

Configuration, fichier ini

REVISION HISTORY			
-------------------------	--	--	--

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Fichiers utilisés pour la configuration	1
2	Organisation du fichier INI	2
2.1	Commentaires	2
2.2	Sections	2
2.3	Variables	3
3	Définition des variables du fichier INI	4
3.1	Section [EMC]	4
3.2	Section [DISPLAY]	4
3.3	Section [EMCMOT]	5
3.4	Section [TASK]	6
3.5	Section [HAL]	6
3.6	Section [TRAJ]	6
3.7	Section [AXIS_<num>]	7
3.7.1	Variables relatives aux prises d'origines	9
3.7.2	Variables relatives aux servomoteurs	9
3.7.3	Variables relatives aux moteurs pas à pas	11
3.8	Section [EMCIO]	11
4	Index	13

Chapter 1

Fichiers utilisés pour la configuration

EMC est entièrement configuré avec des fichiers textes classiques. Tous ces fichiers peuvent être lus et modifiés dans n'importe quel éditeur de texte disponible dans toute distribution Linux¹. Soyez prudent lorsque vous modifierez ces fichiers, certaines erreurs pourraient empêcher le démarrage d'EMC. Ces fichiers sont lus à chaque fois que le logiciel démarre. Certains d'entre eux sont lus de nombreuses fois pendant l'exécution d'CNC.

Les fichiers de configuration inclus:

INI

Le fichier ini écrase les valeurs par défaut compilées dans le code d'EMC. Il contient également des sections qui sont lues directement par le HAL (Hardware Abstraction Layer, couche d'abstraction matérielle).

HAL

Les fichiers hal installent les modules de process, ils créent les liens entre les signaux d'EMC et les broches spécifiques du matériel.

VAR

Ce fichier contient une suite de numéros de variables. Ces variables contiennent les paramètres qui seront utilisés par l'interpréteur. Ces valeurs sont enregistrées d'une exécution à l'autre.

TBL

Ce fichier contient les informations relatives aux outils.

NML

Ce fichier configure les canaux de communication utilisés par EMC. Il est normalement réglé pour lancer toutes les communications avec un seul ordinateur, peut être modifié pour communiquer entre plusieurs ordinateurs.

emcrc::

```
(((.emcrc))) Ce fichier enregistre des informations spécifiques à
l'utilisateur, il a été créé pour enregistrer le nom du répertoire
lorsque l'utilisateur choisit sa première configuration
d'EMC.footnote:[Habituellement, ce fichier est dans le répertoire home
de
l'utilisateur (ex: /home/user/ )
]
```

Les éléments avec le repère (**hal**) sont utilisés seulement pour les fichiers de HAL en exemples. C'est une bonne convention. D'autres éléments sont utilisés directement par EMC et doivent toujours avoir la section et le nom donné à l'item.

¹Ne confondez pas un éditeur de texte et un traitement de texte. Un éditeur de texte comme gedit ou kwrite produisent des fichiers uniquement en texte. Les lignes de textes sont séparées les unes des autres. Un traitement de texte comme Open Office produit des fichiers avec des paragraphes, des mises en formes des mots. Ils ajoutent des codes de contrôles, des polices de formes et de tailles variées etc. Un éditeur de texte n'a rien de tout cela.

Chapter 2

Organisation du fichier INI

Un fichier INI typique suit une organisation simple;

- commentaires.
- sections,
- variables.

Chacun de ces éléments est séparé, sur une seule ligne. Chaque fin de ligne ou retour chariot crée un nouvel élément.

2.1 Commentaires

Une ligne de commentaires débute avec un `;` ou un `#`. Si le logiciel qui analyse le fichier ini rencontre l'un ou l'autre de ces caractères, le reste de la ligne est ignorée. Les commentaires peuvent être utilisés pour décrire ce que font les éléments du fichier INI.

```
; Ceci est le fichier de configuration de ma petite fraiseuse.
```

Des commentaires peuvent également être utilisés pour choisir entre plusieurs valeurs d'une seule variable.

```
# DISPLAY = tkemc
```

Dans cette liste, la variable `DISPLAY` est positionnée sur `axis` puisque toutes les autres sont commentées. Si quelqu'un édite une liste comme celle-ci et par erreur, décommente deux lignes, c'est la première rencontrée qui sera utilisée.

Notez que dans une ligne de variables, les caractères ``#`` et ``;`` n'indiquent pas un commentaire.

```
INCORRECT = value      # and a comment
```

2.2 Sections

Les différentes parties d'un fichier `.ini` sont regroupées dans des sections. Une section commence par son nom en majuscules entre crochets `[UNE_SECTION]`. L'ordre des sections est sans importance. Les sections suivantes sont utilisées par `emc`:

- `[EMC]` informations générales (`[?]`)
 - `[DISPLAY]` sélection du type d'interface graphique (`[?]`)
-

- [RS274NGC] ajustements utilisés par l'interpréteur de g-code
- [EMCMOT] Réglages utilisés par le contrôleur de mouvements temps réel ([?])
- [HAL] spécifications des fichiers .hal ([?])
- [TASK] Réglages utilisés par le contrôleur de tâche ([?])
- [TRAJ] Réglages additionnels utilisés par le contrôleur de mouvements temps réel ([?])
- [AXIS_0] ... [AXIS_n] Groupes de variables pour AXIS ([?])
- [EMCIO] Réglages utilisés par le contrôleur d'entrées/sorties ([?])
- [HALUI] Commandes MDI utilisées par HALUI. Voir le chapitre sur HALUI pour plus d'informations ([?])

2.3 Variables

Une ligne de variables est composée d'un nom de variable, du signe égal (=) et d'une valeur. Tout, du premier caractère non blanc qui suit le signe = jusqu'à la fin de la ligne, est passé comme valeur à la variable. Vous pouvez donc intercaler des espaces entre les symboles si besoin. Un nom de variable est souvent appelé un mot clé.

Les paragraphes suivants détaillent chaque section du fichier de configuration, en utilisant des exemples de variables dans les lignes de configuration.

Certaines de ces variables sont utilisées par EMC. Elles doivent toujours utiliser le nom de section et le nom de variable dans leur appellation. D'autres variables ne sont utilisées que par HAL. Les noms des sections et les noms des variables indiquées sont celles qui sont utilisées dans les exemples de fichiers de configuration.

Chapter 3

Définition des variables du fichier INI

3.1 Section [EMC]]

VERSION = \$Revision: 1.3 \$

Le numéro de version du fichier INI. La valeur indiquée ici semble étrange, car elle est automatiquement mise à jour lors de l'utilisation du système de contrôle de révision. C'est une bonne idée de changer ce numéro à chaque fois que vous modifiez votre fichier. Si vous voulez le modifier manuellement, il suffit de changer le numéro sans toucher au reste.

MACHINE = ma machine

C'est le nom du contrôleur, qui est imprimé dans le haut de la plupart des fenêtres. Vous pouvez insérer ce que vous voulez ici tant que ça reste sur une seule ligne.

RS274NGC_STARTUP_CODE = G21 G90

Une chaîne de codes NC qui sera utilisée pour initialiser l'interpréteur. Elle ne se substitue pas à la spécification des gcodes modaux du début de chaque fichier ngc. Les codes modaux des machines diffèrent, ils pourraient être modifiés par les gcodes interprétés plus tôt dans la session.

3.2 Section [DISPLAY]]

Les différentes interfaces du programme utilisent différentes options. Toutes les options ne sont pas supportées par toutes les interfaces.

DISPLAY = tkemc Le nom de l'interface utilisateur à utiliser. Les options disponibles sont les suivantes:

- axis
- keystick
- mini
- tkemc
- xemc

POSITION_OFFSET = RELATIVE

Le système de coordonnées (RELATIVE ou MACHINE) à utiliser au démarrage de l'interface utilisateur. Le système de coordonnées RELATIVE reflète le G92 et le décalage d'origine G5x actuellement actifs.

POSITION_FEEDBACK = ACTUAL

Valeur de la position (COMMANDED ou ACTUAL) à afficher au démarrage de l'interface utilisateur. La position COMMANDED est la position exacte requise par emc. La position ACTUAL est la position retournée par l'électronique des moteurs.

MAX_FEED_OVERRIDE = 1.2

La correction de vitesse maximum que l'opérateur peut utiliser. 1.2 signifie 120% de la vitesse programmée.

MIN_SPINDLE_OVERRIDE = 0.5

Correction de vitesse minimum de broche que l'opérateur pourra utiliser. 0.5 signifie 50% de la vitesse de broche programmée. (utile si il est dangereux de démarrer un programme avec une vitesse de broche trop basse).

MAX_SPINDLE_OVERRIDE = 1.0

Correction de vitesse maximum de broche que l'opérateur pourra utiliser. 1.0 signifie 100% de la vitesse de broche programmée.

DEFAULT_LINEAR_VELOCITY = .25

Vitesse minimum par défaut pour les jogs linéaires, en unités machine par seconde. Seulement utilisé dans l'interface AXIS.

MAX_LINEAR_VELOCITY = 1.0

Vitesse maximum par défaut pour les jogs linéaires, en unités machine par seconde. Seulement utilisé dans l'interface AXIS.

DEFAULT_ANGULAR_VELOCITY = .25

Vitesse minimum par défaut pour les jogs angulaires, en unités machine par seconde. Seulement utilisé dans l'interface AXIS.

MAX_ANGULAR_VELOCITY = 1.0

Vitesse maximum par défaut pour les jogs angulaires, en unités machine par seconde. Seulement utilisé dans l'interface AXIS.

PROGRAM_PREFIX = ~/emc2/nc_files

Répertoire par défaut des fichiers de g-codes et emplacement des M-codes définis par l'utilisateur.

INCREMENTS = 1 mm, .5 mm, ...

Définit les incréments disponibles pour le jog incremental. Voir la section [?] pour plus d'informations. Seulement utilisé dans l'interface AXIS.

INTRO_GRAPHIC = emc2.gif

L'image affichée sur l'écran d'accueil.

INTRO_TIME = 5

Durée d'affichage de l'écran d'accueil.

OPEN_FILE = /full/path/to/file.ngc

Le fichier NC à utiliser au démarrage d'AXIS.

3.3 Section [EMCMOT]

BASE_PERIOD = 50000

(hal) "Période de base" des tâches, exprimée en nanosecondes. C'est la plus rapide des horloges de la machine.

Avec un système à servomoteurs, il n'y a généralement pas de raison pour que **BASE_PERIOD** soit plus petite que **SERVO_PERIOD**.

Sur une machine de type "step&direction" avec génération logicielle des impulsions de pas, c'est **BASE_PERIOD** qui détermine le nombre maximum de pas par seconde. Si de longues impulsions de pas ou de longs espaces entre les impulsions ne sont pas requis par l'électronique, la fréquence maximum absolue est de un pas par **BASE_PERIOD**. Ainsi, la **BASE_PERIOD** utilisée ici donnera une fréquence de pas maximum absolue de 20000 pas par seconde. 50000ns est une valeur assez large. La plus petite valeur utilisable est liée au résultat du test de latence ([?]), à la longueur des impulsions de pas nécessaire et à la vitesse du μP .

Choisir une **BASE_PERIOD** trop basse peut amener à des messages "Unexpected realtime delay", des bloquages ou des reboots spontanés.

SERVO_PERIOD = 1000000

(hal) Période de la tâche "Servo", exprimée également en nanosecondes. Cette valeur sera arrondie à un multiple entier de **BASE_PERIOD**. Elle est utilisée aussi sur des systèmes basés sur des moteurs pas à pas

C'est la vitesse avec laquelle la nouvelle position des moteurs est traitée, les erreurs de suivi vérifiées, les valeurs des

sorties PID sont rafraichies etc.

Sur la plupart des systèmes **cette** valeur n'est pas à modifier. Il s'agit du taux de mise à jour du planificateur de mouvement de bas niveau.

TRAJ_PERIOD = 1000000

(hal) Période du planificateur de trajectoire, exprimée en nanosecondes. Cette valeur sera arrondie à un multiple entier de **SERVO_PERIOD**.

Excepté pour les machines avec une cinématique particulière (ex: hexapodes) Il n'y a aucune raison de rendre cette valeur supérieure à **SERVO_PERIOD**.

3.4 Section [TASK]

CYCLE_TIME = 0.001

Période exprimée en secondes, à laquelle EMCTASK va tourner. Ce paramètre affecte l'intervalle de polling lors de l'attente de la fin d'un mouvement, lors de l'exécution d'une pause d'instruction et quand une commande provenant d'une interface utilisateur est acceptée. Il n'est généralement pas nécessaire de modifier cette valeur.

3.5 Section [HAL]

HALFILE = example.hal

Exécute le fichier *example.hal* au démarrage. Si **HALFILE** est spécifié plusieurs fois, les fichiers sont exécutés dans l'ordre de leur apparition dans le fichier ini. Presque toutes les configurations auront au moins un **HALFILE**. Les systèmes à moteurs pas à pas ont généralement deux de ces fichiers, un qui spécifie la configuration générale des moteurs (*core_stepper.hal*) et un qui spécifie le brochage des sorties (*xxx_pinout.hal*)

HAL = command

Exécute *command* comme étant une simple commande hal. Si **HAL** est spécifié plusieurs fois, les commandes sont exécutées dans l'ordre où elles apparaissent dans le fichier ini. Les lignes **HAL** sont exécutées après toutes les lignes **HALFILE**.

SHUTDOWN = shutdown.hal

Exécute le fichier *shutdown.hal* quand emc s'arrête. Selon les pilotes de matériel utilisés, il est ainsi possible de positionner les sorties sur des valeurs définies quand emc s'arrête normalement. Cependant, parce qu'il n'y a aucune garantie que ce fichier sera exécuté (par exemple, dans le cas d'une panne de l'ordinateur), il ne remplace pas une véritable chaîne physique d'arrêt d'urgence ou d'autres logiciels de protection des défauts de fonctionnement.

POSTGUI_HALFILE = example2.hal

(Seulement avec l'interface *AXIS*) Exécute *example2.hal* après que l'interface graphique ait créé ses HAL pins.

3.6 Section [TRAJ]

La section [TRAJ] contient les paramètres généraux du module planificateur de trajectoires d'EMCMOT. Vous n'aurez pas à modifier ces valeurs si vous utilisez EMC avec une machine à trois axes en provenance des USA. Si vous êtes dans une zone métrique, utilisant des éléments matériels métriques, vous pourrez utiliser le fichier *stepper_mm.ini* dans lequel les valeurs sont déjà configurées dans cette unité.

COORDINATES = X Y Z

Les noms des axes à contrôler. X, Y, Z, A, B, C, U, V, et W sont valides. Seuls les axes nommés dans **COORDINATES** seront acceptés dans le g-code. Cela n'a aucun effet sur l'ordonnement des noms d'axes depuis le G-code (X- Y- Z-) jusqu'aux numéros d'articulations. Pour une "cinématique triviale", X est toujours l'articulation 0, A est toujours l'articulation 4, U est toujours l'articulation 7 et ainsi de suite. Il est permis d'écrire les noms d'axe par paire (ex: X Y Y Z pour une machine à portique) mais cela n'a aucun effet.

AXES = 3

Une unité de plus que le plus grand numéro d'articulation du système. Pour une machine XYZ, les articulations sont numérotées 0, 1 et 2. Dans ce cas, les AXES sont 3. Pour un système XYUV utilisant une "cinématique triviale", l'articulation V est numérotée 7 et donc les AXES devraient être 8. Pour une machine à cinématique non triviale (ex: scarakins) ce sera généralement le nombre d'articulations contrôlées.

HOME = 0 0 0

Coordonnées de l'origine machine de chaque axe. De nouveau, pour une machine 4 axes, vous devrez avoir 0 0 0 0. Cette valeur est utilisée uniquement pour les machines à cinématique non triviale. Sur les machines avec cinématique triviale, cette valeur est ignorée.

LINEAR_UNITS=<units>

Le nom des unités utilisées dans le fichier INI. Les choix possibles sont *in*, *inch*, *imperial*, *metric*, *mm*. Cela n'affecte pas les unités linéaires du code NC (pour cela il y a les mots G20 et G21).

ANGULAR_UNITS=<units>

Le nom des unités utilisées dans le fichier INI. Les choix possibles sont *deg*, *degree* (360 pour un cercle), *rad*, *radian* (2pi pour un cercle), *grad*, ou *gon* (400 pour un cercle). Cela n'affecte pas les unités angulaires du code NC. Dans le code RS274NGC, les mots A-, B- et C- sont toujours exprimés en degrés.

DEFAULT_VELOCITY = 0.0167

La vitesse initiale de jog des axes linéaires, en unités par seconde. La valeur indiquée ici correspond à une unité par minute.

DEFAULT_ACCELERATION = 2.0

Dans les machines à cinématique non triviale, l'accélération utilisée pour "teleop" jog (espace cartésien), en unités machine par seconde par seconde.

MAX_VELOCITY = 5.0

Vitesse maximale de déplacement pour les axes, exprimée en unités machine par seconde. La valeur indiquée est égale à 300 unités par minute.

MAX_ACCELERATION = 20.0

Accélération maximale pour les axes, exprimée en unités machine par seconde par seconde.

POSITION_FILE = position.txt

Si réglée à une valeur non vide, les positions des axes (joins) sont enregistrées dans ce fichier. Cela permet donc de redémarrer avec les mêmes coordonnées que lors de l'arrêt, ce qui suppose, que hors puissance, la machine ne fera aucun mouvement pendant tout son arrêt. C'est utile pour les petites machines sans contact d'origine machine. Si vide, les positions ne seront pas enregistrées et commenceront à 0 à chaque fois qu'EMC démarrera.

NO_FORCE_HOMING = 1

EMC oblige implicitement l'utilisateur à référencer la machine par une prise d'origine machine avant de pouvoir lancer un programme ou exécuter une commande dans le MDI, seuls les mouvements de Jog sont autorisés avant les prises d'origines. Mettre NO_FORCE_HOMING = 1 permet à l'utilisateur averti de s'affranchir de cette restriction de sécurité lors de la phase de mise au point de la machine.

Attention : NO_FORCE_HOMING mise à 1 permettra à la machine de franchir les limites logicielles pendant les mouvements ce qui n'est pas souhaitable pour un fonctionnement normal!

3.7 Section [AXIS_<num>]

Les sections [AXIS_0], [AXIS_1], etc. contiennent les paramètres généraux des composants individuels du module de contrôle. La numérotation des sections axis commencent à 0 et vont jusqu'au nombre d'axes spécifié dans la variable [TRAJ] AXES, moins 1.

TYPE = LINEAR

Type des axes, soit LINEAR, soit ANGULAR.

UNITS = inch

Ce réglage écrase celui des variables [TRAJ] UNITS si il est spécifié. (ex: [TRAJ]LINEAR_UNITS si le TYPE de cet axe est LINEAR, [TRAJ]ANGULAR_UNITS si le TYPE de cet axe est ANGULAR)

MAX_VELOCITY = 1.2

Vitesse maximum pour cet axe en unités machine par seconde.

MAX_ACCELERATION = 20.0

Accélération maximum pour cet axe en unités machine par seconde au carré.

BACKLASH = 0.000

Valeur de compensation du jeu en unités machine. Peut être utilisée pour atténuer de petites déficiences du matériel utilisé pour piloter cet axe. Si un backlash est ajouté à un axe et que des moteurs pas à pas sont utilisées, la valeur de STEPGEN_MAXACCEL doit être 1.5 à 2 fois plus grande que celle de MAX_ACCELERATION pour cet axe.

COMP_FILE = file.extension

Fichier dans lequel est enregistrée une structure de compensation spécifique à cet axe. Les valeurs internes sont des triplets représentant les positions suivantes:

1. Positions nominales
2. Positions en marche positive
3. Positions en marche négative.

La position nominale est celle où devrait être le mobile. La position en marche positive signifie, où se trouve le mobile pendant le déplacement dans le sens positif. La position en marche négative signifie, où se trouve le mobile pendant le déplacement dans le sens négatif. Un triplet par ligne. Actuellement la limite d'EMC2 est de 256 triplets par axe. Si COMP_FILE est spécifié, BACKLASH est ignoré. Les valeurs sont en unités machine.

COMP_FILE_TYPE = 1

En spécifiant une valeur non nulle, le format des triplets du fichier COMP_FILE sera différent. Pour COMP_FILE_TYPE = 0, les valeurs des triplets seront: position nominale, position en marche positive, position en marche négative. Pour COMP_FILE_TYPE différent de 0, les valeurs dans COMP_FILE seront: position nominale, écart sens positif, écart sens négatif. Comparées aux valeurs définies au dessus elles correspondent à, nominale, nominale-position en marche positive, nominal-position en marche négative.

Exemple de triplet avec COMP_FILE_TYPE = 0: 1.00 1.01 0.99.

Le même exemple de triplet avec COMP_FILE_TYPE = 1: 1.00 -0.01 0.01

MIN_LIMIT = -1000

Limite minimum des mouvements de cet axe (limite soft), en unités machine. Quand cette limite tend à être dépassée, le contrôleur arrête le mouvement.

MAX_LIMIT = 1000

Limite maximum des mouvements de cet axe (limite soft), en unités machine. Quand cette limite tend à être dépassée, le contrôleur arrête le mouvement.

MIN_FERROR = 0.010

Valeur indiquant, en unités machine, de combien le mobile peut dévier à très petite vitesse de la position commandée. Si MIN_FERROR est plus petit que FERROR, les deux produisent une rampe de points de dérive. Vous pouvez imaginer un graphe sur lequel une dimension représente la vitesse et l'autre, l'erreur tolérée. Quand la vitesse augmente, la quantité d'erreurs de suivi augmente également et tend vers la valeur FERROR.

FERROR = 1.0

FERROR est le maximum d'erreurs de suivi tolérable, en unités machine. Si la différence entre la position commandée et la position retournée excède cette valeur, le contrôleur désactive les calculs des servomoteurs, positionne toutes les sorties à 0.0 et coupe les amplis des moteurs. Si MIN_FERROR est présent dans le fichier .ini, une vitesse proportionnelle aux erreurs de suivi est utilisée. Ici, le maximum d'erreur de suivi est proportionnel à la vitesse, quand FERROR est appliqué à la vitesse rapide définie dans [TRAJ]MAX_VELOCITY et proportionnel aux erreurs de suivi pour les petites vitesses.

L'erreur maximale admissible sera toujours supérieure à MIN_FERROR. Cela permet d'éviter que de petites erreurs de suivi sur les axes stationnaires arrêtent les mouvements de manière imprévue. Des petites erreurs de suivi seront toujours présentes à cause des vibrations, etc. La polarité des valeurs de suivi détermine comment les entrées sont interprétées et comment les résultats sont appliqués aux sorties. Elles peuvent généralement être réglées par tâtonnement car il n'y a que deux possibilités. L'utilitaire de calibration peut être utilisé pour les ajuster interactivement et vérifier les résultats, de sorte que les valeurs puissent être mises dans le fichier INI avec un minimum de difficultés. Cet utilitaire est accessible dans Axis depuis le menu «Machine» puis «Calibration» et dans TkEMC depuis le menu «Réglages» puis «Calibration».

3.7.1 Variables relatives aux prises d'origines

Les paramètres suivants sont relatifs aux prises d'origine, pour plus d'informations, lire le chapitre sur la POM. [?]

HOME_OFFSET = 0.0

Position du contact d'origine machine de l'axe ou impulsion d'index, en unités machine.

HOME_SEARCH_VEL = 0.0

Vitesse du mouvement initial de prise d'origine, en unités machine par seconde. Une valeur de zéro suppose que la position courante est l'origine machine. Si votre machine n'a pas de contact d'origine, laissez cette valeur à zéro.

HOME_LATCH_VEL = 0.0

Vitesse du mouvement de dégagement du contact d'origine, en unités machine par seconde.

HOME_FINAL_VEL = 0.0

Vitesse du mouvement final entre le contact d'origine et la position d'origine, en unités machine par seconde. Si cette variable est laissée à 0 ou absente, la vitesse de déplacement rapide est utilisée. Doit avoir une valeur positive.

HOME_USE_INDEX = NO

Si l'encodeur utilisé pour cet axe fournit une impulsion d'index et qu'elle est gérée par la carte contrôleur, vous pouvez mettre sur Yes. Quand il est sur yes, il aura une incidence sur le type de séquence de prise d'origine utilisé.

HOME_IGNORE_LIMITS = NO

Certaines machines utilisent un seul et même contact comme limite d'axe et origine machine de l'axe. Cette variable devra être positionnée sur yes si c'est le cas de votre machine.

HOME_IS_SHARED = <n>

Si l'entrée du contact d'origine est partagée par plusieurs axes, mettre <n> à 0 pour permettre la POM même si un des contacts partagés est déjà attaqué. Le mettre à 1 pour interdire la prise d'origine dans ce cas.

HOME_SEQUENCE = <n>

Utilisé pour définir l'ordre dans lequel les axes se succéderont lors d'une séquence de "POM générale". <n> commence à 0, aucun numéro ne peut être sauté. Si cette variable est absente ou à -1, la POM de l'axe ne pourra pas être exécutée par la commande "POM générale". La POM de plusieurs axes peut se dérouler simultanément.

3.7.2 Variables relatives aux servomoteurs

Les entrées suivantes concernent les systèmes à servomoteurs, comme la carte du système univstep de Pico Systems.¹ Cette description suppose que les unités en sortie du composant PID sont des Volts.

P = 50

(hal) La composante **p** proportionnelle du gain de l'ampli moteur de cet axe. Cette valeur multiplie l'erreur entre la position commandée et la position actuelle en unités machine, elle entre dans le calcul de la tension appliquée à l'ampli moteur. Les unités du gain **P** sont des Volts sur des unités machine, exemple: $\frac{\text{volt}}{\text{mm}}$ si l'unité machine est le millimètre.

¹Référez vous au "Manuel de l'intégrateur d'EMC2" pour des informations complémentaires sur les systèmes à servomoteurs et leur contrôle en PID.

I = 0

(hal) La composante **i** intégrale du gain de l'ampli moteur de cet axe. Cette valeur multiplie l'erreur cumulative entre la position commandée et la position actuelle en unités machine, elle entre dans le calcul de la tension appliquée à l'ampli moteur. Les unités du gain **I** sont des Volts sur des unités machine par seconde, exemple: $\frac{\text{volt}}{\text{mm.s}}$ si l'unité machine est le millimètre.

D = 0

(hal) La composante **d** érivée du gain de l'ampli moteur de cet axe. Cette valeur multiplie la différence entre l'erreur courante et les précédentes, elle entre dans le calcul de la tension appliquée à l'ampli moteur. Les unités du gain **D** sont des Volts sur des unités machine sur des secondes, exemple: $\frac{\text{volt}}{\text{mm/s}}$ si l'unité machine est le millimètre.

FF0 = 0

(hal) Gain à priori (feedforward) d'ordre 0. Cette valeur est multipliée par la position commandée, elle entre dans le calcul de la tension appliquée à l'ampli moteur. Les unités du gain FF0 sont des Volts sur des unités machine, exemple: $\frac{\text{volt}}{\text{mm}}$ si l'unité machine est le millimètre.

FF1 = 0

(hal) Gain à priori (feedforward) de premier ordre. Cette valeur est multipliée par l'écart de la position commandée par seconde, elle entre dans le calcul de la tension appliquée à l'ampli moteur. Les unités du gain FF1 sont des Volts sur des unités machine par seconde, exemple: $\frac{\text{volt}}{\text{mm.s}}$ si l'unité machine est le millimètre.

FF2 = 0

(hal) Gain à priori (feedforward) de second ordre. Cette valeur est multipliée par l'écart de la position commandée par seconde au carré, elle entre dans le calcul de la tension appliquée à l'ampli moteur. Les unités du gain FF2 sont des Volts sur des unités machine par des secondes au carré, exemple: $\frac{\text{volt}}{\text{mm.s}^2}$ si l'unité machine est le millimètre.

OUTPUT_SCALE = 1.000 , OUTPUT_OFFSET = 0.000

(hal) Ces deux valeurs sont, l'échelle et le facteur d'offset de l'ampli moteur de cet axe. La seconde valeur (offset) est soustraite de la valeur de sortie calculée (en Volts) puis divisée par la première valeur (facteur d'échelle), avant d'être écrite dans le convertisseur D/A. Les unités du facteur d'échelle sont des Volts réels par Volts en sortie de DAC. Les unités de la valeur d'offset sont en Volts. Ces valeurs peuvent être utilisées pour linéariser un DAC.

Plus précisément, quand les sorties sont écrites, EMC converti d'abord les unités quasi-SI des sorties concernées en valeurs brutes, exemple: Volts pour un amplificateur DAC. Cette mise à l'échelle ressemble à cela: $[\text{raw} = \frac{\text{out put} - \text{offset}}{\text{scale}}]$ La valeur d'échelle peut être obtenue par analyse des unités, exemple: les unités sont [unités SI en sortie]/[unités de l'actuateur]. Par exemple, sur une machine sur laquelle une tension de consigne de l'ampli de 1 Volt donne une vitesse de 250 mm/sec :
volt}}]

Notez que les unités d'offset sont en unités machine, exemple: mm/sec et qu'elles sont déjà soustraites depuis la sonde de lecture. La valeur de cet offset est obtenue en prenant la valeur de votre sortie qui donne 0,0 sur la sortie de l'actuateur. Si le DAC est linéarisé, cet offset est normalement de 0.0.

L'échelle et l'offset peuvent être utilisés pour linéariser les DAC, d'où des valeurs qui reflètent les effets combinés du gain de l'ampli, de la non linéarité du DAC, des unités du DAC, etc. Pour ce faire, suivez cette procédure:

1. Construire un tableau de calibrage pour la sortie, piloter le DAC avec la tension souhaitée et mesurer le résultat. Voir le tableau [\[cap:Mesure des tensions de sortie\]](#) pour un exemple de mesures de tension.
2. Effectuer un "least squares" linéaire pour obtenir les coefficients a, b tels que: $\text{meas} = a * \text{raw} + b$
3. Notez que nous voulons des sorties brutes de sorte que nos résultats mesurés soient identiques à la sortie commandée. Ce qui signifie:
4. $\text{cmd} = a * \text{raw} + b$
5. $\text{raw} = (\text{cmd} - b) / a$
6. En conséquence, les coefficients a et b d'ajustement linéaire peuvent être directement utilisés comme valeurs d'échelle et d'offset pour le contrôleur.

MAX_OUTPUT = 10

(hal) La valeur maximale pour la sortie de la compensation PID pouvant être envoyée sur l'ampli moteur, en Volts. La valeur calculée de la sortie sera fixée à cette valeur limite. La limite est appliquée avant la mise à l'échelle de la sortie en unités brutes.

MIN_OUTPUT = -10

(hal) La valeur minimale pour la sortie de la compensation PID pouvant être envoyée sur l'ampli moteur, en Volts. La valeur calculée de la sortie sera fixée à cette valeur limite. La limite est appliquée avant la mise à l'échelle de la sortie en unités brutes.

Mesure des tensions de sortie

Raw (brutes)	Mesurées
-10	-9.93
-9	-8.83
0	-0.03
1	0.96
9	9.87
10	10.87

INPUT_SCALE = 20000

(hal) Spécifie le nombre d'impulsions qui correspond à un mouvement d'une unité machine. Un second chiffre, si spécifié, sera ignoré.

Par exemple, sur un codeur de 2000 impulsions par tour, un réducteur de 10 tours/pouce et des unités demandées en mm, nous avons:

$$\begin{aligned} \text{input_scale} &= 2000 \frac{\text{impulsions}}{\text{tour}} * 10 \frac{\text{tours}}{\text{pouce}} \\ &= 20000 \frac{\text{impulsions}}{\text{pouce}} \end{aligned}$$

3.7.3 Variables relatives aux moteurs pas à pas**SCALE = 4000**

(hal) Spécifie le nombre d'impulsions qui correspond à un mouvement d'une unité machine. Pour les systèmes à moteurs pas à pas, c'est le nombre d'impulsions de pas nécessaires pour avancer d'une unité machine. Pour les systèmes à servo, c'est le nombre d'impulsions de retour signifiant que le mobile a avancé d'une unité machine. Un second chiffre, si spécifié, sera ignoré.

Par exemple, un pas moteur de 1.8 degré, en mode demipas, avec une réduction de 10 tours/pouce et des unités souhaitées en mm, nous avons:

$$\begin{aligned} \text{input_scale} &= \frac{2 \text{ pas}}{1.8 \text{ degre}} * 360 \frac{\text{degres}}{\text{tour}} * 10 \frac{\text{tours}}{\text{pouce}} \\ &= 4000 \frac{\text{pas}}{\text{pouce}} \end{aligned}$$

D'anciens fichiers de configuration .ini et .hal utilisaient INPUT_SCALE pour cette valeur.

STEPGEN_MAXACCEL = 21.0

(hal) Limite d'accélération pour le générateur de pas. Elle doit être 1% à 10% supérieure à celle de l'axe MAX_ACCELERATION. Cette valeur améliore les réglages de la "boucle de position" de stepgen. Si une correction de backlash a été appliquée sur un axe, alors STEPGEN_MAXACCEL doit être 1,5 à 2 fois plus grande que MAX_ACCELERATION.

STEPGEN_MAXVEL = 1.4

(hal) Les anciens fichiers de configuration avaient également une limite de vitesse du générateur de pas. Si spécifiée, elle doit aussi être 1% à 10% supérieure à celle de l'axe MAX_VELOCITY. Des tests ultérieurs ont montré que l'utilisation de STEPGEN_MAXVEL n'améliore pas le réglage de la boucle de position de stepgen.

3.8 Section [EMCIO]**CYCLE_TIME = 0.100**

La période en secondes, à laquelle EMCIO va tourner. La mettre à 0.0 ou à une valeur négative fera qu'EMCIO tournera en permanence. Il est préférable de ne pas modifier cette valeur.

TOOL_TABLE = tool.tbl

Ce fichier contient les informations des outils.

TOOL_CHANGE_POSITION = 0 0 2

Quand trois digits sont utilisés, spécifie la position XYZ ou le mobile sera déplacé pour le changement d'outil. Si six digits sont utilisés, spécifie l'emplacement ou sera envoyé le mobile pour réaliser le changement d'outil sur une machine de type XYZABC et de même, sur une machine de type XYZABCUVW lorsque 9 digits sont utilisés. Les variables relatives à la position du changement d'outil peuvent être combinées, par exemple; en combinant TOOL_CHANGE_POSITION avec TOOL_CHANGE_QUILL_UP il est possible de déplacer d'abord Z puis X et Y.

TOOL_CHANGE_WITH_SPINDLE_ON = 1

Avec cette valeur à 1, la broche reste en marche pendant le changement d'outil. Particulièrement utile sur les tours.

TOOL_CHANGE_QUILL_UP = 1

Avec cette valeur à 1, l'axe Z sera déplacé sur son origine machine avant le changement d'outil. C'est l'équivalent d'un G0 G53 Z0.

TOOL_CHANGE_AT_G30 = 1

Avec cette valeur à 1, le mobile sera envoyé sur un point de référence prédéfini par G30 dans les paramètres 5181-5186. Pour plus de détails sur les paramètres de G30, voir le chapitre relatif au G-code dans le Manuel de l'utilisateur.

Chapter 4

Index

A

ANGULAR UNITS, 7

B

BASE PERIOD, 5

C

codeur, 11

commentaires, 2

D

DISPLAY, 2

F

FERROR, 8

Fichier INI-AXIS, 7

Fichier INI-DISPLAY, 4

Fichier INI-EMCIO, 11

Fichier INI-EMCMOT, 5

Fichier INI-HAL, 6

Fichier INI-TASK, 6

Fichier INI-TRAJ, 6

H

HAL, 1

HOME SEARCH VEL, 9

I

INI, 1

INPUT SCALE, 11

L

LINEAR UNITS, 7

M

MAX ACCELERATION, 7

MAX LIMIT, 8

MAX VELOCITY, 7

MIN FERROR, 8

MIN LIMIT, 8

N

NML, 1

R

RS274NGC STARTUP CODE, 4

S

Sections du fichier INI, 4

SERVO PERIOD, 5

T

TBL, 1

TRAJ PERIOD, 6

U

UNITS, 8

V

VAR, 1