Hystérésis dans registre de calcul
;*** Déplacer l'axe, mesurer, calculer ***
N100 MEAC[BACH]=(0) ; Réinitialiser la tâche de mesure
; Réinitialiser la variable FIFO1[4] et assurer un enregistrement de mesure défini
N105 $AC_FIFO1[4]=0 ; Réinitialiser FIFO1
STOPRE
;*** Lire FIFO jusqu'à ce que le nombre de dents soit atteint ***
; Si FIFO1 n'est pas vide et si les dents n'ont pas encore toutes été mesurées,
transférer la valeur
; de mesure de la variable FIFO dans le paramètre d'action synchrone et augmenter le
compteur des valeurs de mesure

ID=1 WHENEVER ($AC_FIFO1[4]>=1) AND (Z_MW<M_ZAEHNE)
DO $AC_PARAM[0+Z_MW]=$AC_FIFO1[0] Z_MW=Z_MW+1
; Lorsque 2 valeurs de mesure sont disponibles, commencer à calculer, calculer
; SEULEMENT la cote d'interstice et la somme des interstices, augmenter le compteur
de valeur de calcul de 2
ID=2 WHENEVER (Z_MW>=Z_RW) AND (Z_RW<M_ZAEHNE)
DO $R1=($AC_PARAM[-1+Z_RW]-$R13)-($AC_PARAM[-2+Z_RW]-$R14) Z_RW=Z_RW+2
$R2=$R2+$R1
;*** Activation de l'axe BACH en tant qu'axe rotatif tournant en continu avec MOV ***
WAITP(BACH)
; Activer
ID=3 EVERY $R6==1 DO MOV[BACH]=1 FA[BACH]=1000
; Désactiver
ID=4 EVERY $R6==0 und ($AA_STAT[BACH]==1) DO MOV[BACH]=0
; Mesurer successivement, enregistrer dans FIFO 1, front descendant MT2, montant MT2
; Mesurer la distance entre 2 dents
; Front descendant...montant, palpeur 2

```

5.7 Couplages d'axes par actions synchrones

```

N310 ID=5 WHEN $R7==1 DO MEAC[BACH]=(2, 1, -2, 2)
; Arrêter la mesure
N320 ID=6 WHEN (Z_MW>=M_ZAEHNE) DO MEAC[BACH]=(0)
M00
STOPRE

;*** Lire et enregistrer les valeurs FIFO ***
; Contenu du premier élément lu
; Réinitialiser la variable FIFO1[4] et assurer un
; enregistrement de mesure défini pour la tâche de mesure suivante
N400 R3=$AC_PARAM[0]
N500 $AC_FIFO1[4]=0

;*** Calculer la différence entre les différentes dents ***
; R4 correspond à un écart de dent moyen
; Division "/1000" supprimée dans les versions de logiciel postérieures
N510 R4=R2/(ZAEHNEZahl)/1000

;*** Calculer la position centrale ***
N520 R3=R3/1000 ; Première position de mesure convertie en de-
; grés
N530 R3=R3 MOD 360 ; Premier point de mesure modulo
N540 R5=(R3-R14)+(R4/2) ; Calcule la position d'interstice
M00
STOPRE
R6=0 ; Arrêter la rotation d'axe de BACH
GOTOB ANFANG
M30

```

5.7 Couplages d'axes par actions synchrones

5.7.1 Couplage sur axe pilote

Enoncé du problème

Une table de courbes cyclique est définie par des segments polynomiaux. Le mouvement de l'axe pilote et la procédure de couplage de l'axe pilote et de l'axe asservi sont activés/désactivés par des variables de calcul.

5.7 Couplages d'axes par actions synchrones

%_N_KOP_SINUS_MPF

```

N5 R1=1 ; ID 1, 2 activer/désactiver le couplage : LEA-
DON (CACB, BACH)
N6 R2=1 ; ID 3, 4 activer/désactiver le mouvement de
l'axe pilote : MOV BACH
N7 R5=36000 ; BACH avance/min
N8 STOPRE

;*** Définir la table périodique n° 4 par des segments polynomiaux ***
N10 CTABDEF (YGEO,XGEO,4,1)
N16 G1 F1200 XGEO=0.000 YGEO=0.000 ; Accoster les positions initiales
N17 POLY PO[XGEO]=(79.944,3.420,0.210) PO[YGEO]=(24.634,0.871,-9.670)
N18 PO[XGEO]=(116.059,0.749,-0.656) PO[YGEO]=(22.429,-5.201,0.345)
N19 PO[XGEO]=(243.941,-17.234,11.489) PO[YGEO]=(-22.429,-58.844,39.229)
N20 PO[XGEO]=(280.056,1.220,-0.656) PO[YGEO]=(-24.634,4.165,0.345)
N21 PO[XGEO]=(360.000,-4.050,0.210) PO[YGEO]=(0.000,28.139,-9.670)
N22 CTABEND ; *** Fin de la définition de la table ***

; Déplacer l'axe pilote et l'axe asservi en marche rapide jusqu'à la position initiale
N80 G0 BACH=0 CACH=0 ; Noms d'axes de canal
N50 LEADOF(CACH,BACH) ; Désactiver le couplage existant le échéant

N235 ;*** Activation du mouvement de couplage pour l'axe CACH ***
N240 WAITP(CACH) ; Synchroniser l'axe sur le canal
N245 ID=1 EVERY $R1==1 DO LEA- ; Coupler avec la table 4
DON(CACH, BACH, 4)
N250 ID=2 EVERY $R1==0 DO LEA- ; Désactivation du couplage
DOF(CACH, BACH)

N265 WAITP(BACH)
N270 ID=3 EVERY $R2==1 DO ; Faire tourner l'axe pilote en continu avec
MOV[BACH]=1 FA[BACH]=R5 avance dans R5
N275 ID=4 EVERY $R2==0 DO ; Arrêter l'axe pilote
MOV[BACH]=0
N280 M00
N285 STOPRE
N290 R1=0 ; Désactivation condition de couplage
N295 R2=0 ; Désactivation condition pour rotation de
l'axe pilote
N300 R5=180 ; Nouvelle avance pour BACH
N305 M30

```

5.7.2 Rectification non circulaire par couplage par valeur pilote

Enoncé du problème

Une pièce non circulaire tournant sur l'axe CACH doit être rectifiée. L'écart entre la meule et la pièce est défini par l'axe XACH. Il dépend de la rotation de la pièce. La corrélation entre les rotations et les mouvements correspondants est définie par la table de courbes 2. La pièce doit se déplacer à des vitesses dépendant du contour de pièce selon la table de courbes 1.

Solution

CACH devient l'axe pilote d'un couplage par valeur pilote. Elle agit :

- sur le mouvement de compensation de l'axe XACH selon la table 2 ;
- sur l'axe "logiciel" CASW selon la table 1.

La correction de l'axe CACH est déterminée à partir des valeurs réelles de l'axe CASW. Ainsi la vitesse de l'axe CACH exigée en fonction du contour est réalisée.

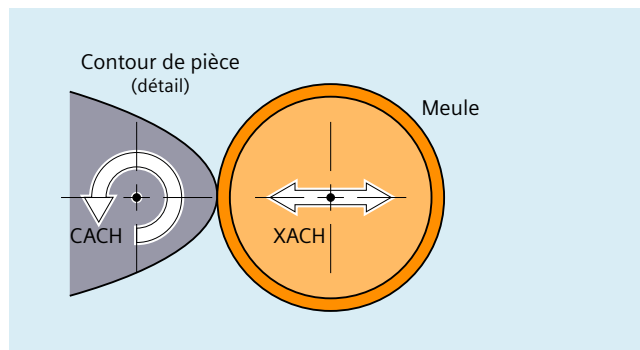


Figure 5-5 Schéma de rectification d'un contour non circulaire

%_N_CURV_TABS_SPF		
PROC CURV_TABS		
N160 ; *** Table 1 Définir la correction ***		
N165 CTABDEF(CASW,CACH,1,1)	;	Table 1 périodique
N170 CACH=0 CASW=10		
N175 CACH=90 CASW=10		
N180 CACH=180 CASW=100		
N185 CACH=350 CASW=10		
N190 CACH=359.999 CASW=10		
N195 CTABEND		

N160 ; *** Table 2 Définir le mouvement de compensation linéaire de l'axe XACH ***		
CTABDEF(YGEO,XGEO,2,1)	;	Table 2 périodique
N16 XGEO=0.000 YGEO=0.000		
N16 XGEO=0.001 YGEO=0.000		

5.7 Couplages d'axes par actions synchrones

N17 POLY PO[XGEO]=(116.000,0.024,0.012) PO[YGEO]=(4.251,0.067,-0.828)	
N18 PO[XGEO]=(244.000,0.072,-0.048) PO[YGEO]=(4.251,-2.937)	
N19 PO[XGEO]=(359.999,-0.060,0.012) PO[YGEO]=(0.000,-2.415,0.828)	
N16 XGEO=360.000 YGEO=0.000	
N20 CTABEND	
M17	

%_N_UNRUND_MPF

; Couple couplé pour usinage non circulaire

; XACH est l'axe de pénétration de la meule.

; CACH est l'axe de la pièce en tant qu'axe rotatif et axe de valeur pilote.

; Application : rectifier un contour non circulaire

; La table 1 représente la correction de l'axe CACH comme fonction de la position de CACH.

; Superposition de l'axe XGEO avec pénétration avec molette pour l'effleurement

N100 DRFOF	; Désactiver la correction par manivelle électronique
N200 MSG("activer DRF, (molette 1 activée) et activation INCRÉMENT.== correction par manivelle électronique ACTIVÉE")	
N300 M00	
N500 MSG()	; Réinitialiser le message
N600 R2=1	; LEADON table 2, activation avec ID=3/4 CACH sur XACH
N700 R3=1	; LEADON table 1, activation avec ID=5/6 CACH sur CASW, correction
N800 R4=1	; Axe rotatif tournant en continu CACH, démarrage avec ID=7/8
N900 R5=36000	; FA[CACH] axe rotatif tournant en continu vitesse de rotation

N1100 STOPRE	
N1200	; *** Régler les axes et l'axe pilote sur AA ***
	; Placer l'axe axe pilote et axe asservi en position initiale
N1300 G0 XGEO=0 CASW=10 CACH=0	
N1400 LEADOF(XACH,CACH)	; Désactiver le couplage XACH mouvement de compensation
N1500 LEADOF(CASW,CACH)	; Désactiver le couplage CASW table de correction
N1600 CURV_TABS	; Sous-programme avec définition des tables

N1700	; *** Interrupteur LEADON mouvement de compensation XACH ***
N1800 WAITP(XGEO)	; Synchroniser l'axe sur le canal

5.7 Couplages d'axes par actions synchrones

N1900 ID=3 EVERY \$R2==1 DO LEADON(XACH,CACH,2)	
N2000 ID=4 EVERY \$R2==0 DO LEADOF(XACH,CACH)	

N2100	; *** Interrupteur LEADON CASW ; table de correction ***
N2200 WAITP(CASW)	
N2300 ID=5 EVERY \$R3==1 DO LEADON(CASW,CACH,1)	; CTAB activation du couplage axe pilote CACH
N2400 ID=6 EVERY \$R3==0 DO LEADOF(CASW,CACH)	; CTAB désactivation du couplage axe pilote CACH

N2500	; *** Correction de CACH de la position ; Influencer CASW avec ID 10 ***
N2700 ID=11 DO \$ \$AA_OVR[CACH]=\$AA_IM[CASW]	; Affecter "position d'axe" CASW sur OVR CACH

N2900 WAITP(CACH)	
N3000 ID=7 EVERY \$R4==1 DO MOV[CACH]=1 FA[CACH]=R5	; Démarrer en tant qu'axe rotatif tournant en continu
N3100 ID=8 EVERY \$R4==0 DO MOV[CACH]=0	; Arrêter en tant qu'axe rotatif tournant en continu

N3200 STOPRE	
N3300 R90=\$AA_COUP_ACT[CASW]	; État du couplage pour CASW pour le contrôle
N3400 MSG("Table de correction CASW activé avec LEADON "<<R90<<", puis FIN avec DÉMARRAGE CN")	

N3500 M00	; *** ARRÊT CN ***
N3600 MSG()	
N3700 STOPRE	; Arrêt du prétraitement des blocs
N3800 R1=0	; Arrêt avec ID=2 CASW axe en tant ; qu'axe rotatif tournant en continu
N3900 R2=0	; LEADOF avec ID=6 FA XACH ; et axe pilote CACH
N4000 R3=0	; LEADOF TAB1 CASW avec ID=7/8 CACH ; sur CASW table de correction
N4100 R4=0	; Arrêter l'axe en tant qu'axe rotatif tournant ; en continu, ID=4 CACH
N4200 M30	

Extensions possibles

L'exemple ci-dessus peut être étendu comme suit :

- Introduction d'un axe Z pour déplacer la meule ou la pièce d'un élément non circulaire à l'autre sur le même arbre (arbre à came).
- Changements de tables lorsque les cames ont des contours différents par exemple pour l'entrée et la sortie.
ID = ... <condition> DO LEADOF(XACH, CACH) LEADON(XACH, CACH, <nouveau numéro de table>)
- Dressage de la meule par correction de pièce en ligne conformément au chapitre "Correction d'outil en ligne FTOC".

5.7.3 Tronçonnage au vol

Enoncé du problème

Un tube, qui traverse à vitesse continue la zone de travail d'une scie, doit être découpé en tronçons de même longueur.

Axe X : axe de déplacement du tube, SCP

Axe X1 : axe machine du tube, SCM

Axe Y : axe dans lequel la scie "se déplace avec" le tube

On suppose que le déplacement de la scie et sa commande sont pilotés par l'AP. Pour vérifier le synchronisme entre le tube et la scie, il est possible d'exploiter les signaux de l'interface AP.

Actions

Activer le couplage, LEADON

Désactiver le couplage, LEADOF

Forçage de valeur réelle, PRESETON

Code de programme	Commentaire
%_N_SCHERE1_MPF	
;\$PATH=/_N_WKS_DIR/_N_DEMOFBE_WPD	
N100 R3=1500	; Longueur d'une pièce à tronçonner
N200 R2=100000 R13=R2/300	
N300 R4=100000	
N400 R6=30	; Position de départ de l'axe Y
N500 R1=1	; Condition de départ pour l'axe de la bande
N600 LEADOF(Y,X)	; Suppression d'un éventuel couplage existant
N700 CTABDEF(Y,X,1,0)	; Définition de table
N800 X=30 Y=30	; Paires de valeurs

5.8 Cycles technologiques de positionnement de la broche

Code de programme	Commentaire
N900 X=R13 Y=R13	
N1000 X=2*R13 Y=30	
N1100 CTABEND	; Fin de la définition de table
N1200 PRESETON(X1,0)	; Décalage Preset au début
N1300 Y=R6 G0	; Position de départ de l'axe Y
	; L'axe est un axe linéaire.
N1400 ID=1 EVERY \$AA_IW[X]>\$R3 DO PRESETON(X1,0)	; Décalage Preset après la longueur R3, PRESETON uniquement avec WHEN et EVERY
	; Nouveau départ après le tronçonnage
N1500 WAITP(Y)	
N1800 ID=6 EVERY \$AA_IM[X] < 10 DO LEADON(Y,X,1)	; Couplage d'Y à X par la table 1 lors- que X < 10
N1900 ID=10 EVERY \$AA_IM[X] > \$R3-30 DO LEADOF(Y,X)	; Séparation du couplage à > 30 avant la longueur de tronçonnage parcourue
N2000 WAITP(X)	
N2100 ID=7 WHEN \$R1==1 DO MOV[X]=1 FA[X]=\$R4	; Mise en mouvement continue de l'axe du tube
N2200 M30	

5.8 Cycles technologiques de positionnement de la broche

Application

En interaction avec le programme AP, la broche qui réalise un changement d'outil doit :

- être placée dans une position initiale ;
- être placée sur une valeur définie où se trouve l'outil pour le changement d'outil.

Voir chap. "Démarrer des axes de commande" chap. "Influence de l'AP".

Coordination

La coordination AP / NCK est réalisée à partir de la version de logiciel 4 avec les données communes disponibles (voir chap. "Liste des variables système essentielles des actions synchrones").

- \$A_DBB[0] : 1 placer en position initiale
- \$A_DBB[1] : 1 placer en position cible
- \$A_DBW[1] : valeur de positionnement +/-, chemin le plus court calculé par l'AP

Actions synchrones

%_N_MAIN_MPF

| ...

5.8 Cycles technologiques de positionnement de la broche

<pre>IDS=1 EVERY \$A_DBB[0]==1 DO NULL_POS IDS=2 EVERY \$A_DBB[1]==1 DO ZIEL_POS ...</pre>	<pre>; Lorsque \$A_DBB[0] est défini par l'AP, placer en position ini- tiale. ; Lorsque \$A_DBB[1] est défini par l'AP, positionner la broche sur la valeur enregistrée dans \$A_DBW[1].</pre>
--	---

Cycle technologique NULL_POS

%_N_NULL_POS_SPF

<pre>PROC NULL_POS SPOS=0 \$A_DBB[0]=0</pre>	<pre>; Placer l'entraînement dans la position initiale pour le changement d'outil. ; Position initiale exécutée dans la CN.</pre>
---	--

Cycle technologique ZIEL_POS

%_N_ZIEL_POS_SPF

<pre>PROC ZIEL_POS SPOS=IC(\$A_DBW[1]) \$A_DBB[1]=0</pre>	<pre>; Positionner la broche sur la valeur enregistrée dans \$A_DBW[1] par l'AP, cote relative. ; Positionnement cible exécuté dans la CN.</pre>
--	---

5.9 Actions synchrones dans le domaine WZW/BAZ

Introduction

La figure ci-dessous montre le déroulement schématisé du cycle de changement d'outil.

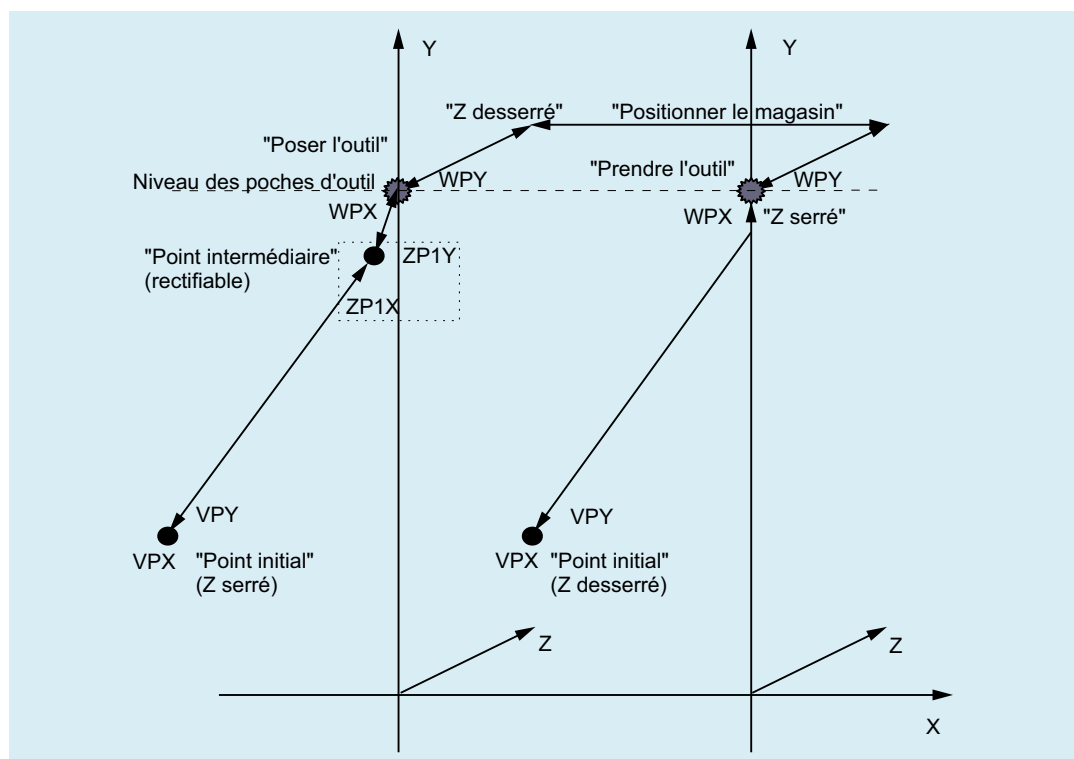


Figure 5-6 Déroulement schématisé du cycle de changement d'outil

Diagramme séquentiel

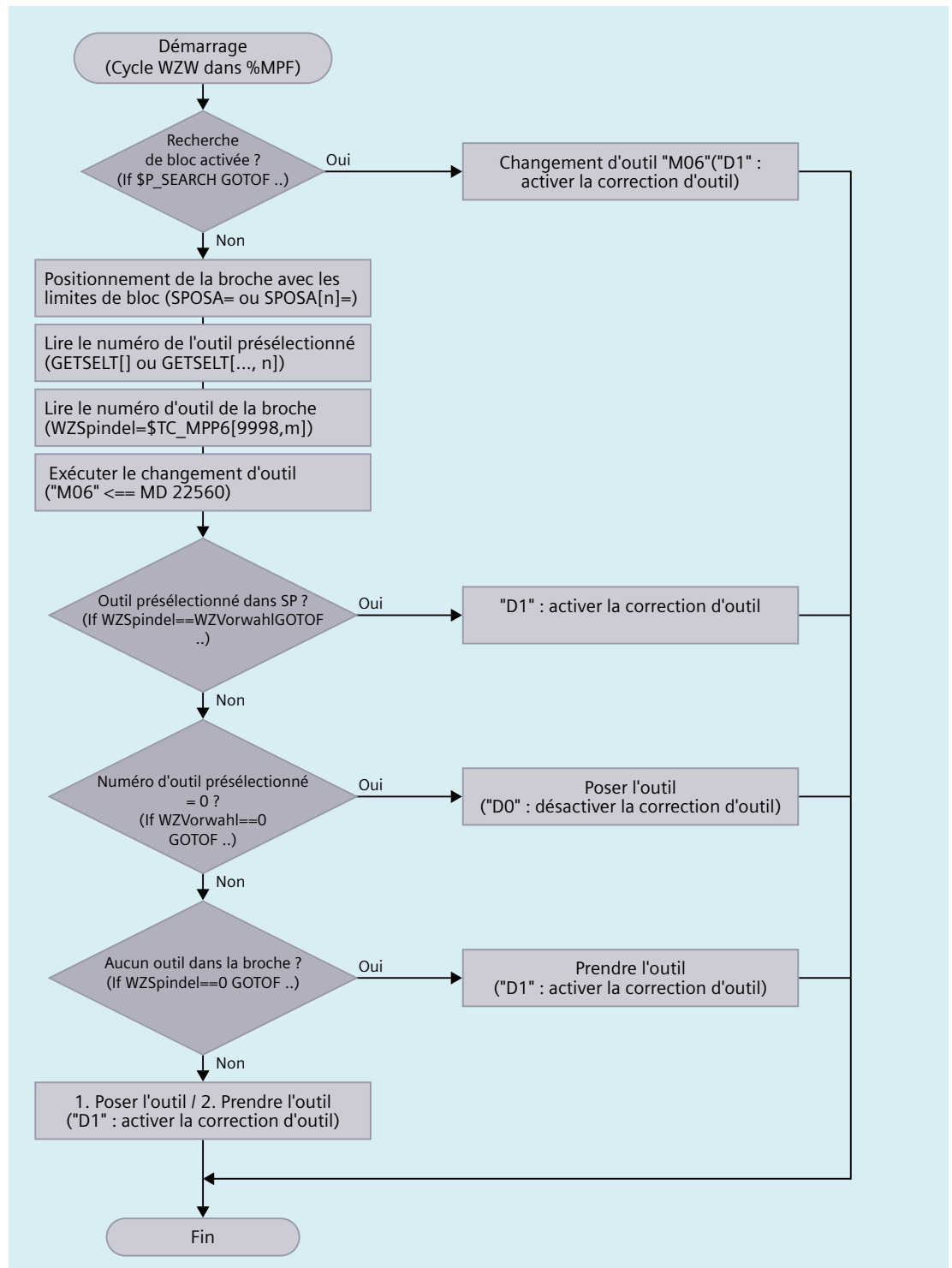


Figure 5-7 Diagramme séquentiel du cycle de changement d'outil

5.9 Actions synchrones dans le domaine WZW/BAZ

Programme CN	Commentaire
<pre> %_N_WZW_SPF ; \$PATH=/_N_SPF_DIR N10 DEF INT WZVorwahl,WZSpindel N15 WHEN \$AC_PATHN<10 DO \$AC_MARKER[0]=0 \$AC_MARKER[1]=0 \$AC_MARKER[2]=0 N20 ID=3 WHENEVER \$A_IN[9]==TRUE DO \$AC_MARKER[1]=1 N25 ID=4 WHENEVER \$A_IN[10]==TRUE DO \$AC_MARKER[2]=1 N30 IF \$P_SEARCH GOTOF wzw_vorlauf N35 SPOSA=0 DO N40 GETSELT(WZVorwahl) N45 WZSpindel=\$TC_MPP6[9998,1] N50 M06 N55 IF WZSpindel==WZVorwahl GOTOF wz_in_spindel IF WZVor- wahl==0 GOTOF ablegen1 IF WZSpindel==0 GOTOF holen1 ;***Prendre et poser l'outil*** ablegen1holen1: N65 WHENEVER \$AA_VACTM[C2]<>0 DO \$AC_MARKER[1]=1 N70 G01 G40 G53 G64 G90 X=Magazin1VPX Y=Magazin1VPY Z=Maga- zin1ZGespannt F70000 M=QU(120) M=QU(123) M=QU(9) N75 WHENEVER \$AA_STAT[S1]<>4 DO \$AC_OVR=0 N80 WHENEVER \$AA_VACTM[C2]<>0 DO \$AC_MARKER[1]=1 N85 WHENEVER \$AC_MARKER[1]==0 DO \$AC_OVR=0 N90 WHENEVER \$AA_STAT[C2]<>4 DO \$AC_OVR=0 N95 WHENEVER \$AA_DTEB[C2]>0 DO \$AC_OVR=0 N100 G53 G64 X=Magazin1ZP1X Y=Magazin1ZP1Y F60000 N105 G53 G64 X=Magazin1WPX Y=Magazin1WPY F60000 N110 M20 N115 G53 G64 Z=MR_Magazin1ZGeloest F40000 N120 WHENEVER \$AA_VACTM[C2]<>0 DO \$AC_MARKER[2]=1; N125 WHENEVER \$AC_MARKER[2]==0 DO \$AC_OVR=0 N130 WHENEVER \$AA_STAT[C2]<>4 DO \$AC_OVR=0 N135 WHENEVER \$AA_DTEB[C2]>0 DO \$AC_OVR=0 N140 G53 G64 Z=Magazin1ZGespannt F40000 N145 M18 N150 WHEN \$AC_PATHN<10 DO M=QU(150) M=QU(121) N155 G53 G64 X=Magazin1VPX Y=Magazin1VPY F60000 D1 M17 </pre>	<pre> ;Marqueur sur = 1 lorsque AxeMag a été déplacé ;Marqueur sur = 1 lorsque AxeMag a été déplacé ;Recherche de bloc activée ? -> ;Lire le numéro de l'outil présélec- tionné ;Lire l'outil dans la broche ;Lorsqu'AxeMag se déplace marqueur = 1. ;Broche en position ;Interroger AxeMag se déplace ;Correction=0 lorsque l'axe n'a pas été déplacé ;Correction=0 lorsque l'AxeMag pas en pos précis ;Correction=0 lorsque la distance res- tant à parcourir AxeMag > 0 ; Desserrer l'outil ;Serrer l'outil ;Condition toujours remplie </pre>

5.9 Actions synchrones dans le domaine WZW/BAZ

Programme CN	Commentaire
;***Poser l'outil***	
ablegen1:	
N160 WHENEVER \$AA_VACTM[C2]<>0 DO \$AC_MARKER[1]=1	
N165 G01 G40 G53 G64 G90 X=Magazin1VPX Y=Magazin1VPY Z=Maga-	
zin1ZGespannt F70000 M=QU(120) M=QU(123) M=QU(9)	
N170 WHENEVER \$AA_STAT[S1]<>4 DO \$AC_OVR=0	
N175 WHENEVER \$AA_VACTM[C2]<>0 DO \$AC_MARKER[1]=1	
N180 WHENEVER \$AC_MARKER[1]==0 DO \$AC_OVR=0	
N185 WHENEVER \$AA_STAT[C2]<>4 DO \$AC_OVR=0	
N190 WHENEVER \$AA_DTEB[C2]>0 DO \$AC_OVR=0	
N195 G53 G64 X=Magazin1ZPlX Y=Magazin1ZPlY F60000	
N200 G53 G64 X=Magazin1WPX Y=Magazin1WPY F60000	
N205 M20	;Desserrer l'outil
N210 G53 G64 Z=Magazin1ZGeloest F40000	
N215 G53 G64 X=Magazin1VPX Y=Magazin1VPY F60000 M=QU(150)	
M=QU(121) D0 M17	
;***Prendre l'outil***	
holen1:	
N220 WHENEVER \$AA_VACTM[C2]<>0 DO \$AC_MARKER[2]=1	
N225 G01 G40 G53 G64 G90 X=Magazin1VPX Y=Magazin1VPY Z=Maga-	
zin1ZGeloest F70000 M=QU(120) M=QU(123) M=QU(9)	
N230 G53 G64 X=Magazin1WPX Y=Magazin1WPY F60000	
N235 WHENEVER \$AA_STAT[S1]<>4 DO \$AC_OVR=0	
N240 WHENEVER \$AA_VACTM[C2]<>0 DO \$AC_MARKER[2]=1	
N245 WHENEVER \$AC_MARKER[2]==0 DO \$AC_OVR=0	
N250 WHENEVER \$AA_STAT[C2]<>4 DO \$AC_OVR=0	
N255 WHENEVER \$AA_DTEB[C2]>0 DO \$AC_OVR=0	
N260 G53 G64 Z=Magazin1ZGespannt F40000	
N265 M18	;Serrer l'outil
N270 G53 G64 X=Magazin1VPX Y=Magazin1VPY F60000 M=QU(150)	
M=QU(121) D1 M17	
;***Outil dans la broche***	
wz_in_spindel:	
N275 M=QU(121) D1 M17	
;***Recherche de bloc***	
wzw_vorlauf:	
N280 STOPRE	
N285 D0	
N290 M06	
N295 D1 M17	

Annexes

A.1 Liste des abréviations

A	
A	Sortie
AFIS	Automatic Filter Switch: Commutation automatique du filtre
ASCII	American Standard Code for Information Interchange: code standard américain pour l'échange d'information
ASIC	Application Specific Integrated Circuit: circuit utilisateur
ASUP	Asynchrone Unterprogramm : sous-programme asynchrone
AUTO	Mode de fonctionnement "Automatic"
AUXFU	Auxiliary Function: fonction auxiliaire
AWL	Liste d'instructions

B	
BAG	Groupe à mode de fonctionnement commun
BCD	Binary Coded Decimals: décimales codées en binaire
BICO	Binector Connector
BIN	Binary Files: fichiers binaires
BKS	Système de coordonnées de base
BM	Message de fonctionnement
BO	Binector Output
BTSS	Operator Panel Interface : interface du pupitre opérateur

C	
CLC	Régulation de distance
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung
COM	Communication
CP	Communication Processor
CPU	Central Processing Unit: unité centrale de traitement
CST	Configured Stop: Arrêt configuré

D	
DDS	Drive Data Set: jeu de paramètres d'entraînement
DIR	Directory: répertoire
DO	Drive Object
DRF	Differential Resolver Function: fonction de résolveur différentiel (manivelle électronique)

A.1 Liste des abréviations

D	
DRY	Dry Run: avance de marche d'essai
DWORD	Double mot (soit 32 bits)

E	
E	Entrée
EES	Execution from External Storage
E/A	Entrée/sortie
ESR	Arrêt étendu et retrait
ETC	Touche ETC ">" ; permet d'étendre la barre de touches logicielles dans le même menu

F	
FDD	Feed Disable: Blocage de l'avance
FdStop	Feed Stop: Arrêt avance
FIFO	First In First Out: premier entré, premier sorti ; mémoire fonctionnant sans adressage et dont les données sont lues dans l'ordre de leur stockage
FM	Message d'erreur
FUP	Logigramme (méthode de programmation pour AP)
FW	Firmware

G	
GEO	Géométrie, par exemple axe géométrique
GP	Programme de base (AP)
GUD	Global User Data: données utilisateur globales

H	
HEX	Abréviation pour nombre hexadécimal
HiFu	Fonction auxiliaire
HMI	Human Machine Interface: interface utilisateur SINUMERIK
HSA	Entraînement de la broche principale
HT	Handheld Terminal
HW	Hardware : matériel

I	
IBN	Mise en service
INC	Increment: déplacement en manuel incrémental
INI	Initializing Data: données d'initialisation
IPO	Interpolator

J	
JOG	Jogging : mode réglage

K	
KOP	Schéma à contacts (méthode de programmation pour AP)

L	
LED	Light Emitting Diode: diode électroluminescente
LMS	Lagemesssystem : système de mesure de position
LR	Lageregler : régulateur de position

M	
Main	Main program: programme principal (OB1, AP)
MCP	Machine Control Panel: tableau de commande machine
MD	Paramètre(s) machine
MDA	Manual Data Automatic: saisie manuelle des données
MDS	Motor Data Set: jeu de paramètres moteur
MELDW	Mot de signalisation
MKS	Système de coordonnées machine
MM	Motor Module
MPF	Main Program File: programme principal (CN)
MPI	Multi Point Interface
MSTT	Tableau de commande machine

N	
NC	Numerical Control: commande numérique avec traitement de blocs, plage de déplacement, etc.
NCU	Numerical Control Unit: unité matérielle de la CN
NCK	Numerical Control Kernel
NCSD	NC Start Disable
NST	Signal d'interface
NV	Décalage d'origine
NX	Numerical Extension: module d'extension d'axe

O	
OB	Bloc d'organisation dans l'AP
OEM	Original Equipment Manufacturer
OP	Operation Panel: pupitre opérateur

A.1 Liste des abréviations

P	
PCU	PC Unit: boîtier PC (unité de traitement)
PG	Console de programmation
PI	Instance de programme
PLC	Programmable Logic Control: programmable
PN	PROFINET
PO	Power On
POS	Position/Positionnement
PPO	Parameter Prozessdaten Objekt : télégramme de données cyclique lors de la transmission avec PROFIBUS-DP et le profil "Entraînements à vitesse variable"
PPU	Panel Processing Unit (matériel central d'une commande CNC à pupitre opérateur, p. ex. SINUMERIK 828D)
PROFIBUS	Process Field Bus: bus de données série
PRT	Test du programme
PTP	Point to Point: point à point
PZD	Prozessdaten : partie données process d'un PPO

R	
REF	Fonction accostage de point de référence
REPOS	Fonction repositionnement
RESU	Retrace support
RID	Read In Disable
RP	Paramètre R : paramètre de calcul, variable utilisateur prédéfinie

S	
SA	Action synchrone
SBL	Single Block: bloc par bloc
SBT	Safe Brake Test
SCC	Safety Control Channel
SCL	Structured Control Language
SD	Settingdatum ou données de réglage
SDI	Safe Direction
SERUPRO	Search-Run by Program Test: Recherche en mode test du programme
SIC	Safety Info Channel
SKP	Skip: fonction de masquage d'un bloc de programme pièce
SLP	Safe Limited Position
SLS	Safely Limited Speed
SMI	Sensor Module Integrated
SOS	Safe Operating Stop
SPF	Sub Program File: fichier de sous-programme (CN)
SS1	Safe Stop 1
SS2	Safe Stop 2

S	
STO	Safe Torque Off
STW	Steuerwort : mot de commande
SUG	Vitesse circonférentielle de meule
SW	Logiciel

T	
TCU	Thin Client Unit
TIA	Totally Integrated Automation
TM	Terminal Module (SINAMICS)
TO	Tool Offset: correction d'outil
TOA	Tool Offset Active: identificateur (type de fichier) pour corrections d'outil
TOFF	Correction en ligne de la longueur d'outil
TRANSMIT	Transform Milling Into Turning: transformation des coordonnées pour des opérations de fraisage sur un tour

U	
UP	Sous-programme
USB	Universal Serial Bus

V	
VDI	Interface de communication interne entre CN et AP

W	
WKS	Système de coordonnées pièce
WPD	Work Piece Directory: répertoire pièce
WZ	Outil
WZV	Gestion des outils

Z	
ZSW	Zustandswort : mot d'état (de l'entraînement)

Index

\$

\$A_INA, 78
\$AA_AXCHANGE_STAT, 96
\$AA_AXCHANGE_TYP, 96
\$AA_JERK_COUNT, 48
\$AA_JERK_TIME, 48
\$AA_JERK_TOT, 48
\$AA_MEAAC, 118
\$AA_OFF, 51
\$AA_OFF_LIMIT, 52
\$AA_OVR, 42
\$AA_PLC_OVR, 42
\$AA_TOFF, 54
\$AA_TOFF_VAL, 55
\$AA_TOTAL_OVR, 43
\$AA_TRAVEL_COUNT, 48
\$AA_TRAVEL_COUNT_HS, 48
\$AA_TRAVEL_DIST, 48
\$AA_TRAVEL_DIST_HS, 48
\$AA_TRAVEL_TIME, 48
\$AA_TRAVEL_TIME_HS, 48
\$AC_BLOCKTYPE, 58
\$AC_BLOCKTYPEINFO, 58
\$AC_DTEB, 84
\$AC_FCT0, 49
\$AC_FCT1, 49
\$AC_FCT2, 49
\$AC_FCT3, 49
\$AC_FCTLL, 49
\$AC_FCTUL, 49
\$AC_FIFO, 36
\$AC_MARKER, 30
\$AC_OVR, 41
\$AC_PARAM, 31
\$AC_PLC_OVR, 42
\$AC_SPLITBLOCK, 60
\$AC_SYNA_STATE, 65
\$AC_SYNC_ACT_LOAD, 44
\$AC_SYNC_AVERAGE_LOAD, 44
\$AC_SYNC_MAX_LOAD, 44
\$AC_TANEB, 41
\$AC_TIMER, 34
\$AC_TOTAL_OVR, 43
\$AN_IPO_ACT_LOAD, 43
\$AN_IPO_LOAD_LIMIT, 45
\$AN_IPO_LOAD_PERCENT, 43
\$AN_IPO_MAX_LOAD, 43

\$AN_IPO_MIN_LOAD, 43
\$AN_SERVO_ACT_LOAD, 44
\$AN_SERVO_MAX_LOAD, 44
\$AN_SERVO_MIN_LOAD, 44
\$AN_SYNC_ACT_LOAD, 44
\$AN_SYNC_MAX_LOAD, 44
\$AN_SYNC_TO_IPO, 44
\$P_TECCYCLE, 128
\$SA_WORKAREA_MINUS_ENABLE, 46
\$SA_WORKAREA_PLUS_ENABLE, 46
\$SN_SW_CAM_MINUS_POS_TAB_1, 46
\$SN_SW_CAM_MINUS_POS_TAB_2, 46
\$SN_SW_CAM_MINUS_TIME_TAB_1, 47
\$SN_SW_CAM_MINUS_TIME_TAB_2, 47
\$SN_SW_CAM_PLUS_POS_TAB_1, 47
\$SN_SW_CAM_PLUS_POS_TAB_2, 47
\$SN_SW_CAM_PLUS_TIME_TAB_1, 47
\$SN_SW_CAM_PLUS_TIME_TAB_2, 47

A

Action synchrone à effet non modal, 20
Action synchrone modale, 20
Action synchrone statique, 20
Actions synchrones
 Adaptation additive via SYNFACT, 76
 Exemple : Presses, couplages d'axes, 150
 Exemple : Régulation AC, 143
 Exemple : Régulation de l'avance tangentielle, 144
 Exemple : Régulation par correction
 dynamique, 145
Arrêt CN, 134
AXTOCHAN (SA), 94

C

CANCEL, 131
CANCELSUB, 124
Combinaisons booléennes, 23
CP..., 113
CTAB..., 113
Cycle technologique, 24
Cycles technologiques, 125

D

DELDTG, 85
DO, 24

E

ELSE (SA), 24
EVERY, 21

F

FA (SA), 93
FOC, 120
FOCOF, 120
FOCON, 120
Fonctions G
 Action, 24
 Condition, 22
FROM, 21
FTOC, 81
FXS, 120
FXST, 120
FXSW, 120

G

G70 (SA), 90
G700 (SA), 90
G71 (SA), 90
G710 (SA), 90
GET (SA), 94
GUD (SA), 66

I

ICYCOF, 126
ICYCON, 126
ID, 20
IDS, 20
Interruption du niveau de sous-programme
 CANCELSUB, 124
Interruption du niveau de sous-programme actuel
 CANCELSUB, 124

L

LEAD..., 113
LOCK, 131

M

M, 100

MD10070, 133
MD10722, 96
MD11110, 74
MD11510, 44
MD18660, 66
MD18661, 66
MD18662, 66
MD18663, 66
MD18664, 66
MD18665, 66
MD20110, 57, 135, 136
MD21190, 55
MD21194, 54
MD21196, 54
MD22200, 74
MD22210, 74
MD22230, 74
MD28250, 132
MD28252, 133
MD28254, 31
MD28255, 32
MD28256, 30
MD28257, 31
MD28258, 34
MD28260, 39
MD28262, 39
MD28264, 39
MD28266, 39
MD30450, 90
MD30460, 107
MD32060, 94
MD32070, 51
MD32074, 88
MD32420, 51
MD32430, 52
MD35040, 136
MD36750, 52, 143
MEAC (SA), 117
MEAWA (SA), 117
MOV, 93

N

Numéro d'identification, 21

O

Ordre d'exécution, 21

P

PM28050, 32
POS (SA), 86
POSRANGE, 92
Power On, 133
PRESETON (SA), 101
PRESETONS (SA), 107

R

RDISABLE, 83
Régulation AC, 143
 Exemple, 144
RELEASE (SA), 94
REPOS, 138
Reset CN, 134

S

S (SA), 100
SD42122, 41
SD43300, 94
SD43350, 52, 143
SPOS (SA), 100
STOPRE (SA), 86
STOPREOF, 84
SYNFCT, 75
 Exemples, 143

T

TRAIL..., 113

U

UNLOCK, 131

W

WHEN, 21
WHENEVER, 21

