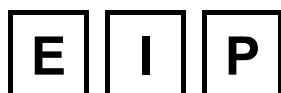


# **UNIPROG+ version 1.10**

**Langage de programmation universel**

Version: **3 mars 2006**



## Table des matières :

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction.....</b>   | <b>5</b>  |
| 1.1      | Utilisation de ce manuel.....                                    | 5         |
| <b>2</b> | <b>Organisation des zones de stockage .....</b>                  | <b>6</b>  |
| 2.1      | Mémoires de la Commande E-100.....                               | 6         |
| <b>3</b> | <b>Géométrie et cinématique.....</b>                             | <b>7</b>  |
| 3.1      | Position de référence .....                                      | 7         |
| 3.2      | Changement de Référentiel.....                                   | 8         |
| 3.3      | Constantes Cinématiques.....                                     | 9         |
| 3.3.1    | Le facteur d'échelle SCALEK.....                                 | 9         |
| 3.3.2    | Fréquence maximale, Paramètre DIV.....                           | 9         |
| 3.3.3    | Constante d'accélération KUP, et de décélération KDN.....        | 10        |
| 3.3.4    | Constante de Vitesse FEEDK.....                                  | 10        |
| 3.3.5    | Course de l'axe, STROKE + / STROKE -.....                        | 11        |
| 3.3.6    | Sur-excitation du moteur : commande BOOST.....                   | 11        |
| 3.3.7    | Mode pour impulsions de commande.....                            | 12        |
| <b>4</b> | <b>Mode opératoire du clavier et fonctions de UNIPROG+.....</b>  | <b>13</b> |
| 4.1      | Mise sous tension de la commande E-100.....                      | 13        |
| 4.2      | Sélection des Menus.....   | 14        |
| 4.3      | Menu "DIVERS".....   | 15        |
| 4.3.1    | 'ACCES' Accès aux protections.....                               | 15        |
| 4.3.2    | 'VER' Version.....   | 15        |
| 4.3.3    | 'CLK' réglage de l'horloge.....                                  | 15        |
| 4.3.4    | 'COUNT' compteur de cycles.....                                  | 15        |
| 4.4      | Menu "CONFIGURATION".....  | 16        |
| 4.4.1    | Configuration du Générateur de Mouvement (Motion GENERator)..... | 16        |
| 4.4.2    | Configuration de la Prise de Référence (REFERence).....          | 16        |
| 4.4.3    | Configuration des Entrées de Contrôle (CONTRoL).....             | 17        |
| 4.5      | Menu "MODE MANUEL".....  | 18        |
| 4.5.1    | Mouvements Manuels, JOG.....                                     | 18        |
| 4.5.2    | Contrôle de fermeture, CLOS.....                                 | 19        |
| 4.5.3    | Utilitaire "I/O" control.....                                    | 20        |
| 4.6      | Menu "PROGRAMMATION".....  | 21        |
| 4.6.1    | Tableau d'outils "TOOL".....                                     | 21        |
| 4.6.2    | Vitesses Présélectionnées (FEED rates).....                      | 22        |
| 4.6.3    | Sauvegarde dans la mémoire FLASH (SAVE).....                     | 22        |
| 4.7      | Manipulation de Fichiers, UTILITAIRES PROG.....                  | 23        |
| 4.7.1    | Répertoire, (DIRectory).....                                     | 23        |
| 4.7.2    | Effacement d'un Fichier (DELeTe).....                            | 23        |
| 4.7.3    | Copie d'un Fichier (COPIE).....                                  | 23        |
| 4.7.4    | Chargement depuis la mémoire FLASH (LOAD).....                   | 24        |
| <b>5</b> | <b>Instructions du langage UNIPROG+ .....</b>                    | <b>25</b> |
| 5.1      | Instructions de Positionnement.....                              | 25        |

|          |  |           |
|----------|--|-----------|
| 5.1.1    | Positionnement absolu avec attente de la fin du mouvement :.....   | 26        |
| 5.1.2    | Positionnement absolu sans attendre la fin du mouvement : .....    | 26        |
| 5.1.3    | Positionnement relatif avec attente de la fin du mouvement : ..... | 26        |
| 5.1.4    | Positionnement relatif sans attendre la fin du mouvement : .....   | 26        |
| 5.1.5    | Arrêt d'un mouvement .....   | 26        |
| 5.1.6    | Définition des outils .....  | 26        |
| 5.2      | Autres instructions Cinématiques .....                             | 27        |
| 5.2.1    | Prise de référence : .....   | 27        |
| 5.2.2    | Test de fermeture : .....  | 27        |
| 5.2.3    | Définition de paramètres .....                                     | 27        |
| 5.2.4    | Cycle de perçage .....   | 27        |
| 5.2.5    | Instruction de taraudage.....                                      | 28        |
| 5.3      | Instructions d'entrées/sorties .....                               | 29        |
| 5.3.1    | Attente sur entrée : .....   | 29        |
| 5.3.2    | Branchement sur entrée : .....                                     | 29        |
| 5.3.3    | Commande des sorties : .....                                       | 29        |
| 5.3.4    | Inversion d'une sortie .....                                       | 31        |
| 5.4      | Instructions d'Affichage.....                                      | 31        |
| 5.4.1    | Affichage direct.....  | 31        |
| 5.4.2    | Restitution de la fenêtre de base.....                             | 31        |
| 5.4.3    | Affichage de texte.....  | 31        |
| 5.5      | Manipulation de Quantités Numériques.....                          | 32        |
| 5.5.1    | Chargement de l'accumulateur.....                                  | 32        |
| 5.5.2    | Dépose de l'accumulateur.....                                      | 32        |
| 5.5.3    | Incrémentation/Décrémentation d'un pointeur : .....                | 32        |
| 5.5.4    | Entrée de nombre.....  | 33        |
| 5.6      | Utilisation des entrées-sorties analogiques .....                  | 33        |
| 5.6.1    | Lecture d'une entrée analogique .....                              | 33        |
| 5.6.2    | Chargement d'une sortie analogique.....                            | 33        |
| 5.6.3    | Vitesse de la broche .....   | 33        |
| 5.6.4    | Vitesse de rotation des moteurs .....                              | 33        |
| 5.7      | Instructions de Contrôle du Programme .....                        | 34        |
| 5.7.1    | Saut inconditionnel : .....  | 34        |
| 5.7.2    | Appel de sous-programme : .....                                    | 34        |
| 5.7.3    | Fin de programme, de sous-programme : .....                        | 34        |
| 5.7.4    | Répétition : .....   | 34        |
| 5.7.5    | Appel de programmes simultanés : .....                             | 35        |
| 5.7.6    | Branchements conditionnels (Test de l'accumulateur): .....         | 35        |
| 5.8      | Instructions de Temporisation.....                                 | 35        |
| 5.9      | Instructions Arithmétiques.....                                    | 36        |
| 5.10     | Instructions relatives à l'horloge interne .....                   | 36        |
| 5.10.1   | Lecture de la date.....  | 36        |
| 5.10.2   | Lecture du jour de la semaine .....                                | 36        |
| 5.10.3   | Lecture de l'heure.....  | 36        |
| 5.11     | NOP et "Pseudo-Instructions" .....                                 | 37        |
| 5.12     | Marqueur de Pause .....  | 37        |
| <b>6</b> | <b>L'éditeur UNIPROG+ .....</b>                                    | <b>38</b> |
| 6.1      | Examen d'un programme.....   | 38        |
| 6.2      | Modification du Contenu d'une ligne.....                           | 38        |

|          |  |           |
|----------|--|-----------|
| 6.3      | Insertion et Suppression d'une ligne.....        | 39        |
| 6.4      | Mise en Place du Marqueur de Pause.....          | 39        |
| <b>7</b> | <b>Exécution des programmes.....</b>             | <b>40</b> |
| 7.1      | Menu "EXECUTION" .....                           | 40        |
| 7.1.1    | Fonctions des touches START, STOP.....           | 40        |
| 7.1.2    | Programmes à exécuter (start VECTors) .....      | 40        |
| 7.1.3    | "MODE", fonctions des modes MOD1, MOD2, SAT..... | 41        |
| 7.1.4    | Utilitaire "TRACE".....                          | 41        |
| 7.1.5    | Utilitaire "I/O" control .....                   | 42        |
| 7.2      | Gestion des fautes .....                         | 42        |
| <b>8</b> | <b>Récapitulation UNIPROG + .....</b>            | <b>43</b> |
| 8.1      | Instructions.....                                | 43        |
| 8.2      | Entrées-Sorties .....                            | 45        |

### Liste des figures :

|            |   |    |
|------------|---|----|
| Figure 3-1 | : Courses et prise de référence .....         | 7  |
| Figure 3-2 | : Les référentiels UNIPROG+ .....             | 9  |
| Figure 3-3 | : Fréquence ou vitesse en fonction de t ..... | 10 |
| Figure 4-1 | : Menu "I/O Control".....                     | 21 |
| Figure 5-1 | : Cycle de perçage.....                       | 28 |

### Liste des tableaux :

|             |                                    |    |
|-------------|------------------------------------|----|
| Tableau 4.1 | : Paramètre 'REF INPUT'.....       | 17 |
| Tableau 5.1 | : Entrées/Sorties UNIPROG+ .....   | 30 |
| Tableau 5.2 | : Adresse des modules IN/OUT ..... | 30 |

# 1 Introduction

La commande de positionnement de la série E-100 est destinée à l'équipement de manipulateurs et de petites machines spéciales et résout les problèmes de cinématique et d'automatisme les plus variés.

- Elle est disponible en version compacte 1 axe pour moteurs pas-à-pas 2 phases ou No-Drive pour toutes sortes de moteurs..
- E.I.P. SA a créé le langage de programmation PINX-E et l'outil de développement APEX afin de répondre de façon efficace aux situations les plus variées et les plus exigeantes. La pratique a montré que nombre de fonctions se retrouvent dans la plupart des applications mais que les séquences de travail sont infiniment variables. D'autre part, la maîtrise du langage PINX-E demande un effort certain et l'utilisation d'un PC comme outil de développement n'est pas souhaitable dans un environnement industriel.
- Ces considérations ont conduit E.I.P. au programme **UNIPROG+** qui offre un ensemble de fonctions utilitaires et la possibilité d'écrire des programmes performants simplement à partir du panneau de la commande.

## 1.1 Utilisation de ce manuel

Ce manuel doit permettre à un utilisateur sans expérience en informatique de maîtriser UNIPROG+ après une lecture attentive. Quelques connaissances des entraînements par moteurs pas-à-pas et particulièrement des systèmes asservis sont nécessaires pour éviter des tâtonnements laborieux. Par contre, il n'est pas utile de connaître la programmation des automates programmables.

- Le lecteur aura intérêt à bien étudier les chapitres 2 et 3 avant d'essayer d'écrire un programme ou d'utiliser les fonctions utilitaires.
- Le chapitre 4 décrit le mode opératoire du clavier à partir de la mise sous tension.

## 2 Organisation des zones de stockage

### 2.1 Mémoires de la Commande E-100

Les programmes écrits par l'utilisateur sont mémorisés dans la mémoire de travail (RAM), sauvegardée par batterie, de la commande E-100. Il est possible de mémoriser les données dans la mémoire de sauvegarde (FLASH) afin de prévenir une défaillance de la pile (voir commandes SAVE et LOAD). Les capacités de ces mémoire sont égales.

A l'intérieur de la zone mémoire, un espace est réservé aux paramètres de configuration (voir paragraphe 4.4). Le reste de la mémoire est utilisable pour des programmes.

L'éditeur UNIPROG travaille par lignes. Une instruction ou une donnée numérique est toujours mémorisée sous forme d'une ligne.

- 1250 lignes sont disponibles.
- 100 fichiers peuvent être créés. Ils sont numérotés de 00 à 99.

### 3 Géométrie et cinématique

Ce chapitre fixe le système de référentiels utilisé par UNIPROG+ et donne les informations utiles sur les générateurs de mouvements de la commande E-100.

#### 3.1 Position de référence

L'entraînement à moteur pas-à-pas ou asservi étant incrémentiel par nature, tout cycle de travail doit débuter par la fixation des positions de référence de l'axe. Deux cas pratiques se présentent :

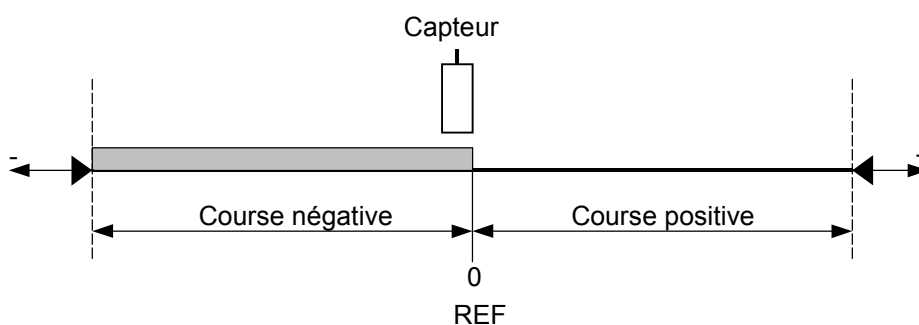
- a) Le référentiel est lié au bâti de la machine, c'est le cas d'une table de palettisation, par exemple.
- b) Le référentiel est fixé par l'opérateur en un point de la course et à un instant choisi par lui, c'est le cas d'un appareil à diviser.

Dans le cas a), l'axe doit être muni d'un contact de référence qui permet la fixation automatique et précise du **référentiel initial**.

La prise de la position de référence peut se faire sur ordre manuel ou être intégrée à un programme d'initialisation.

UNIPROG+ permet de placer le contact de référence n'importe où dans la course.

**Le contact de référence doit être actif en permanence sur l'une des courses afin de signaler à la commande de quel côté se situe la coulisse (voir Figure 3-1)**



**Figure 3-1 : Courses et prise de référence**

UNIPROG+ effectue la prise de référence en trois phases :

- Phase 1 : Cette phase de la prise de référence est exécutée uniquement si le contact est déjà actif : Sortie du contact de référence à une vitesse configurée avec

décélération configurée à la sortie du contact.

Phase 2 : Déplacement vers le contact de référence à une vitesse configurée avec décélération sur contact.

Phase 3 : Déplacement en sens opposé à vitesse réduite (fraction configurée de la vitesse de prise de référence) et arrêt instantané dès que le contact n'est plus actionné.

- La phase 3 donne la précision de la position de référence, variable en fonction de la vitesse de retour configurée. Il peut être intéressant de noter le sens de déplacement de la phase 2 : l'utilisateur sait ainsi dans quel sens sont pris en compte les éventuels jeux et hystérésis de la transmission.
- Une entrée parmi les 8 disponibles sur la commande E-100 fonctionne comme signal de référence.
- Lorsque l'axe n'utilise pas de contact de référence, l'origine sera fixée par l'exécution d'une prise de référence sans aucun mouvement.
- La course (STROKE +, resp. STROKE -) introduite par le menu de configuration délimite la course positive, resp. négative de 0 à STROKE +, resp. STROKE -, dans laquelle se déplacent l'axe.
- Si la course est uniquement positive (STROKE - = 0), seuls les points dont les coordonnées dans le référentiel initial sont positives seront accessibles, voir Figure 3-1. Le sens de prise de référence tient compte du signe de la vitesse de prise de référence introduite par le menu de configuration.
- Lorsqu'une limite de course n'est pas souhaitée sur l'axe, par exemple dans la commande d'un plateau-diviseur, il suffit de configurer une course nulle. Dans ce cas, aucune entrée de référence ne doit être attribuée à cet axe.
- Respecter la condition du §4.4.1 p.15

## 3.2 Changement de Référentiel

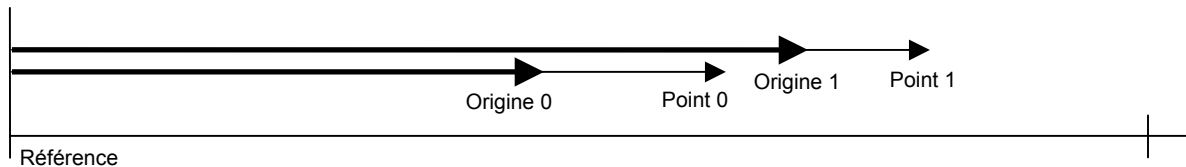
Les mouvements à effectuer sont donnés en valeur "relative" ou en coordonnée "absolue". La notion de mouvement relatif et absolu est applicable tant aux déplacements commandés manuellement dans le menu "JOGGING" qu'aux instructions de mouvement.

Dans un mouvement relatif, le déplacement est relatif au point où se trouve l'axe avant le mouvement. La notion de référentiel est évidemment sans objet.

Dans un mouvement absolu, c'est la valeur des coordonnées du point à atteindre qui est donnée. Le référentiel dans lequel sont mesurées les coordonnées prend alors tout son sens.



Le **référentiel courant** est introduit par une instruction **TOOL**. Le programmeur peut changer le référentiel courant aussi souvent qu'il en éprouve le besoin. Si aucune instruction **TOOL** n'est exécutée, le référentiel courant est confondu avec le référentiel de base. Le changement de référentiel courant est utile dans les situations telles que les machines à outils multiples ou les manipulateurs dans lesquels on distingue un espace de prise et un espace de dépose.



*Figure 3-2 : Les référentiels UNIPROG+*

### 3.3 Constantes Cinématiques

Les constantes discutées dans ce paragraphe seront introduites par le menu de configuration. Ces constantes sont propres à chaque axe.

#### 3.3.1 Le facteur d'échelle SCALEK

Ce facteur permet de programmer et d'afficher les cotes et les déplacements en unités techniques.

Dans le cas des moteurs pas-à-pas, SCALEK est le nombre d'impulsions requis à l'entrée de l'étage de puissance pour produire un déplacement de une unité.

Les étages de puissance E.I.P. pour moteurs 2 phases, opèrent toujours en mode micro-pas avec une division par 8 du pas entier. Avec des moteurs courants (200 pas par tour) il faut donc 1600 impulsions par tour de moteur.

Respecter la condition du §4.4.1 p.15

##### Exemples: coulisse à vis

Moteur 2 phases, 1,8 deg. par pas, transmission par courroie de rapport 1:2, vis au pas de 5 mm, unité: 1 mm:

1600 impulsions pour 1 tour du moteur,

3200 impulsions pour 1 tour de la vis,

Donc  $3200/5 = 640$  impulsions par mm, **SCALEK = 640**.

#### 3.3.2 Fréquence maximale, Paramètre DIV

Lors de l'exécution d'un mouvement, la fréquence de l'axe (ou la vitesse de l'axe) a l'allure reproduite à la Figure 3-3. La grandeur de l'accélération ou de la décélération décroît linéairement en fonction de la vitesse pour prendre en compte la baisse du couple du

moteur aux vitesses élevées. La fréquence maximale doit être déterminée expérimentalement pour chaque axe de façon à conserver une marge de couple suffisante aux plus hautes vitesses.

Dans tous les cas de fonctionnement, la vitesse sera inférieure ou égale à la vitesse maximale résultant de DIV. Lors de l'accélération, le générateur de mouvement tronque la courbe de vitesse à la vitesse palier prescrite. Les pentes d'accélération et de décélération sont conservées.

Respecter la condition du §4.4.1 p.15

Si  $f_{max}$  est la fréquence maximale des impulsions à l'entrée de l'étage de puissance du moteur pas-à-pas, le paramètre DIV est donné par :

$$\text{DIV} = 18290/f_{max} \text{ [kHz]}$$

Avec  $\text{DIV} = 183$  on obtient approximativement 100 kHz, avec 457, 40 kHz.

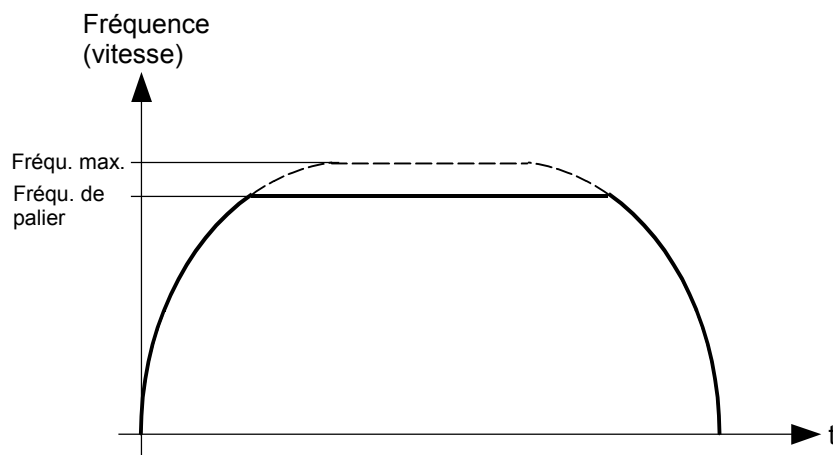


Figure 3-3 : Fréquence ou vitesse en fonction de t

### 3.3.3 Constante d'accélération KUP, et de décélération KDN

Ces paramètres fixent la pente initiale (respectivement finale) des rampes. Ils s'expriment en kHz/s ou en kimpulsions/s<sup>2</sup>. Comme au paragraphe précédent, il s'agit des impulsions à l'entrée de l'étage de puissance.

Des valeurs de 200 à 2000 kHz/s sont en général satisfaisantes.

### 3.3.4 Constante de Vitesse FEEDK

La constante de vitesse (ou constante d'avance) a pour but l'expression des vitesses en unités techniques: mm/s, m/min, t/min etc.

Pour les entraînements pas-à-pas :

**FEEDK = fréquence des impulsions pour une unité de vitesse [kHz]**

**Exemple:** (Voir paragraphe 3.3.1.)

On veut exprimer la vitesse en m/min.

SCALEK = 640, soit 640 impulsions pour 1 mm

ou encore 640'000 impulsions pour 1 m,

ou encore une fréquence de 640 kHz pour 1 m/s,

soit 640/60 kHz pour 1 m/min, **FEEDK = 10,667**

La vitesse effectivement atteinte ne saurait dépasser la vitesse maximale fixée par DIV. Exprimée dans l'unité choisie, la plus grande vitesse réalisable est donc :

**18290/DIV/FEEDK**

**N.B.** La vitesse d'avance ne correspond au calcul que si le potentiomètre est tourné à fond dans le sens horaire. Il est en jeu dans tous les mouvements à l'exception de la prise de référence et du test de fermeture.

### 3.3.5 Course de l'axe, STROKE + / STROKE –

- Ces paramètres, exprimés dans l'unité de longueur choisie au paragraphe 3.3.1, limite la course dans le sens positif et négatif de l'axe dans les mouvements programmés et les mouvements manuels. Respecter la condition du §4.4.1 p.15.

### 3.3.6 Sur-excitation du moteur : commande BOOST

Ce paramètre définit l'action du signal /BOOST et ne peut prendre que l'une des valeurs 0, 1, 2, 3.

BOOST = 0 : Le signal /BOOST est toujours inactif (haut).

BOOST = 1 : Le signal /BOOST est "bas" pendant les mouvements, "haut" en dehors des mouvements de l'axe

BOOST = 2 : Le signal /BOOST est toujours actif (bas).

BOOST = 3 : Le signal /BOOST est "haut" pendant les mouvements, "bas" en dehors des mouvements de l'axe.

Avec le translateurs intégré dans le E-100, on utilise normalement BOOST = 1. De cette façon, le courant est nominal pendant le mouvement et il est réduit à 60 % au repos.

On peut travailler avec BOOST = 0 pour de petits moteurs ou alors avec BOOST = 2 si un couple de maintien maximal est requis à l'arrêt.

### 3.3.7 Mode pour impulsions de commande

Permet de choisir le mode impulsion/direction ou quadrature pour les signaux de commande envoyés à l'ampli. Le mode pour les translateurs EIP est le mode impulsion-direction. Pour les amplis Yaskawa le mode quadrature est préféré.

PULSE = 0 : Le mode impulsion-direction est activé

PULSE = 1 : Le mode quadrature est activé

## 4 Mode opératoire du clavier et fonctions de UNIPROG+.

Ce chapitre décrit l'utilisation de la commande E-100 sous le programme UNIPROG+. La description part de la mise sous tension et suppose le raccordement effectué. Les menus ne sont pas abordés dans l'ordre d'affichage, mais bien dans l'ordre chronologique requis pour une première mise en service. La programmation, l'usage de l'éditeur, l'exécution des programmes et leur test font l'objet des chapitres suivants.

### 4.1 Mise sous tension de la commande E-100.

Le message suivant apparaît indiquant quelles versions sont contenues dans la commande.

```
* UNIPROG + V x.xx *  
E-100 x.xx
```

Après cela, UNIPROG+ exécute éventuellement le programme d'initialisation et présente le premier menu:

```
0 EXECUTION  
START STOP VECT MODE
```

Pendant l'exécution du programme d'initialisation (et de tout autre programme) l'opérateur ne peut utiliser les fonctions des menus à l'exception des utilitaires TRACE et I/O. Le programme peut être arrêté par la touche STOP (F2).

Si à la mise sous tension le message suivant apparaît

```
DEFAULT AMPLI ou  
ARRET D'URGENCE : ESC
```

la commande reçoit un signal de défaut de l'amplificateur. Cela peut aussi être une détection de fin-de-course ou un arrêt d'urgence activé. Dans le cas du fin-de-course, les touches flèches gauche et droite permettent de dégager l'axe. Dans le cas de l'a.u. le réarmement du circuit fera disparaître le message.

Si le message "COURSE DEPASSEE" apparaît, c'est que la condition du §4.4.1 p.15 n'est pas respectée

## 4.2 Sélection des Menus

Pour choisir un menu, il faut l'amener à l'affichage à l'aide des touches ↓ ou ↑. Les menus suivants défilent:

|                      |
|----------------------|
| 0 EXECUTION          |
| START STOP VECT MODE |

|               |
|---------------|
| 1 MODE MANUEL |
| JOG CLOS I/O  |

|                     |
|---------------------|
| 2 PROGRAMMATION     |
| EDIT TOOL FEED SAVE |

|                     |
|---------------------|
| 3 UTILITAIRES PROG. |
| DIR DEL COPY LOAD   |

|                 |
|-----------------|
| 4 CONFIGURATION |
| MGEN REF CTRL   |

|                     |
|---------------------|
| 5 DIVERS            |
| VER COUNT CLK ACCES |

On choisit les fonctions par les touches F1 à F4. La touche **ESC** permet toujours de sortir d'une fonction.

Si le message :

|                    |
|--------------------|
| NON ACCESSIBLE     |
| Presser une touche |

apparaît lorsqu'on tente d'entrer dans une fonction, cette fonction est protégée par le code d'accès, voir paragraphe 4.3.1.

## 4.3 Menu "DIVERS"

### 4.3.1 'ACCES' Accès aux protections

Afin de donner un accès sélectif aux différentes fonctions d'UNIPROG+, des drapeaux d'accès peuvent être attribués individuellement. On peut, par exemple, permettre à l'utilisateur final d'accéder aux mouvements manuels mais pas à l'éditeur.

Quel que soit l'état des drapeaux d'accès, l'introduction de la clé numérique donne l'accès général. A la mise sous tension de la commande, l'accès aux fonctions protégées est toujours fermé.

Pour obtenir l'accès général à toutes les fonctions, amener le menu ACCES. Le message "ENTER ACCESS CODE" invite à entrer la clef d'accès qui est le nombre

**31415**

(En général, toute entrée de nombre se termine par une pression sur ENTER et pendant l'introduction d'un nombre on peut corriger les erreurs de frappe par la touche CLR.)

Une pression sur ESC permet de retourner aux menus; toutes les fonctions sont maintenant accessibles.

L'introduction de la clé lorsque l'accès général est établi referme les fonctions protégées.

Pour placer les drapeaux d'accès individuels, rester dans le menu "ACCES" après avoir introduit la clé. Les fonctions ou groupes de fonctions apparaissent à l'affichage. Comme dans tous les menus, les flèches permettent de parcourir la liste. L'introduction d'un "1" donne l'accès, un "0" ne donne l'accès à la fonction qu'à l'aide de la clé.

### 4.3.2 'VER' Version

Tant que la pression sur la touche F1 est maintenue, l'écran présente le nom et les numéros des différentes versions installées.

Ces informations sont importantes à relever lors d'éventuelles questions à notre service technique.

### 4.3.3 'CLK' réglage de l'horloge

La fonction REGL (F4) de ce menu permet de régler l'horloge. Les touches + et – changent les valeurs tandis que les touches flèches gauche et droite sélectionnent la valeur. ESC valide le changement.

### 4.3.4 'COUNT' compteur de cycles

Le nombre de départs cycle est comptabilisé. Le compteur est remis à 0 avec la touche CLR.

## 4.4 Menu "CONFIGURATION"

### 4.4.1 Configuration du Générateur de Mouvement (Motion GENERator)

La touche F2 introduit ce sous-menu à partir du menu CONFIGURATION.

Tous les paramètres de ce menu ont été discutés au chapitre 3. Pour mémoire, rappelons leur signification:

|                 |   |
|-----------------|---|
| <b>DIV</b>      | Diviseur de fréquence                                     |
| <b>KUP</b>      | Constante d'accélération                                  |
| <b>KDN</b>      | Constante de décélération                                 |
| <b>SCALEK</b>   | Facteur d'échelle pour la longueur                        |
| <b>FEEDK</b>    | Facteur d'échelle pour la vitesse                         |
| <b>STROKE +</b> | Course de l'axe positif                                   |
| <b>STROKE -</b> | Course de l'axe négatif                                   |
| <b>BOOST</b>    | Surexcitation (voir description des modules de puissance) |
| <b>PULSE</b>    | Mode pour impulsions de commande                          |

Condition à respecter:  $DIV * SCALEK * 1.1 * (|STROKE-| + STROKE +) \leq 2^{31} - 1$

### 4.4.2 Configuration de la Prise de Référence (REFerence)

La touche F3 introduit le menu de configuration du dispositif de prise de référence.

|                       |   |
|-----------------------|---|
| <b>REF INPUT</b>      | Numéro de l'entrée pour prise de référence (voir tableau). La prise de référence peut être inhibée en spécifiant le numéro 8. Dans ce cas, lors de l'appel de la fonction de prise de référence, le compteur de position est remis à 0 sans qu'il n'y ait de mouvement. |
| <b>SPEED TO REF</b>   | Vitesse pour prise de référence. Cette vitesse est donnée dans l'unité définie par FEEDK. Une vitesse négative, inverse le sens de prise de référence.  |
| <b>CLOSURE GAP</b>    | Tolérance de fermeture, voir paragraphe 4.5.2.  |
| <b>SWITCH:</b>        | Nature du contact ou du détecteur de référence. Entrer "1" pour un contact normalement ouvert, "0" pour un contact normalement fermé.   |
| <b>REF BACK SPEED</b> | Divise la vitesse de prise de référence pour quitter le contact, plus le diviseur est grand, plus grande sera la précision.   |



| Paramètre (REF INPUT) | Entrée physique                      | Remarque   |
|-----------------------|--------------------------------------|--|
| 0..7                  | IN 0..7                              |  |
| 8                     | -                                    | Mise à 0 sans mouvement                                  |
| 9                     | Signal FAULT de l'étage de puissance |  |
| 18                    | INA issu du E-600-18 pour Yaskawa    | Pour Yaskawa et en cas de switch commun limite-référence |
| 60                    | INA ou HOME SWITCH                   |  |
| 61                    | INB                                  |  |

**Tableau 4.1 : Paramètre 'REF INPUT'**

#### 4.4.3 Configuration des Entrées de Contrôle (CONTRoL)

Ce sous-menu introduit par F4 permet de fixer l'attribution d'entrées aux fonctions de contrôle de l'exécution du programme, en parallèle avec les touches de la face avant, voir le chapitre 7 et le paragraphe 5.3.

On introduit le numéro de l'entrée attribuée. Si la fonction n'est pas requise, entrer 64.

- EXTERNAL START** Départ du programme, contact normalement ouvert
- EXTERNAL PAUSE** Suspension de l'exécution et des mouvements, contact normalement fermé.
- EXTERNAL STOP** Avorte les programmes en cours. Contact normalement fermé (élément d'une chaîne de sécurité).

Ce sous-menu contient également les paramètres suivants :

- DISPLAY FORMAT 1-6.** Ce sous-menu contient également la définition du format de l'affichage des cotes; le nombre introduit fixe le nombre de chiffres à droite du point décimal
- 2 HANDS START** Permet de configurer le start à deux mains sur 2 entrées choisies sous EXTERNAL START et 2 HANDS START. (Valeur de 0 à 8).
- Dans ce cas, seules les entrées 0 à 7 sont utilisables pour ces deux paramètres.
  - La valeur 8 introduite dans le paramètre 2 HANDS START rétablit le start simple.
- LAST TOOL NB** (Dernier numéro d'outil) permet de limiter le nombre d'outils utilisés, afin de ne pas remplir la mémoire de programme inutilement. (Valeur de 0 à 63).
- MAX RPM 10 volts 0** Permet de configurer la vitesse de la broche en fonction de la tension du DAC 0. Entrer la vitesse de la broche en t/min qui doit correspondre à une valeur DAC 0 de 10V (cette valeur peut varier entre 0 et 99999).
- MAX RPM 10 volts 1** Permet de configurer la vitesse de la broche en fonction de la tension du DAC 1. Entrer la vitesse de la broche en t/min qui doit correspondre à une valeur DAC 1 de 10V (cette valeur peut varier entre 0 et 99999).

|                         |  |
|-------------------------|--|
| <b>FEED CTRL BY ADC</b> | Si ce paramètre vaut 0, la vitesse des moteurs pas à pas est contrôlée par l'entrée ADC.<br><br>Si ce paramètre vaut 1, la vitesse est contrôlée par le potentiomètre.   |
| <b>LAST DELAYED OUT</b> | Permet de configurer un nombre de sorties dont la remise à zéro est retardée. (Valeur de 0 à 7). <ul style="list-style-type: none"> <li>• Le paramètre est compris entre 0 et 7, il donne le nombre total de sorties retardées à partir de 0.</li> <li>• Avec 0 aucune sortie n'est retardée. Avec 7 les sorties 0,1,2,3,4,5,6 et 7 sont retardées.</li> <li>• La valeur du retard par défaut vaut 1 s, cette valeur pouvant être modifiée dans le programme (voir SET BRK_D, au paragraphe 5.2.3).</li> </ul> |
| <b>LANGUAGE</b>         | Permet d'afficher certains messages dans les langues suivantes: 0 = Anglais, 1 = Français, 2 = Allemand.   |
| <b>JOGGING SPEED</b>    | Vitesse de jogging (mouvement manuels), unités techniques  |

## 4.5 Menu "MODE MANUEL"

Ce menu introduit 3 fonctions ou sous-menus permettant d'effectuer des mouvements sous contrôle manuel et d'afficher les cotes de l'axe.

Les fonctions TOOL (Tableau d'outils), JOG (Jogging) et CLOS (Closure check) ne sont pas accessibles pendant l'exécution d'un programme.

### 4.5.1 Mouvements Manuels, JOG

Dans ce sous-menu, la ligne supérieure de l'affichage montre la cote de l'axe dans le référentiel de base, la ligne inférieure indique l'outil courant et l'"incrément" en vigueur lors de la pression des touches "flèches".

- **Action des touches flèches gauche et droite:**

Une pression sur l'une de ces touches effectue un mouvement de la longueur affichée pour autant que la touche soit maintenue pendant tout le mouvement. Le relâchement de la touche, stoppe le mouvement avec une décélération gouvernée par KDN. Une nouvelle pression sur les touches JOG +/- produit à nouveau un incrément complet.

- **Action des touches F3 et F4 :**

Ces touches permettent de choisir la valeur de l'incrément de mouvement.

- **Action des touches flèches haut et bas:**

Elle permet de se déplacer entre la ligne supérieure et inférieure.

Si le curseur est sur la ligne inférieure, une valeur d'incrément peut être entrée avec le pavé numérique, au lieu des touches F3 et F4.

Si le curseur est sur la ligne supérieure, une coordonnée de destination peut être entrée au moyen du pavé numérique. Le mouvement est lancé à la pression de la touche ENTER. Cette touche doit rester pressée durant tout le mouvement.

**Le menu JOG permet en plus d'enregistrer les origines des outils.** La sélection de l'outil se fait au moyen des touches F1 et F2.

L'enregistrement d'une origine se fait de la manière suivante:

- Atteindre la position voulue en Jogging.
- Sélectionner l'outil concerné.
- Une pression de la touche F5 autorise l'entrée d'une nouvelle valeur d'origine pour l'outil.
- Entrée de la valeur au pavé numérique puis validation de cette dernière en pressant la touche ENTER.

Les nouvelles origines sont stockées dans le fichier 0, qui contient des "FDATA". Pour consulter ces origines plus facilement, aller dans le menu TOOL du groupe MOTION CONTROL. Des modifications peuvent également être faites depuis ce menu.

Pour de petits ajustements, la correction d'origine d'outil peut être entrée de façon incrémentale avec les touches "+" et "-" du jogging. La valeur, affichée dans l'angle inférieur droit (l'incrément), sera soustraite ou additionnée à l'origine. Cela sans générer de déplacement.

Dans l'angle inférieur gauche l'affichage indique "ALL". Si F1 est pressé la correction d'origine s'applique sur tous les outils, La sélection est confirmée par la LED F1 clignotante et par le texte en bas à gauche "ADJ ALL" (ajuste tout). Dans cette situation chaque pression sur une touche "flèche" décale l'origine de tous les outils.

Une pression de la touche F7 (REF), provoque une prise de référence. A la mise sous tension aucun mouvement ne peut s'effectuer si la prise de référence n'a pas été faite. Toutefois les mouvements avec le jogging sont possibles.

#### **4.5.2 Contrôle de fermeture, CLOS**

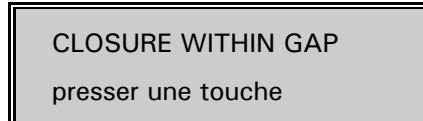
L'entraînement à moteur pas-à-pas se fait en boucle ouverte (sans feed-back). Il est souvent souhaitable de détecter les éventuelles anomalies de fonctionnement conduisant à des erreurs de position en vérifiant périodiquement la position 0 de l'axe avec une tolérance donnée.

Le contrôle de fermeture se fait naturellement autour du contact de référence et la tolérance est introduite sous "CLOSURE GAP". Ce paramètre est exprimé dans l'unité de l'axe et le contrôle est réputé bon si la fermeture a lieu dans une bande de  $\pm$  GAP autour du contact.

La mesure s'effectue dans les sens de la prise de référence pour éliminer l'influence des jeux et hystérésis. Si le contrôle détecte une erreur de position, le programme est stoppé et l'erreur est affichée. Dans tous les cas, l'axe se trouve à la position de référence après le contrôle de fermeture.

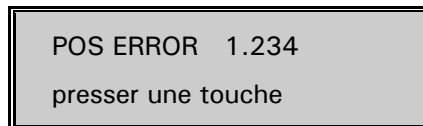
Presser la touche F3 pour accéder à ce sous-menu. Il suffit de presser la touche REF. Le résultat du contrôle s'affiche à l'écran:

Si le contrôle est bon, le message suivant apparaît :



CLOSURE WITHIN GAP  
presser une touche

dans le cas contraire :



POS ERROR 1.234  
presser une touche

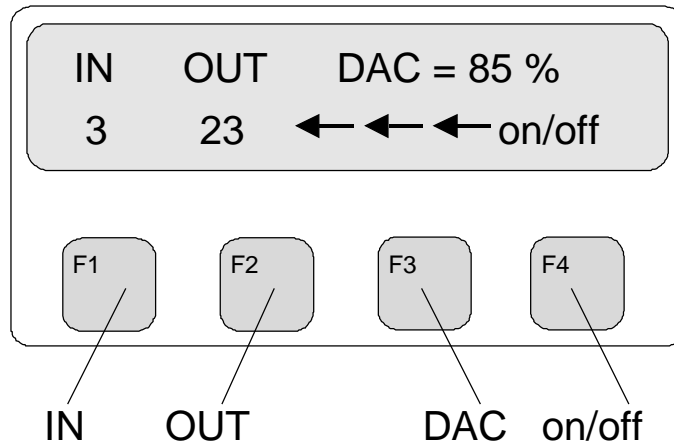
Si l'axe n'a pas de contact de référence attribué, le contrôle réagit comme s'il était bon. Rappelons qu'après un contrôle de fermeture, l'axe se trouve à son point de référence.

### 4.5.3 Utilitaire "I/O" control

Cette fonction est extrêmement utile pour la mise au point d'une installation et pour la recherche de pannes. Elle permet la lecture des états de toutes les entrées et sorties sous contrôle d'UNIPROG+, la modification de l'état des sorties et le chargement du registre du convertisseur digital/analogique (DAC).

Pour la signification et la numérotation des entrées et des sorties, se reporter au Tableau 5.1

L'écran du menu I/O CONTROL regroupe les affichages d'une entrée, d'une sortie et du DAC:



**Figure 4-1 : Menu "I/O Control"**

Dans cet exemple, l'état de l'entrée 3 est affiché sur le témoin de la touche F1, l'état de la sortie 23 sur le témoin de F2 et le DAC est à 85 % de sa valeur de fin d'échelle, soit 8,5 V. Les touches F1, F2 et F3 déplacent le curseur respectivement sur IN, OUT et DAC. Pour changer le numéro d'une entrée ou d'une sortie, placer le curseur et entrer le numéro. Le procédé est le même pour la valeur à charger dans le DAC. Une 2<sup>e</sup> pression sur F3 choisit le DAC 1.

- La touche F4 modifie l'état de la sortie affichée de façon rotative (toggle).
- Il faut remarquer que cette fonction peut être utilisée pendant l'exécution d'un programme ce qui facilite le diagnostic de dysfonctions telles que le défaut de quittance.
- Selon le Tableau 5.1, les pseudo-I/O numéro 8, 9, 10 sont les états d'activation des programmes simultanés (paragraphe **Erreur! Source du renvoi introuvable.**); la touche F4 n'a pas d'effet sur ces trois pseudo-I/O. Les pseudo-I/O 11...15 sont des drapeaux et les états IN et OUT coïncident. La pseudo I/O numéro 58 est le flag RUN donnant l'état d'un mouvement, 0 = arrêté 1 = marche.

## 4.6 Menu "PROGRAMMATION".

Ce menu contient toutes les fonctions nécessaires à l'écriture des programmes. L'éditeur UNIPROG+ sera discuté au chapitre 6 après l'étude des instructions.

### 4.6.1 Tableau d'outils "TOOL".

Dans ce menu, les origines des outils sont accessibles

Les valeurs visibles dans ce tableau d'outils sont déposées dans le fichier "0" qui est ouvert automatiquement lorsque l'on configure le paramètre "LAST TOOL NB". Si par le fait d'une manipulation erronée, le fichier "0" ne présente pas la taille adaptée, le message suivant apparaît :

FICH. 0 NON FORMATE

Presser une touche

Dans ce cas, le paramètre "LAST TOOL NB" doit être reconfiguré.

#### 4.6.2 Vitesses Présélectionnées (FEED rates)

La vitesse présélectionnée entrant dans les instructions de mouvements est puisée dans le tableau de ce menu. Les 7 vitesses, 0 à 6, peuvent être introduites. Il faut les exprimer dans l'unité choisie pour le calcul de FEEDK.

#### 4.6.3 Sauvegarde dans la mémoire FLASH (SAVE)

Cette fonction mémorise le contenu de la mémoire de travail (RAM) dans la mémoire de sauvegarde (FLASH). **Le contenu de la mémoire FLASH sera remplacé.** Le contenu de la RAM est conservé lors de la mise hors tension de l'appareil grâce à la pile intégrée. La fonction SAVE est utile pour prévenir une défaillance de la pile.

## 4.7 Manipulation de Fichiers, UTILITAIRES PROG.

### 4.7.1 Répertoire, (DIRectory)

L'écran présente des informations sur chaque fichier ouvert. Un fichier peut être ouvert par l'éditeur ou par copie d'un fichier existant.

| FILE | SIZE | PROT | FREE |
|------|------|------|------|
| 12   | 45   | NO   | 670  |

Cet écran signifie que le fichier (FILE) 12 existe, que sa taille (SIZE) est de 45 lignes, qu'il n'est pas protégé (PROT) et qu'il reste 670 lignes disponibles (FREE). Si le fichier 18 n'est pas ouvert, le message suivant apparaît :

| FILE        | SIZE | PROT | FREE |
|-------------|------|------|------|
| INTROUVABLE |      |      | 670  |

Différentes façons d'utiliser l'utilitaire DIR sont offertes :

- Examen de tous les fichiers :  
Les touches flèches permettent de balayer, dans les deux sens, l'ensemble des fichiers ouverts.
- Examen d'un fichier particulier :  
Entrer directement son numéro (suivi de ENTER), l'un des deux écrans ci-dessus apparaît.
- Modification de la protection :  
La touche F3 agit en pas-à-pas (toggle) et fait apparaître à la rubrique PROT les mots YES ou NO. Un fichier protégé ne peut être modifié par l'éditeur ou effacé. En bloquant l'accès à l'utilitaire DIR et en laissant l'éditeur ouvert, on pourra protéger sélectivement les fichiers.

### 4.7.2 Effacement d'un Fichier (DELeTe)

L'écran invite à composer le numéro du fichier à effacer. Afin d'éviter tout effacement involontaire, le message "presser CLR pour EFF" invite à presser la touche CLR. On peut alors encore revenir sans action au menu de base en pressant ESC. Après effacement on retrouve le menu "FILE UTILITIES". Si le fichier est protégé, on ne retourne pas au menu de base mais au répertoire avec le fichier protégé à l'affichage.

### 4.7.3 Copie d'un Fichier (COPIY)

L'écran invite à composer le numéro du fichier à copier (SOURCE FILE), puis celui du fichier de destination (DEST. FILE). Plusieurs actions peuvent se produire :

- Le fichier source n'est pas ouvert: aucune action, retour à "FILE UTILITIES"

- Le fichier destination n'est pas créé ou ouvert: un nouveau fichier est créé.
- Le fichier destination existe déjà: le fichier source est mis à la suite du fichier destination.
- Il n'y a pas assez de place dans l'espace des 1250 lignes pour la copie du fichier source: aucune action mais le fait est signalé :



TOO LARGE  
press any key

#### 4.7.4 Chargement depuis la mémoire FLASH (LOAD)

Cette fonction charge le contenu de la mémoire FLASH (mémoire de sauvegarde) dans la mémoire RAM qui est la mémoire de travail. **Le contenu de la mémoire RAM sera effacé et remplacé.**



## 5 Instructions du langage UNIPROG+

Les instructions sont décrites ci-après en utilisant les symboles "mnémoniques" apparaissant à l'écran lors de l'édition ou dans l'emploi de l'utilitaire TRACE. Le code numérique nécessaire pour l'entrée au clavier de la commande E-100 accompagne la description des instructions. Dans le descriptif formel d'une instruction, les symboles seront écrits en majuscule et le genre de l'argument en minuscules.

Une instruction ou une donnée numérique occupe une ligne dans la zone de la mémoire réservée à l'utilisateur. Nous appellerons **adresse** d'une ligne le nombre formé par la mise bout à bout du numéro de la ligne dans le fichier (ou le programme) et du numéro du fichier écrit à deux chiffres.

**Exemples:** 1245 est l'adresse de la ligne 12 du programme 45,  
102 est l'adresse de la ligne 1 du programme 2,  
6 est l'adresse de la ligne 0 du programme 6....

L'éditeur utilise le vocable "LINE/PROG" pour "adresse".

Les instructions peuvent puiser leur argument numérique principal -la cote dans les déplacements, le temps dans un temporisateur- de plusieurs façons.

- Argument numérique **immédiat** : La valeur numérique se trouve dans la ligne contenant l'instruction.
- Argument numérique **direct** : Dans l'instruction on trouve l'adresse de la ligne contenant la valeur numérique.
- Argument numérique **indirect** : l'instruction contient l'adresse d'un pointeur contenant lui-même l'adresse de la valeur numérique.

**exemples:** voir instructions POSA, POSAD, POSAI ci-dessous

UNIPROG+ peut gérer trois programmes simultanés (fonctionnement multi-tâches). Chaque programme simultané possède un **accumulateur**, registre par lequel on peut faire transiter des quantités numériques.

### 5.1 Instructions de Positionnement

UNIPROG+ prévoit 12 instructions de positionnement : 6 effectuent des positionnements "absolus", c'est-à-dire qu'elles exigent comme argument une cote dans le référentiel courant (voir chapitre 3); les 6 autres effectuent des déplacements "relatifs".

Il faut remarquer qu'UNIPROG+ calcule le point d'arrivée d'un déplacement relatif dans le référentiel initial; une suite de déplacements relatifs ne conduit donc pas à un cumul des erreurs d'arrondi.

### 5.1.1 Positionnement absolu avec attente de la fin du mouvement :

---

|    |       |             |                         |
|----|-------|-------------|-------------------------|
| 10 | POSA  | < vitesse > | < cote >                |
| 11 | POSAD | < vitesse > | < adresse de la cote >  |
| 12 | POSAI | < vitesse > | < adresse du pointeur > |

---

### 5.1.2 Positionnement absolu sans attendre la fin du mouvement :

---

|    |       |             |                         |
|----|-------|-------------|-------------------------|
| 72 | PANW  | < vitesse > | < cote >                |
| 73 | PADNW | < vitesse > | < adresse de la cote >  |
| 74 | PAINW | < vitesse > | < adresse du pointeur > |

---

### 5.1.3 Positionnement relatif avec attente de la fin du mouvement :

---

|    |       |             |                         |
|----|-------|-------------|-------------------------|
| 14 | POSR  | < vitesse > | < déplacement >         |
| 15 | POSRD | < vitesse > | < adresse déplacement > |
| 16 | POSRI | < vitesse > | < adresse du pointeur > |

---

### 5.1.4 Positionnement relatif sans attendre la fin du mouvement :

---

|    |       |             |                         |
|----|-------|-------------|-------------------------|
| 75 | PRNW  | < vitesse > | < déplacement >         |
| 76 | PRDNW | < vitesse > | < adresse déplacement > |
| 77 | PRINW | < vitesse > | < adresse du pointeur > |

---

|                        |  |
|------------------------|--|
| Argument "vitesse ":   | Accepte un entier 0..7. De 0 à 6, la vitesse est prise dans le tableau du paragraphe 4.6.2. Si vitesse = 7, sa valeur est prise dans l'accumulateur. |
| Argument "cote"        | Position absolue dans le référentiel de l'outil courant  |
| Argument "déplacement" | Distance à parcourir depuis la position actuelle   |

### 5.1.5 Arrêt d'un mouvement

---

|    |       |
|----|-------|
| 78 | STOPM |
|----|-------|

---

Permet d'arrêter un mouvement avant qu'il ait atteint sa destination. Le mouvement s'arrête avec décélération.

### 5.1.6 Définition des outils

---

|    |      |                    |
|----|------|--------------------|
| 19 | TOOL | < numéro d'outil > |
|----|------|--------------------|

---

L'instruction TOOL fixe un nouveau référentiel valable pour toutes les instructions de positionnement subséquentes. La composante du vecteur de translation de l'origine est déposée dans le fichier "0".

- Le tableau "TOOL" permet la consultation ou la modification de ces origines.
- Ces origines peuvent aussi être apprises et validées directement depuis le jogging.
- L'argument "numéro d'outil" sélectionne la composante associée à un numéro d'outil.

## 5.2 Autres instructions Cinématiques

### 5.2.1 Prise de référence :

---

|    |     |
|----|-----|
| 17 | REF |
|----|-----|

---

La prise de référence se fait comme indiqué au paragraphe 3.1. La vitesse de mouvement est prise dans la configuration REF, voir paragraphe 4.4.2.

Respecter la condition du §4.4.1 p.15

### 5.2.2 Test de fermeture :

---

|    |      |           |
|----|------|-----------|
| 18 | CLOS | <vitesse> |
|----|------|-----------|

---

La fonction du test de fermeture fait l'objet du paragraphe 4.5.2. L'argument "vitesse" est celui des instructions de positionnement (paragraphe 5.1.3).

### 5.2.3 Définition de paramètres

---

|    |     |                       |                       |
|----|-----|-----------------------|-----------------------|
| 83 | SET | <numéro du paramètre> | <Valeur du paramètre> |
|----|-----|-----------------------|-----------------------|

---

Numéro du paramètre:

- |   |       |  |
|---|-------|--|
| 0 | PASSE | Valeur de la passe de débouillage.                               |
| 1 | GAP   | Lors du retour rapide, garde avant l'avance lente.               |
| 2 | DELAY | Délai de fond de perçage.  |
| 3 | BRK-D | Retard de déclenchement des sorties 0 à 7 (LAST DELAYED OUTPUT). |

Les numéros de paramètre 0, 1, 2 concernent le perçage.

### 5.2.4 Cycle de perçage

---

|    |      |                 |                            |          |
|----|------|-----------------|----------------------------|----------|
| 84 | PECK | <vitesse lente> | <position fond de perçage> | <mode-d> |
|----|------|-----------------|----------------------------|----------|

---

Le cycle de perçage détermine le nombre de passes à effectuer. La passe sera toujours un nombre positif. La position atteinte en fin de cycle de perçage est la position absolue par rapport à l'outil sélectionné.

4 modes sont possibles :

- |            |   |
|------------|---|
| Mode-d = 0 | Débouillage avec retour en fin de cycle à la position de départ.        |
| Mode-d = 1 | Brise copeaux avec retour en fin de cycle.                              |
| Mode-d = 2 | Débouillage <b>sans</b> retour en fin de cycle à la position de départ. |
| Mode-d = 3 | Brise copeaux <b>sans</b> retour en fin de cycle.                       |

- Le retour rapide s'arrête en retrait de l'usinage. La valeur du retrait est appelée la garde (GAP), elle est configurée par l'instruction SET 1 GAP et vaut par défaut 0.1 mm.
- Une temporisation configurée par l'instruction SET 2 DELAY est active en fin de

perçage en mode 0 et 1. La temporisation vaut par défaut 0.1 s.

- Les modes 2 et 3 permettent d'enchaîner plusieurs perçages pour obtenir de la progressivité.

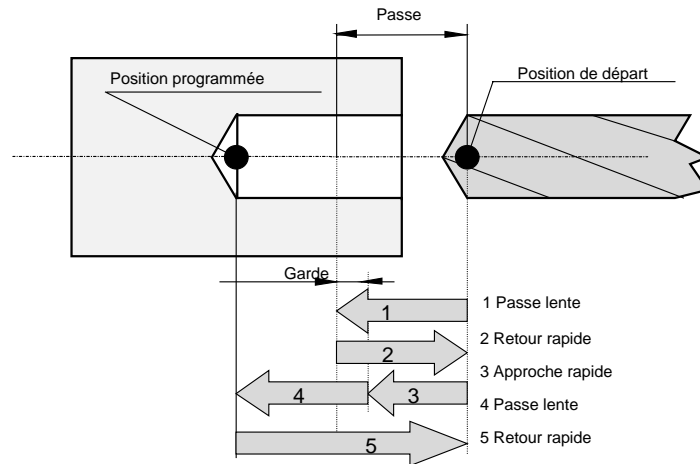


Figure 5-1 : Cycle de perçage

## 5.2.5 Instruction de taraudage

---

**81 TPING** <pas> <position fond de taraudage>

---

Cette instruction facilite le choix de l'avance en fonction de la vitesse de rotation de la broche et du pas.

- L'instruction SPVEL détermine la vitesse de rotation de la broche, pilotée par un convertisseur de fréquence.
- Le pas est choisi dans le menu FEED.
- Pour inverser la rotation en fin de taraudage, la sortie 7 s'inverse. Elle peut ainsi être câblée sur l'entrée "sens" du convertisseur de fréquence.
- La rotation de la broche s'inverse lorsque la position finale de l'axe d'avance est atteinte. La rotation de la broche n'est pas mesurée et n'a donc pas d'incidence sur la durée du taraudage. Par conséquent, la profondeur du taraudage peut varier en raison inverse du couple de taraudage.

**Exemple:** Taraudage pas de 1 mm; Rotation 600 t/min; profondeur 50

Menu FEED : RATE # 3 = 1.00 = Pas

```

00 30 10 POSA 4 0.00
01 30 57 SPVEL 600 ; Vitesse de rotation
02 30 29 ON 7 ; Impose le sens sortie 22 à 1
03 30 29 ON 6 ; Démarre le convertisseur
04 30 72 TPING 3 50 ; Démarre le taraudage

```

**Remarque:**

l'avance est déterminée sans être asservie sur la vitesse de rotation de la broche ce qui

exige l'usage d'un porte-outil avec **compensation de longueur** pour absorber les variations de synchronisation.

## 5.3 Instructions d'entrées/sorties

Ces instructions permettent de brancher le programme en fonction de l'état d'une entrée, d'attendre selon cet état et modifier l'état d'une sortie.

### 5.3.1 Attente sur entrée :

---

|    |       |          |
|----|-------|----------|
| 20 | WAIT0 | <entrée> |
| 21 | WAIT1 | <entrée> |

---

Le programme attend tant que l'entrée désignée soit 0 ou 1 respectivement.

### 5.3.2 Branchement sur entrée :

---

|    |       |          |           |
|----|-------|----------|-----------|
| 22 | BRINO | <entrée> | <adresse> |
| 23 | BRIN1 | <entrée> | <adresse> |

---

Si l'entrée désignée est 0 ou 1 resp., le programme branche à la ligne "adresse". Dans le cas contraire, l'instruction suivante est exécutée. Au sujet de l'adresse de branchement, voir la remarque du paragraphe 5.7.1.

### 5.3.3 Commande des sorties :

---

|    |     |          |
|----|-----|----------|
| 28 | OFF | <sortie> |
| 29 | ON  | <sortie> |

---

La sortie désignée est déclenchée ou enclenchée.

UNIPROG+ reconnaît comme entrées et sorties les ensembles de variables booléennes du Tableau 5.1

| Numéro dans instruction | Entrée physique ou flag | Numéro dans instruction | Sortie physique ou flag       |
|-------------------------|-------------------------|-------------------------|-------------------------------|
| 0                       | IN(0)                   | 0                       | OUT(0)                        |
| 1                       | IN(1)                   | 1                       | OUT(1)                        |
| 2                       | IN(2)                   | 2                       | OUT(2)                        |
| 3                       | IN(3)                   | 3                       | OUT(3)                        |
| 4                       | IN(4)                   | 4                       | OUT(4)                        |
| 5                       | IN(5)                   | 5                       | OUT(5)                        |
| 6                       | IN(6)                   | 6                       | OUT(6)                        |
| 7                       | IN(7)                   | 7                       | OUT(7)                        |
| 8                       | SIM(0)                  | 8                       | SIM(0)                        |
| 9                       | SIM(1)                  | 9                       | SIM(1)                        |
| 10                      | SIM(2)                  | 10                      | SIM(2)                        |
| 11                      | FLAG(0)                 | 11                      | FLAG(0)                       |
| 12                      | FLAG(1)                 | 12                      | FLAG(1)                       |
| 13                      | FLAG(2)                 | 13                      | FLAG(2)                       |
| 14                      | FLAG(3)                 | 14                      | FLAG(3)                       |
| 15                      | FLAG(4)                 | 15                      | FLAG(4)                       |
| 16..57                  | IN(16)..IN(57)          | 16..57                  | OUT(16)..OUT(57)              |
| 56                      | Boost (Enable Yaskawa)  | 56                      | Sortie Boost (enable Yaskawa) |
| 58                      | Flag RUN                | 58                      | Flag RUN                      |
| 60                      | INA ou HOME SWITCH      |                         |                               |
| 61                      | INB                     |                         |                               |

**Tableau 5.1 : Entrées/Sorties UNIPROG+**

- Les pseudos I/O, SIM0, SIM1, SIM2, sont les états d'activation des trois programmes simultanés de UNIPROG + .
- Les FLAG(0..4) ne sont pas véritablement des entrées mais des marqueurs -ou drapeaux- internes, librement utilisables par le programmeur. Ces 5 marqueurs sont mis à 0 ou à 1 par les instructions OFF et ON.
- Les entrées IN(16..55) et les sorties OUT(16..55) sont matérialisées par des modules d'extension I/O connectés par le bus d'extension. Chaque module porte une adresse à sélectionner sur son commutateur.

| Sélecteur |     |     |     | Adresse<br>IN OUT |
|-----------|-----|-----|-----|-------------------|
| 1         | 2   | 3   | 4   |                   |
| on        | off | off | off | 16...23           |
| off       | on  | off | off | 24...31           |
| on        | on  | off | off | 32...39           |
| off       | off | on  | off | 40...47           |
| on        | off | on  | off | 48...55           |

**Tableau 5.2 : Adresse des modules IN/OUT**

Les sorties OUT(0) à OUT(7) sont les sorties comprises dans le boîtier E-100 et disponibles sur le connecteur de la face arrière. Idem pour les entrées IN(0) à OUT(7).

### 5.3.4 Inversion d'une sortie

---

**95 CPL < numéro de la sortie >**

---

Cette instruction complémente (inverse) la sortie en argument.

**Exemple:**

Pour l'activation et la désactivation (toggle) d'une sortie 18 dans un programme manuel 90 (touche 0):

```
90 00 95 CPL 18
```

## 5.4 Instructions d'Affichage

### 5.4.1 Affichage direct

---

**79 DISPD < position > < adresse >**

---

L'instruction DISPD (Display Direct) affiche le contenu numérique de la ligne adressée. La position est un nombre qui spécifie le quadrant dans lequel sera affiché le nombre:

0 = coin supérieur gauche

1 = coin supérieur droite

2 = coin inférieur gauche

3 = coin inférieur droite

L'instruction DISPD commute la fenêtre d'écran, c'est-à-dire que les informations affichées par un utilitaire ne sont pas perdues mais peuvent être restituées par l'instruction RBW (Restore Basic Window). Les touche F1 (START) et F2 (STOP) sont toujours actives.

### 5.4.2 Restitution de la fenêtre de base

---

**80 RBW**

---

Voir instruction DISPD ci-dessus.

### 5.4.3 Affichage de texte

Affichage d'un texte préalablement chargé en usine par EIP

---

**88 PRSTR < position > < numéro >**

---

< position > : voir l'instruction DISPD.

< numéro > : numéro du texte.

## 5.5 Manipulation de Quantités Numériques

UNIPROG+ distingue les nombres réels (Floating Numbers) des nombres entiers (Integers). Les vitesses, les cotes, les origines, les temps sont des réels; les adresses, les nombres de cycles sont des entiers.

Les instructions de manipulation de nombres et les instructions arithmétiques supposent a priori que l'argument adressé contient un réel, à l'exception d'une ligne "IDATA" dont le contenu est alors considéré comme entier. Il est donc sans autre possible de modifier par le programme le contenu de l'argument numérique - même immédiat- d'une instruction. Ce contenu doit être un réel.

### 5.5.1 Chargement de l'accumulateur

Chargement immédiat de l'accumulateur:

---

|           |              |                   |
|-----------|--------------|-------------------|
| <b>50</b> | <b>FLOAD</b> | < nombre réel >   |
| <b>51</b> | <b>ILOAD</b> | < nombre entier > |

---

Chargement direct de l'accumulateur:

---

|           |              |             |
|-----------|--------------|-------------|
| <b>52</b> | <b>LOADD</b> | < adresse > |
|-----------|--------------|-------------|

---

Chargement indirect de l'accumulateur :

---

|           |              |                      |
|-----------|--------------|----------------------|
| <b>53</b> | <b>LOADI</b> | < adresse pointeur > |
|-----------|--------------|----------------------|

---

Chargement de la position absolue (0 machine) dans l'accumulateur:

---

|           |              |  |
|-----------|--------------|--|
| <b>68</b> | <b>PLOAD</b> |  |
|-----------|--------------|--|

---

### 5.5.2 Dépose de l'accumulateur

Dépose directe de l'accumulateur :

---

|           |              |             |
|-----------|--------------|-------------|
| <b>55</b> | <b>STORD</b> | < adresse > |
|-----------|--------------|-------------|

---

Dépose indirecte de l'accumulateur:

---

|           |              |                      |
|-----------|--------------|----------------------|
| <b>56</b> | <b>STORI</b> | < adresse pointeur > |
|-----------|--------------|----------------------|

---

### 5.5.3 Incrémentation/Décrémentation d'un pointeur :

---

|           |             |             |
|-----------|-------------|-------------|
| <b>58</b> | <b>INCD</b> | < adresse > |
| <b>59</b> | <b>DECD</b> | < adresse > |

---

Ces deux instructions sont utilisées en liaison avec l'adressage indirect; elles permettent de passer à la ligne suivante ou précédente en ajoutant ou en retranchant 100 à l'adresse



composite, voir début du présent chapitre. Ces instructions ne permettent pas de franchir les limites d'un fichier.

### 5.5.4 Entrée de nombre

Entrée d'un nombre au clavier et dépose à l'adresse spécifiée

---

**54**    **NBINP**    <position>    <adresse>

---

L'argument <position> définit l'endroit où sera afficher le nombre tapé au clavier (voir l'instruction DISPD)

Après l'entrée du nombre et une pression sur la touche ENT, celui-ci est stocké à l'adresse contenant un IDATA ou un FDATA et le programme passe à la ligne suivante. L'instruction RBW doit être utilisée pour avoir à nouveau la fenêtre d'affichage standard.

## 5.6 Utilisation des entrées-sorties analogiques

### 5.6.1 Lecture d'une entrée analogique

---

**87**    **RADC**    <numéro de l'ADC>

---

La valeur de l'ADC spécifié est déposée dans l'accu sous la forme d'un réel valant 0.0 à 255.0.

L'argument vaut de 0 à 2. 0 étant le potentiomètre de la face avant du E-100.

### 5.6.2 Chargement d'une sortie analogique

---

**86**    **LDAC**    <numéro du DAC>

---

La valeur entière 0 à 255 se trouvant dans l'accu est chargée dans le DAC spécifié. L'argument vaut 0 ou 1.

### 5.6.3 Vitesse de la broche

---

**57**    **SPVEL**    <numéro du DAC>    <tours par minute>

---

Le premier argument (0 ou 1) choisit le DAC. Le 2<sup>e</sup> argument est directement la vitesse en tours par minute. Le paramètre MAX RPM 10 volts 0 ou 1 doit être configuré (avec la valeur 100 la vitesse de rotation est en pour-cent).

### 5.6.4 Vitesse de rotation des moteurs

---

**85**    **MOTOR**    <numéro du moteur>    <tours par minute>

---

Cette instruction réalise le changement de vitesse des moteurs de broches en assurant les délais impératifs de commutation des contacteurs du convertisseur de fréquence. (seulement pour DAC 0).

En effet le convertisseur de fréquence doit être commuté impérativement sans courant,

donc à vitesse nul.

Le cycle de l'instruction MOTOR met à zéro la vitesse puis attend le délai de l'instruction SET 4 BRK-D (BReaK-Delay), pour ensuite réactiver la vitesse directement mise en tour par minute.

Avec "MOTOR 0" tous les moteurs sont mis à zéro.

Avec "MOTOR 1" le moteur 2 est mis à zéro et le moteur 1 démarre (OUT 1).

Avec "MOTOR 2" le moteur 1 est mis à zéro et le moteur 2 démarre (OUT 2).

etc... jusqu'à MOTOR 7.

## 5.7 Instructions de Contrôle du Programme

### 5.7.1 Saut inconditionnel :

---

**60**    **JMP**    < adresse >

---

L'exécution du programme est transférée à "adresse".

**Remarque :** L'éditeur d'UNIPROG+ peut insérer et supprimer des lignes de programme, opérations qui modifient la numérotation des lignes dans un fichier. Il est donc recommandé (mais pas impératif) d'organiser les programmes de façon à ce que les sauts, appels de sous-programmes etc. se fassent à la ligne 0 d'un fichier.

### 5.7.2 Appel de sous-programme :

---

**61**    **CALL**    < adresse >

---

Le contrôle est transféré au sous-programme débutant à "adresse". A la fin du sous-programme, le contrôle revient à l'instruction qui suit directement l'appel.

- 10 niveaux d'imbrication des sous-programmes sont autorisés.

### 5.7.3 Fin de programme, de sous-programme :

---

**62**    **END**

---

Cette instruction marque la fin d'un programme ou d'un sous-programme. Si la dernière instruction d'un programme ou d'un sous-programme coïncide avec la dernière ligne du fichier, l'instruction END est superflue.

### 5.7.4 Répétition :

Les instructions suivantes permettent de construire des boucles à répétition sans avoir à préparer des compteurs de cycles. Il est licite d'imbriquer jusqu'à 10 boucles et une boucle peut s'étendre sur plusieurs fichiers. Une boucle doit nécessairement se terminer par

ENDRP.

---

|           |              |             |                         |
|-----------|--------------|-------------|-------------------------|
| <b>63</b> | <b>REP</b>   | < nombre >  | argument immédiat       |
| <b>64</b> | <b>REPD</b>  | < adresse > | argument direct, nombre |
| <b>65</b> | <b>ENDRP</b> |             | limite de la boucle     |

---

Dans l'instruction REP, l'argument immédiat est nécessairement un entier; dans l'instruction REPD, le contenu de la ligne adressée peut être un entier ou un réel. Dans ce dernier cas, le nombre de répétitions est égal à la partie entière.

### 5.7.5 Appel de programmes simultanés :

---

|           |            |            |             |
|-----------|------------|------------|-------------|
| <b>67</b> | <b>SIM</b> | < numéro > | < adresse > |
|-----------|------------|------------|-------------|

---

Le programme simultané -ou programme parallèle- dont le numéro (0,1 ou 2) est spécifié est mis en marche à l'adresse donnée dans l'instruction. Un programme simultané s'arrête de lui-même s'il contient une instruction END implicite ou explicite. Un programme simultané peut être mis en pause - à partir d'un programme parallèle - en imposant un zéro dans l'élément SIMO/1/2 (voir Tableau 5.1) par l'instruction OFF et réactivé par l'instruction ON. L'échantillonnage de SIMO.. SIM2 permet à un programme de connaître l'état des autres simultanés.

L'appel d'un simultané actif à une adresse quelconque suspend l'exécution en cours et ce simultané reprend son activité à la nouvelle adresse. La touche STOP avorte tous les simultanés.

### 5.7.6 Branchements conditionnels (Test de l'accumulateur):

Ces instructions permettent de tester le contenu de l'accumulateur, en général après une opération arithmétique. Veuillez noter que le code numérique de ces instructions les place après BRINO et BRIN1.

Si la condition de branchement n'est pas remplie, le programme exécute l'instruction suivante. Au sujet de l'argument "adresse", voir la remarque du paragraphe 5.7.1.

---

|           |            |             |  |
|-----------|------------|-------------|--|
| <b>24</b> | <b>BRM</b> | < adresse > | Branchement si le contenu de l'accu. est négatif         |
| <b>25</b> | <b>BRP</b> | < adresse > | Branchement si le contenu de l'accu. est positif ou nul. |

---

---

|           |             |             |   |
|-----------|-------------|-------------|---|
| <b>26</b> | <b>BRZ</b>  | < adresse > | Branchement si le contenu de l'accu. est nul.     |
| <b>27</b> | <b>BRNZ</b> | < adresse > | Branchement si le contenu de l'accu. est non nul. |

---

## 5.8 Instructions de Temporisation

---

|           |              |             |                                       |
|-----------|--------------|-------------|---------------------------------------|
| <b>70</b> | <b>WAIT</b>  | < temps >   | Argument immédiat                     |
| <b>71</b> | <b>WAITD</b> | < adresse > | Argument direct, temps dans [adresse] |

---

Ces instructions introduisent des temps morts dans un programme. Le temps est exprimé en secondes et le contenu de la ligne adressée par WAITD doit être un réel.

## 5.9 Instructions Arithmétiques

Les Instructions arithmétiques opèrent sur l'accumulateur et le contenu de leur argument direct ; Le résultat est retourné à l'accumulateur. La nature de l'opération est dictée par le type de l'argument direct. Plus précisément : si la ligne adressée est une ligne "IDATA", l'instruction effectue une opération sur des entiers, dans tous les autres cas, elle traite les opérandes comme des réels.

---

|    |             |             |                         |
|----|-------------|-------------|-------------------------|
| 91 | <b>ADDD</b> | < adresse > | Accu = Accu + [adresse] |
| 92 | <b>SUBD</b> | < adresse > | Accu = Accu - [adresse] |
| 93 | <b>MULD</b> | < adresse > | Accu = Accu . [adresse] |
| 94 | <b>DIVD</b> | < adresse > | Accu = Accu / [adresse] |

---

## 5.10 Instructions relatives à l'horloge interne

Le E-100 comporte une horloge temps réel qui est maintenue par la pile lorsque l'appareil est mis hors tension. Ces instructions donnent des valeurs qui doivent être traitées et testées avec les instructions concernant les valeurs numériques.

### 5.10.1 Lecture de la date

---

**69 RDATE**

---

Charge la date courante dans l'accu sous la forme d'un entier avec le format 'aammjj' :

Exemple: 4 décembre 2002

**021204**  
    ┌──┬──┬──┐  
    |  |  |  |  
Année mois jour

### 5.10.2 Lecture du jour de la semaine

---

**96 RDAY**

---

Charge le jour de la semaine (1..7) courant dans l'accu: 1 = Lundi, 7 = dimanche

### 5.10.3 Lecture de l'heure

---

**97 RTIME**

---

Charge l'heure courante dans l'accu sous la forme d'un entier avec le format 'hhmmss' :

Exemple: 9 h 12 min 5 sec

**091205**  
    ┌──┬──┬──┐  
    |  |  |  |  
Heure minutes secondes

## 5.11 NOP et "Pseudo-Instructions"

---

**90**    **NOP**

---

L'instruction NOP (NO Operation) est en général utilisée pour réserver des lignes en vue d'adjonctions ultérieures. Lors de l'édition, NOP apparaît toujours dans la ligne qui n'est pas encore ouverte.

---

**98**    **FDATA** <nombre réel>  
**99**    **IDATA** <nombre entier>

---

Il ne s'agit pas d'instructions mais l'éditeur utilise les préfixes FDATA et IDATA pour introduire des quantités numériques dans un fichier. Le préfixe "IDATA" est utilisé par les instructions pour reconnaître la nature des nombres.

Il est intéressant de noter que les pseudo-instructions 98 et 99 ont l'effet d'un NOP lorsqu'elles sont écrites dans un programme. Il est donc possible de placer des données numériques dans le corps d'un programme.

## 5.12 Marqueur de Pause

Chaque instruction peut être dotée d'un marqueur de pause qui est transparent lorsque le programme est exécuté en mode 1 mais qui suspend le déroulement du programme en mode 2. Pour plus de détails, se reporter au chapitre 7.

## 6 L'éditeur UNIPROG+

On entre dans l'éditeur en sélectionnant le menu "PROGRAMMING" puis la fonction EDIT comme indiqué au paragraphe 4.2. L'affichage invite alors l'opérateur à composer le numéro du fichier ou programme à éditer. La ligne 0 du programme arrive alors à l'écran:

```
POSA 3  12.234
10                0  p11
```

La première ligne affiche le symbole de l'instruction et la valeur de ses éventuels arguments. La ligne inférieure contient le code numérique de l'instruction (sous le symbole) et les numéros de ligne et de fichier. Le curseur clignotant se trouve sur le symbole.

Si le fichier n'est pas ouvert, l'écran affiche un NOP et le fichier ne sera effectivement ouvert qu'après le stockage d'au moins une instruction.

### 6.1 Examen d'un programme

- Touches flèches : Affichage de la ligne précédente ↑ ou de la ligne suivante ↓
- Touche F2 : Permet d'afficher directement la ligne introduite au clavier
- Touche ESC : Retour à l'éditeur, invitation à choisir un autre fichier. Une nouvelle pression sur ESC est nécessaire pour retourner au menu de base.

### 6.2 Modification du Contenu d'une ligne

L'introduction d'une nouvelle ligne dans un fichier revient à modifier une ligne contenant NOP. Il suffit donc de décrire la modification d'une ligne.

- Touche ENTER: Déplace le curseur vers la droite sur le prochain argument. La ligne inférieure indique la nature de l'argument. Après le dernier argument, ou après l'instruction s'il n'y pas d'argument, la ligne est stockée et la ligne suivante est amenée à l'écran.
- Touche CLR : Déplace le curseur vers la gauche sur l'argument précédent. Pas d'effet lorsque le curseur se trouve sur l'instruction.
- Touche F5 : Si le curseur se trouve sur le symbole de l'instruction, F5 fait défiler de façon rotative et dans le sens croissant tous les choix correspondant à la position de curseur. Sur un argument, F5 est sans effet.

Pour entrer une instruction, il n'est pas nécessaire de faire défiler tous ceux possibles à l'aide de F5. Il est en général plus rapide de taper directement le code numérique correspondant. Avec la pratique, les codes se mémorisent facilement. Il faut remarquer qu'afin d'accélérer la frappe du code de l'instruction, l'entrée du nombre est automatique (sans Enter) après le deuxième chiffre. Pour utiliser F5, il faut dans ce cas reculer le curseur.

Il est naturellement possible de passer de l'examen à la modification de lignes et inversement à chaque instant.

**Exemple:** Entrer l'instruction POSR dans une ligne vide.

- Entrer 14 (sans ENTER), le symbole POSR est affiché, le curseur est sur l'argument "vitesse".
- Reculer le curseur avec CLR, puis presser F5 en observant le symbole; POSR arrive avec le code 14. Presser alors ENTER.
- L'affichage invite alors à choisir la vitesse présélectionnée (SEL.SPEED). L'argument suivant est le déplacement (DISP'MENT) qui sera entré en unités techniques.
- ENTER met l'instruction en mémoire et présente la ligne suivante pour édition.

On peut aussi corriger un seul argument d'une instruction en amenant le curseur sur cet argument. Après avoir entré le nouvel argument il faut presser ENTER jusqu'à ce que le curseur disparaisse à droite.

## 6.3 Insertion et Suppression d'une ligne

Une pression sur F3 insère une ligne à la position affichée.

**Exemple:** La ligne 12 est affichée et contient l'instruction WAIT. Après insertion, la ligne 12 contient un NOP et l'instruction WAIT est dans la ligne 13.

Une pression sur F4 supprime la ligne affichée ; la ligne qui la suivait prend son numéro et vient s'afficher.

## 6.4 Mise en Place du Marqueur de Pause

Le marqueur de pause peut être placé ou enlevé de l'instruction à n'importe quel stade de son édition en pressant F1. Le témoin de F1 indique la présence du marqueur. Pour modifier seulement l'état du marqueur il est nécessaire de quitter l'instruction par une ou plusieurs pressions sur ENTER.

# 7 Exécution des programmes

## 7.1 Menu "EXECUTION"

L'exécution d'un programme est contrôlée d'une part par les touches 'START' et 'STOP' (ou les entrées associées désignées par la configuration CTRL) et d'autre part par le mode défini dans le menu MODE. Lorsqu'un programme est exécuté, les fonctions sous F3 et F4 changent et deviennent "TRACE" et "I/O". L'affichage de la cote et de la vitesse réelle se fait sur la ligne supérieure, ainsi que le numéro d'outil courant (si il en existe). **Les cotes sont mesurées dans le référentiel courant.**

### 7.1.1 Fonctions des touches START, STOP

#### START

Si les témoins START et STOP sont éteints, le programme désigné sous "START PROGRAMME" dans "VECT" est exécuté.

Si le témoin STOP est allumé, le programme désigné sous "POWER ON PROGRAMME" est exécuté.

#### PAUSE

Le déroulement du programme peut être stoppé à la fin de l'instruction en cours en appuyant sur la touche 'F6'. Tous les mouvements sont cependant stoppés selon les rampes de décélération configurées, sans perdre leurs positions. Le témoin START clignote.

#### STOP

Si un cycle est en exécution, une première pression sur la touche 'STOP' avorte immédiatement tous les programmes en cours. Les sorties, la vitesse de broche sont remis à zéro. Les mouvements sont stoppés selon les rampes de décélération configurées, sans perdre leurs positions.

Une deuxième pression sur la touche 'STOP' conserve le témoin (LED) STOP allumé, indiquant que le prochain programme exécuté sera le programme d'initialisation "POWER ON PROGRAMME" dans "VECT".

A la mise sous tension de la commande E-100, le mode MOD1 est établi et le "POWER ON PROGRAMME" est exécuté sans aucune intervention. Si on ne désire pas de programme initial, on peut introduire 100 comme "POWER ON PROGRAMME".

### 7.1.2 Programmes à exécuter (start VECTors)

Par ce sous-menu, l'opérateur choisit le "POWER-ON-PROGRAMME". Ce programme est celui qui est exécuté automatiquement à la mise sous tension ou lorsque la LED du bouton STOP est allumée.

L'opérateur choisit également le "START PROGRAMME". Ce programme démarre à chaque



pression sur START si la LED STOP est éteinte (ou à chaque activation de l'entrée désignée dans le menu de configuration).

Il faut remarquer que le programme d'initialisation est exécuté à nouveau après un arrêt complet du programme et une pression de la touche STOP ou de l'entrée désignée. Si aucun programme d'initialisation n'est souhaité, on peut lui attribuer un fichier vide mais ouvert ou plus simplement le numéro de fichier 100.

### 7.1.3 "MODE", fonctions des modes MOD1, MOD2, SAT

#### Choix du mode d'exécution:

Le choix du mode d'exécution se fait au moyen des touches F1 à F3 et le mode courant est affiché sur la 2<sup>ème</sup> ligne.

- F1** Choix du mode SAT, saturation des vitesses de déplacement des axes.
- F2** Choix du mode MOD1, exécution normale, les marqueurs de pause ne sont pas pris en compte.
- F3** Choix du mode MOD2, le programme s'arrête avant l'exécution de l'instruction marquée.

#### MOD1 (F2):

Exécution normale, les marqueurs de pause ne sont pas pris en compte. Le témoin START est allumé.

#### MOD2 (F3):

Le marqueur de pause arrête le programme **avant** l'exécution de l'instruction marquée. Les programmes simultanés continuent sauf s'ils contiennent aussi des instructions marquées.

Pendant la pause, le témoin START clignote. Une pression sur 'START' redémarre l'exécution jusqu'à la prochaine instruction marquée. Ce mode est particulièrement utile en liaison avec la fonction TRACE.

#### SAT (F1):

Ce mode d'exécution offre l'avantage de saturer les vitesses de déplacement tout en respectant les points d'arrêt introduits en édition par la touche F1.

### 7.1.4 Utilitaire "TRACE"

Cet utilitaire n'est significatif que pendant l'exécution d'un programme. Il permet de visualiser l'instruction en cours d'exécution. UNIPROG+ étant capable d'un fonctionnement multi-tâches, il faut choisir le "programme simultané" (la tâche) que l'opérateur veut visualiser. La touche F1 permet le choix rotatif du programme simultané.

La ligne supérieure de l'écran affiche l'instruction sous le même format que l'éditeur. La ligne inférieure montre le programme simultané dans lequel se fait la trace, la ligne et le programme :

Cet affichage signifie que l'on trace dans le simultané 1, et l'instruction en cours est à la ligne 45 du programme 12.

### 7.1.5 Utilitaire "I/O" control

Cette fonction permet la lecture des états de toutes les entrées et sorties sous contrôle d'UNIPROG+.

Pour la signification et la numérotation des entrées et des sorties, se reporter au Tableau 5.1

- Les touches F3 et F4 déplacent le curseur respectivement sur IN et OUT. Pour changer le numéro d'une entrée ou d'une sortie, placer le curseur et entrer le numéro
- Selon le Tableau 5.1, les pseudo-I/O numéro 8, 9, 10 sont les états d'activation des programmes simultanés (paragraphe **Erreur! Source du renvoi introuvable.** Les pseudo-I/O 11...15 sont des drapeaux et les états IN et OUT coïncident.

## 7.2 Gestion des fautes

Une condition de faute est affichée par UNIPROG+ :

1. La faute générée par l'étage de puissance (voir 4.1),

Quand une faute est active, UNIPROG+ stoppe immédiatement les mouvements mais le sorties restent inchangées. L'écran affiche le message suivant:

DEFAULT AMPLI ou  
ARRET D'URGENCE: ESC

L'axe en défaut peut être déplacé lentement avec les touches de jogging afin de quitter les fins de courses. La touche <ESC> permet de revenir au menu initial. Si la condition de faute est toujours présente, l'arrêt de la commande est nécessaire.

Suite à tous les cas d'erreurs la commande exécutera le "POWER ON PROGRAMME".

#### ATTENTION:

Le bouton 'STOP', ainsi que l'entrée configurée comme STOP externe, ne peuvent pas être considérés comme arrêt d'urgence au sens des réglementations en vigueur.

## 8 Récapitulation UNIPROG +

### 8.1 Instructions

| Code | Instruction  | 1 <sup>er</sup> arg. | 2 <sup>ème</sup> arg. | Description                               | Page |
|------|--------------|----------------------|-----------------------|---|------|
| 91   | <b>ADDD</b>  | adresse              |                       | Addition Directe -> accu                  | 36   |
| 22   | <b>BRIN0</b> | entrée               | adresse               | Branche si entrée fausse                  | 29   |
| 23   | <b>BRIN1</b> | entrée               | adresse               | Branche si entrée vraie                   | 29   |
| 24   | <b>BRM</b>   | adresse              |                       | Branche si accu < 0                       | 35   |
| 27   | <b>BRNZ</b>  | adresse              |                       | Branche si accu <> 0                      | 35   |
| 25   | <b>BRP</b>   | adresse              |                       | Branche si Accu >= 0                      | 35   |
| 26   | <b>BRZ</b>   | adresse              |                       | Branche si Accu = 0                       | 35   |
| 61   | <b>CALL</b>  | adresse              |                       | Appel un sous-programme                   | 34   |
| 18   | <b>CLOS</b>  | vitesse              |                       | Test de fermeture (perte de pas)          | 27   |
| 95   | <b>CPL</b>   | N° sortie            |                       | Complémentaire (inverse) une sortie       | 31   |
| 59   | <b>DECD</b>  | adresse              |                       | Décrémentation Directe de ligne (adresse) | 32   |
| 79   | <b>DISPD</b> | Position             | adresse               | Affiche le contenu de l'adresse           | 31   |
| 94   | <b>DIVD</b>  | adresse              |                       | Division Directe -> accu                  | 36   |
| 62   | <b>END</b>   |                      |                       | Fin des programmes et des sous-programmes | 34   |
| 65   | <b>ENDRP</b> |                      |                       | Fin d'une boucle de répétition            | 35   |
| 98   | <b>FDATA</b> | réel                 |                       | Définition d'un nombre réel               | 37   |
| 50   | <b>FLOAD</b> | réel                 |                       | Charge Accu Immédiat, réel                | 32   |
| 99   | <b>IDATA</b> | Entier               |                       | Définition d'un nombre entier             | 37   |
| 51   | <b>ILOAD</b> | Entier               |                       | Charge Accu Immédiat, entier              | 32   |
| 58   | <b>INCD</b>  | adresse              |                       | Incrémentation Directe de ligne (adresse) | 32   |
| 60   | <b>JMP</b>   | adresse              |                       | Saut inconditionnel                       | 34   |
| 86   | <b>LDAC</b>  | n°DAC                |                       | Charge l'accu dans un DAC                 | 33   |
| 52   | <b>LOADD</b> | adresse              |                       | Charge accu Direct (adresse)              | 32   |
| 53   | <b>LOADI</b> | pointeur             |                       | Charge accu Indirect (pointeur)           | 32   |
| 85   | <b>MOTOR</b> | n° moteur            | Vitesse rot.          | Vitesse rotation moteur broche sur DAC(0) | 33   |
| 93   | <b>MULD</b>  | adresse              |                       | Multiplication Directe -> accu            | 36   |
| 54   | <b>NBINP</b> | position             | adresse               | Entrée d'un nombre                        | 33   |
| 90   | <b>NOP</b>   |                      |                       | No Operation                              | 37   |
| 28   | <b>OFF</b>   | Sortie               |                       | Déclenche une sortie                      | 29   |
| 29   | <b>ON</b>    | Sortie               |                       | Enclenche une sortie                      | 29   |
| 73   | <b>PADNW</b> | Vitesse              | adresse               | Positionnement Absolu Direct sans attente | 26   |
| 74   | <b>PAINW</b> | Vitesse              | pointeur              | Positionnement Absolu Indirect sans att.  | 26   |
| 72   | <b>PANW</b>  | Vitesse              | Coordon.              | Positionnement Absolu sans att.           | 26   |
| 84   | <b>PECK</b>  | Vit. lente           | Pos. Perc.            | Cycle de perçage                          | 27   |
| 68   | <b>PLOAD</b> |                      |                       | Position -> accu                          | 32   |
| 10   | <b>POSA</b>  | Vitesse              | Coordon.              | Positionnement Absolutus                  | 26   |
| 11   | <b>POSAD</b> | Vitesse              | adresse               | Positionnement Absolu Direct              | 26   |
| 12   | <b>POSAI</b> | Vitesse              | pointeur              | Positionnement Absolu Indirect            | 26   |
| 14   | <b>POSR</b>  | Vitesse              | Dépl.                 | Positionnement Relatif                    | 26   |
| 15   | <b>POSRD</b> | Vitesse              | adresse               | Positionnement Relatif Direct             | 26   |
| 16   | <b>POSRI</b> | Vitesse              | pointeur              | Positionnement Relatif Indirect           | 26   |

| Code | Instruction  | 1 <sup>er</sup> arg. | 2 <sup>ème</sup> arg. | Description                           | Page |
|------|--------------|----------------------|-----------------------|---------------------------------------|------|
| 76   | <b>PRDNW</b> | Vitesse              | adresse               | Positionnement Relatif Direct         | 26   |
| 77   | <b>PRINW</b> | Vitesse              | pointeur              | Positionnement Relatif Indirect       | 26   |
| 75   | <b>PRNW</b>  | Vitesse              | Dépl.                 | Positionnement Relatif sans attente   | 26   |
| 88   | <b>PRSTR</b> | position             | numéro                | Affichage de texte                    | 31   |
| 87   | <b>RADC</b>  | n°ADC                |                       | Lit un ADC -> accu                    | 33   |
| 80   | <b>RBW</b>   |                      |                       | Restore la fenêtre principale         | 31   |
| 69   | <b>RDATE</b> |                      |                       | Lecture de la Date -> accu            | 36   |
| 96   | <b>RDAY</b>  |                      |                       | Lecture du jour de la semaine -> accu | 36   |
| 17   | <b>REF</b>   |                      |                       | Prise de référence                    | 27   |
| 63   | <b>REP</b>   | nb de fois           |                       | Répétition                            | 35   |
| 64   | <b>REPD</b>  | Adresse              |                       | Répétition, Directe                   | 35   |
| 97   | <b>RTIME</b> |                      |                       | Lit l'heure -> accu                   | 36   |
| 83   | <b>SET</b>   | n° param             | Val. param.           | Attribution de variables diverses     | 27   |
| 67   | <b>SIM</b>   | numéro               | adresse               | Démarre un programme simultané        | 35   |
| 57   | <b>SPVEL</b> | n° DAC               | Rot/min               | Vitesse de broche sur DAC's           | 33   |
| 78   | <b>STOPM</b> |                      |                       | Stop le Mouvement                     | 26   |
| 55   | <b>STORD</b> | adresse              |                       | Sauvegarde accu, Directe              | 32   |
| 56   | <b>STORI</b> | pointeur             |                       | Sauvegarde accu, Indirecte            | 32   |
| 92   | <b>SUBD</b>  | adresse              |                       | Soustraction Directe -> accu          | 36   |
| 19   | <b>TOOL</b>  | N° outil             |                       | Définition référentiel outil          | 26   |
| 81   | <b>TPING</b> | Pas                  | Pos. fin              | Définition d'un taraudage             | 28   |
| 70   | <b>WAIT</b>  | Temps                |                       | Temporisation en secondes             | 35   |
| 20   | <b>WAIT0</b> | Entrée               |                       | Attente si entrée = 0                 | 29   |
| 21   | <b>WAIT1</b> | Entrée               |                       | Attente si entrée = 1                 | 29   |
| 71   | <b>WAITD</b> | adresse              |                       | Temporisation Directe                 | 35   |

## 8.2 Entrées-Sorties

| Input  | Item                   | Output | Item                   |
|--------|------------------------|--------|------------------------|
| 0      | IN(0)                  | 0      | OUT(0)                 |
| 1      | IN(1)                  | 1      | OUT(1)                 |
| 2      | IN(2)                  | 2      | OUT(2)                 |
| 3      | IN(3)                  | 3      | OUT(3)                 |
| 4      | IN(4)                  | 4      | OUT(4)                 |
| 5      | IN(5)                  | 5      | OUT(5)                 |
| 6      | IN(6)                  | 6      | OUT(6)                 |
| 7      | IN(7)                  | 7      | OUT(7)                 |
| 8      | SIM(0)                 | 8      | SIM(0)                 |
| 9      | SIM(1)                 | 9      | SIM(1)                 |
| 10     | SIM(2)                 | 0      | SIM(2)                 |
| 11     | FLAG(1)                | 11     | FLAG(1)                |
| 12     | FLAG(2)                | 12     | FLAG(2)                |
| 13     | FLAG(3)                | 13     | FLAG(3)                |
| 14     | FLAG(4)                | 14     | FLAG(4)                |
| 15     | FLAG(5)                | 15     | FLAG(5)                |
| 16..55 | IN(16..55)             | 16..55 | OUT(16..56)            |
| 56     | Boost (enable Yaskawa) | 56     | Boost (enable Yaskawa) |
| 58     | RUN                    | 58     | RUN                    |
| 60     | INA                    |        |                        |
| 61     | INB                    |        |                        |