



# PROGRAMME STANDARD **ABB**

## APPLICATION PSA

## NOTICE UTILISATEUR

AUTEURS:

O.NIZON

**ABB MC**

A.DODEMONT

**ABB MC**

VERSION: 3.7\_6  
DATE : 06.02.2006

**ABB MC**

## SOMMAIRE

<b>I - PRESENTATION.....</b>	<b>7</b>
<b>II - INSTALLATION ET CONFIGURATION .....</b>	<b>12</b>
1 - Procédure d'installation initiale: .....	12
2 - Premier Démarrage du Programme:.....	14
3 - Sauvegarde des programmes et données:.....	18
4 - Procédure de Rechargement à froid (Cold Start): .....	22
a - Rechargement à partir de la disquette Robot: .....	.22
b - Rechargement à partir d'un PC distant:.....	.22
5 - Mises à jour du standard et redémarrage (P-START): .....	22
<b>III - PROGRAMMATION.....</b>	<b>24</b>
1 – Création des points remarquables.....	24
2 – Création des trajectoires de Service.....	26
3 – Création des trajectoires de Travail.....	29
4 – Création et test des Cycles, gestion des Hors Zone .....	33
5 - Interverouillages entre robots:.....	39
6 – Création des trajectoires de Reprise .....	41
7 – Création des trajectoires de substitution .....	45
8 – Exécution de routines sur point de passage.....	46
9 – Mise au point et Optimisation.....	47
a - Mise au point: .....	.47
b - Optimisation de la mémoire:.....	.48
c - Optimisation des temps de cycle:.....	.48
10 – Gestion des cames de protection Opérateur.....	49
<b>IV – COMMANDES MANUELLES .....</b>	<b>51</b>
1 – Si le robot est en mode Manuel et Hors asservissements : .....	51
2 – Si le robot est en mode Automatique : .....	53
<b>V – MENU REGLAGES.....</b>	<b>56</b>
1 - Substitution:.....	56
2 - Surveillances: .....	57
3 - Calibration / Definition des Repères: .....	58
a- Calibration .....	.58
b - Définition des Repères:.....	.58
4 - Rechargement des actions manuelles: .....	58
<b>VI - CAS DU CHANGEUR D ' Outils.....</b>	<b>60</b>
Trajectoires de service, demandes de rodage:.....	64
1- Sur demande de l'opérateur.....	.64
2 - Automatique .....	.65
3 – Programmation.....	.65

Détections de changement d'outil:.....	<b>66</b>
En cas de panne d'un outil :.....	<b>67</b>
Reprises de trajectoire: .....	<b>67</b>
<b>VII - FONCTIONNEMENT EN PRODUCTION</b> .....	<b>69</b>
1 - Surveillance Communication Automate : .....	<b>72</b>
2 – Procédure d'Arret Cycle .....	<b>73</b>
<b>VIII - DEMANDES DE SERVICE</b> .....	<b>74</b>
<b>IX - DETAIL DES DIFFERENTS PROCESS</b> .....	<b>76</b>
<b>1 - Fonctions communes à tous les process:</b> .....	<b>76</b>
a- Gestion du mode Avec/Sans Piece et des clés Avec/Sans Process: .....	.76
b - Gestion du redémarrage après arrêt: .....	.76
<b>2 – Encollage Debit (contrôle par Volucompteur)</b> .....	<b>77</b>
Purge à tout moment .....	.80
<b>3 - Encollage Billes de Verre et Pulvérisation (contrôle par Doseur)</b> .....	<b>81</b>
<b>4 – Encollage Mastic (contrôle par Pressions)</b> .....	<b>83</b>
Purge à tout moment : .....	.85
Fonction Test: .....	.85
Constantes modifiables: .....	.85
<b>Fonctions communes aux applications Pression, Volucompteur et Doseur:</b> .....	<b>86</b>
Defaut Groupe Non Bloquant: .....	.86
Defaut Groupe Bloquant: .....	.86
Présence Pistolet(s) .....	.87
Trajectoires de Presentation .....	.87
Mode Test Colle:.....	.88
<b>4 - Clinchage.....</b>	<b>89</b>
Actions manuelles:.....	.91
<b>5 – Soudure SR Pneumatique .....</b>	<b>91</b>
Contrôle Dec .....	.92
Rodage : .....	.95
Rodage avec contrôle de vitesse (AMDP ou EXROD):.....	.96
Changement Electrodes .....	.97
Cas de la soudure sur fausse pince : .....	.97
Constantes modifiables: .....	.98
<b>6 – Soudure SR Electrique .....</b>	<b>100</b>
6.1: Avec séquenceur interne (swit) .....	.100
6.2: Avec séquenceur externe (Aro).....	.101
6.3: Cas de la baie S4CPLUS AUTOMOTIVE: .....	.105
<b>7 – Changements d'électrodes .....</b>	<b>105</b>
a) Contrôle de brillance.....	.105
b) Changeur automatique d'électrodes.....	.107
<b>8 - Goujonnage .....</b>	<b>109</b>
<b>9 - Manutention .....</b>	<b>110</b>
<b>10 – Soudure ARC :</b> .....	<b>113</b>
Balayage :.....	.116
Gestion des reprises après arrêt : .....	.120
Décrassage buse : .....	.121
Trajectoire de contrôle : .....	.121
Actions manuelles : .....	.121

11 – Rivetage électrique : .....	122
12 – Depalettisation: .....	123
13 -Palettisation: .....	129
<b>X - HISTORIQUE DES DEFAUTS.....</b>	<b>134</b>
<b>XI - FONCTIONNEMENT SOUS DESKWARE .....</b>	<b>136</b>
<b>XII - LISTE DES ROUTINES ACCESSIBLES AU PROGRAMMEUR.....</b>	<b>138</b>
<b>XIII - LISTE DES MESSAGES D'ERREUR .....</b>	<b>142</b>
<b>XIV - LISTE DES MODULES.....</b>	<b>146</b>
1 - Liste des modules utilisés uniquement par l'avant-plan:.....	146
2 - Liste des modules partagés entre l'avant et l'arrière plan: .....	148
3 - Liste des modules utilisés uniquement par l'arrière-plan: .....	149
<b>XV – FONCTIONNEMENT DES SORTIES VERS L'AUTOMATE .....</b>	<b>151</b>
<b>XVI – FICHIER LISEZ-MOI DE LA DISQUETTE .....</b>	<b>154</b>

Pour information

<b>PRINCIPALES EVOLUTIONS</b>	<b>VERSION</b>
Nouveau process encollage billes de verre	3.3
Nouveau process pince électrique et séquenceur externe	3.4
Gestion des rodeuses avec contrôle AMDP	3.4
Gestion Changeur Automatique d'Electrodes SCIAKY	3.4
Modification fonctionnelle Attends_PP et Attends_Entrees	3.4
Modification gestion affichages sur arrêt programme avant validation	3.4
Prise en compte langue Espagnole	3.4
Vitesse d'arrière plan modifiable	3.4
Création fonction Teste lampes	3.4
Affichage du nom du point process	3.4
Nouvelle gestion de la sortie do_SA_Robot	3.4
Nouveau process Doseur Simple et double pistolet (A 8)	3.6
Nouveau process Palettisation	3.6
Nouveau process double swit double ARO	3.6
Evolution majeure process Depalettisation	3.6
Nouveau process soudure en S4C+ sans carte swit	3.6
Changement automatique de l'AVP	3.6
Préparation pour mises à jour automatiques du standard	3.6
Modification du principe de changement electrodes pour les applis double pince	3.6
Tous les textes sont maintenant regroupés (provisoirement) dans 1 seul fichier	3.6
Mise à jour automatique du standard	3.67
Nouveau process soudure S4C+ Automotive	3.68
Séparation des fichiers texte pour gain de mémoire	3.68
Fusion des différents modules soudure pour gain place disquette	3.68
Gestion de 3 têtes goujon sur changeur	3.68
Gestion de 7 outils différents sur changeur	3.68
Gestion des pinces pneumatiques à basculement.	3.68
Gestion d'une deuxième rodeuse.	3.68
Gestion des zones "consommées" dans un cycle lors d'une reprise.	3.68
Possibilité d'affecter les noms des robots pour la substitution + gestion de 10 robots substituables différents.	3.68
Reprise "intelligente" pour les process réputés non reprenables.	3.68
Possibilité de choisir le mode de redémarrage suite à perte communication automate.	3.68
Affichage de l'état de la supervision de mouvement par appui sur P1 en auto.	3.68
Modification de l'arborescence sur le PC distant	3.68
Nombreuses autres évolutions mineures, voir le LISEZMOI.doc	3.68
Possibilité de reprises de trajets même avec changeur d'outils	3.69

Nouveau changeur d'électrodes EXROD	3.69
Création gestion double équilibrage pour pinces en C ou GP	3.69_5
ARC : ajout paramètre optionnel POSTGAZ dans la routine Fin_Cordon	3.69_5
Gestion des changeurs d'électrodes	3.69_5
Gestion de l'arrêt cycle	3.69_6
Ajout appel automatique des trajs de présentation colle	3.7
Validation du système de rodage avec abattants électriques	3.7
Validation du Contrôle Alignement Electrodes	3.72
Ajout d'un message pour prévenir le recul de l'abattant suite à un spécial 3	3.74
Ajout de la possibilité de faire une demande d'arrêt en fin de chaque process.	3.74
Ajout des fichiers pour l'installation des modifications sur les BACK_ACTIONS	3.76

Pour information

## I - PRESENTATION

Le standard T5 constitue une évolution importante par rapport au standard T1 / N6. Toutefois, il s'en inspire puisque toutes les fonctionnalités jugées "bonnes" ont été reconduites.

Dans un but de diminution des coûts de mise en route, de programmation et de maintenance, l'un des objectifs a consisté à simplifier au maximum la partie "visible" et modifiable du programme, quitte à augmenter sensiblement la partie "cachée" et non modifiable, sauf cas vraiment exceptionnel.

Le programme permet de masquer presque complètement l'utilisation des menus de base du système.

Le travail de l'intégrateur, du trajectoriste et du technicien de maintenance s'en trouvent considérablement simplifiés, tout en conservant et en enrichissant les fonctionnalités du standard précédent.

Dans sa version 3.7, il comporte:

- 18 métiers différents (appelés "process") mixables entre eux. (sauf rares exceptions).
- Plusieurs options pour chaque métier (soudure 1 pince ou soudure 2 pinces...).
- Plusieurs systèmes de communication (Profibus, FipIO, Ethernet...).
- De nombreuses configurations d'entrées sorties.

Les principaux process supportés sont:

- Manutention.
- Soudure pneumatique 1 pince (fixe ou embarquée).
- Soudure pneumatique 2 pinces (fixes ou embarquées).
- Soudure électrique avec carte swit.
- Soudure électrique 1 ou 2 pinces avec séquenceur externe (ARO).
- Soudure électrique 1 pince sans carte swit et avec séquenceur externe.
- Clinchage 1 ou 2 pinces avec différents procédés de mesure.
- Rivetage électrique.
- Soudure Goujons 1 ou 2 têtes.
- Soudure Arc.
- Dépalettisation.
- Palettisation.
- Encollage 1 ou 2 pistolets avec contrôle par pression (mastic) et autotest automatique.
- Encollage 1 ou 2 pistolets avec contrôle par volucompteur (Débit).
- Encollage 1 pistolet avec contrôle par doseur. (ou 2 pistolets simultanés à partir de A 8).
- Encollage 1 pistolet par pulvérisation (variante).

- Changeur automatique d'outils (2 types différents).
- Changeur automatique d'Electrodes. (3 types différents)

Lorsqu'une pince est fixe au sol, le rodage peut être effectué en temps masqué grâce à une rodeuse montée sur un abattant.

Chaque process de base possède généralement plusieurs variantes. Par exemple, le process "Pince Electrique" regroupe:

- 1 pince électrique avec carte swit.
- 1 pince électrique avec carte swit et changeur automatique d'électrodes.
- 1 pince électrique avec sequenceur externe.(ARO)
- 1 pince électrique avec sequenceur externe et changeur d'électrodes.
- 2 pinces électriques avec 2 sequenceurs externes.
- 2 pinces électriques dont 1 embarquée avec 2 sequenceurs externes et 1 changeur d'électrodes.
- 3 pinces électriques avec changeur d'outils.(sans changeur d'électrodes).
- 1 pince électrique sans carte Swit (S4C+).

Soit pas moins de 8 variantes pour ce seul process "de base".

Le produit de toutes ces possibilités représente plusieurs milliers de configurations différentes possibles, bien que le programme soit toujours le même. Naturellement, seuls les modules correspondant aux métiers déclarés sont réellement chargés en mémoire.

L'interface utilisateur se compose de menus déroulants simples ayant chacun au maximum 4 choix possibles. Le programmeur a juste à naviguer dans les menus pour réaliser toute sa programmation.

Chaque menu est doté d'une touche "QUITTER" qui permet de revenir au menu immédiatement précédent. Seuls certains menus pour lesquels un choix est obligatoire ne proposent pas d'option "QUITTER".

Excepté pour des cas très particuliers, il n'est pas nécessaire de bien connaître la baie S4, ni même, à la limite, le langage RAPID. Il suffit de maîtriser les notions de cycle et de trajectoires, comme décrit ci-dessous, ainsi que les règles de fonctionnement propres à PSA, et de savoir créer des points et des repères.

Sauf exception, il ne sera donc pas fait référence à l'utilisation des menus du système, qui ne sont jamais utilisés en exploitation normale.

#### NOTIONS FONDAMENTALES:

Le programme standard ne s'arrête jamais : il est rebouclé sur lui-même. Il est toujours en exécution, sauf si l'on fait un STOP.

Le principe d'utilisation est simple : on fait toujours « EXEC », en auto ou en manuel. Lorsque une donnée est nécessaire, le programme s'arrête de lui-même à l'endroit où la donnée est requise. Il vous suffit de la documenter et de faire « EXEC ».

**Il est inutile et fortement déconseillé d'utiliser les menus système « Aller à Routine » ou « Voir Routine ».**

Le programme peut être soit En Production, soit Hors Production. Le passage de l'un à l'autre se fait en appuyant sur les boutons « Départ » et « Repli ».

Le robot, quand à lui, peut être soit en mode Automatique, soit en mode Manuel. Lorsqu'on quitte la production en appuyant sur Repli, le programme revient au menu principal, mais ne s'arrête pas. On peut alors soit relancer en production, soit naviguer dans les menus.

Toutes les fonctions d'apprentissage ou de réglages ne sont accessibles que Hors Production. Certaines peuvent être exécutées en Auto, d'autres exigent un passage en Manuel pour des raisons de sécurité.

Lorsque le robot est « En Production », il reçoit un code programme de l'automate. Ce code corresponds à l'ensemble des travaux à réaliser sur une pièce. Exemple:

- Approche d'un montage.
- Prise d'une pièce.
- Encollage sous un pistolet.
- Dépose sur un poste de mariage.
- Soudure de quelques points (ou cordons en soudage arc).
- Reprise de la pièce.
- Dépose sur un convoyeur d'évacuation.
- Retour sur un point de dégagement et attente d'un nouveau code programme.

Chacun de ces travaux élémentaires s'appelle une **Trajectoire**, et l'ensemble constitue un **Cycle**.

Sur le standard T5, au maximum 255 codes cycles différents (31 par défaut) peuvent être reçus de l'automate. C'est pourquoi ces 31 cycles sont **déjà créés** en mémoire, mais ne contiennent aucune trajectoire. Vous n'avez donc pas à les créer vous même, il faudra simplement les remplir.

**ATTENTION:** Le cycle n° 1 corresponds au code reçu n° 1, le 2 au 2, et ainsi de suite. Il n'est pas possible d'exécuter le cycle 4 si on a reçu le code programme 12. Le numéro de cycle est imposé par l'automate.

Le nombre de trajectoires n'est en théorie pas limité. Toutefois, pour ne pas utiliser de la mémoire inutilement, le nombre maxi de trajectoires possibles est limité à 50 (modifiable dans la constante *Nb\_Traj\_Max*), et le nombre de trajectoires pré-existantes est de 10. (de Traj1 à Traj10). Si ce nombre est insuffisant, il faudra en créer de nouvelles en utilisant le menu "DUPLIQUER" à partir d'une trajectoire existante.

Le nombre maximum de cycles possibles est également modifiable en changeant la valeur de la constante *Nb\_Cycle\_Max*. Si vous changez cette valeur, pensez à adapter la taille du groupe d'entrées de dialogue automate (*gi\_code\_cyc*).

Il est judicieux de positionner ces variables au nombre de cycles/trajectoires réellement utilisés: ainsi l'affichage sera plus rapide, et cela évite de remplir les archives avec le message "Référence à une donnée inconnue: TrajXX". De plus, cela permet de créer des trajectoires "de test", par exemple, qui deviennent "invisibles". Ex: Si on utilise 10 trajectoires de travail, mettre *Nb\_Traj\_Max* à 10. On peut alors créer une trajectoire 23, celle ci ne sera jamais accessible par le menu "Apprentissage". Mais elle est exécutable par le menu "Autre Routine".

On verra un peu plus loin qu'en réalité, le numéro de trajectoire n'a besoin d'être connu qu'à une seule occasion. Le reste du temps, c'est la description de ce qu'elle fait, en texte clair, qui est utilisée et présentée. Par exemple:

- "Soudure coté droit du renfort Pavillon Z8."

Naturellement, une même trajectoire peut être utilisée dans des cycles différents, ce qui évite de la dupliquer et de recopier les éventuelles modifications.

Un **cycle** peut appeler autant de **trajectoires** que bon lui semble, et de n'importe quel numéro.

Chaque trajectoire peut être affectée d'un numéro de zone correspondant à une sécurité automate. Par exemple, on pourra décrire que la prise de pièce se fait dans la zone 4, et que la dépose se fait dans la zone 2, en accord avec l'automatien. A partir du moment où ce numéro est défini, la gestion des accès zone, libération et surveillance est automatique, sans programmation par le trajectoiriste.

Concernant les trajectoires de reprise, leur gestion est automatique. Le programmeur n'a aucune ligne de "programme" à écrire. Il doit simplement créer les points des trajectoires de reprise, comme expliqué au chapitre "CREATION DES TRAJECTOIRES DE REPRISE".

Pour ceux qui connaissaient le standard T1 ou N6, on peut citer les évolutions majeures suivantes:

- Tous les modules sont "système", donc conservés en mémoire sur "Fermer Programme".
- Le fichier TRAAUTO.MOD s'appelle désormais TRAJECT.SYS.
- Les routines standard ne sont pas modifiables.
- La routine "AUTOSEC" n'existe plus.
- La routine "SELECT\_REPRISE" n'existe plus.
- Le fichier "TRAJECT" est le seul modifiable par l'utilisateur (excepté la configuration des actions manuelles). Il contient exclusivement les cycles, trajectoires et données des trajectoires.

- Les positions remarquables (PEO, REPLI,...) sont de type JointTarget, donc **indépendantes** de l'outil utilisé. Seules les données de masse et d'inertie sont prises en compte pour le mouvement.
- Tous les modules standard sont strictement identiques quelque soient les applications installées. Seuls les fichiers de déclaration d'entrées/sorties diffèrent.
- Même si tous les modules de tous les process sont présents sur la disquette d'installation, seuls les modules correspondant au process déclaré sont effectivement chargés en mémoire.
- Seules les données propres à un robot considéré sont sauvegardées en Ramdisk ou sur une disquette spécifique au robot. Les modules standard, communs et identiques à tous les robots, ne sont jamais sauvegardés. Ils n'existent que sur la disquette "STANDARD T5" de la version courante.
- Toutes les applications "métier" sont compatibles entre elles, sans modification. Il est ainsi possible de mixer toutes les variantes imaginables, à l'exception du cas "Soudure pneumatique" incompatible avec le cas "Soudure électrique". On peut ainsi déclarer qu'un robot possède 2 préhenseurs, 1 pince à clincher, avec également 1 pince de soudure, le tout porté par le robot grâce à un changeur d'outil, et 1 pistolet d'encollage mastic et 1 pistolet d'encollage Débit. Soit, dans cet exemple un peu exagéré, 6 process différents simultanés. Attention toutefois: la compatibilité est garantie au niveau du programme, mais l'affectation des entrées sorties n'existe pas pour tous les cas de figure.
- L'installation et la configuration initiale du logiciel s'effectuent grâce à un programme d'installation fourni. Une installation manuelle n'est plus possible, car trop complexe.
  - Le rechargement du logiciel suite à un « froid » se fait de façon automatique et extrêmement simple.
  - Un "écran de production", rafraîchi dès que nécessaire, informe l'opérateur de tout ce qui se passe en cours de production: Alarmes, défauts, position du robot, attente d'événement, temps de cycle...
  - Un fichier Historique permet de retrouver tous les défauts apparus, avec la date et l'heure.
  - La mise à jour du standard s'effectue de façon quasi-automatique.
- Et bien d'autres choses encore, que nous vous invitons à découvrir...

## II - INSTALLATION ET CONFIGURATION

Cette action n'est à faire **qu'une fois dans la vie de l'installation**. On suppose que le démarrage de la baie a été fait avec succès. (Installation et démarrage de ROBOTWARE 3.1 ou supérieur).

On se retrouve donc avec un robot "vierge" de tout programme, mais prêt à être utilisé pour la première fois.

Cette procédure concerne essentiellement l'intégrateur. Elle ne sera jamais utilisée après une première installation, et **surtout pas** après un redémarrage "à froid" sur un robot ayant déjà produit.

Dans ce cas, on se reportera au chapitre "*Procédure de rechargement*".

Elle peut également être utilisée lors du changement de configuration d'un site. Exemple: ajout d'un deuxième pistolet d'encollage, ou passage d'une application 1 pince à 2 pinces. Dans ce cas, le programme propose d'effacer ou non les anciennes trajectoires, qui peuvent donc être récupérées. Toutefois, leur totale compatibilité avec la nouvelle configuration ne peut être garantie qu'en cas d'ajout de nouvelles applications, pas en cas de suppression.

### **1 - Procédure d'installation initiale:**

- 1 - Mettez la disquette intitulée "STANDARD PSA " dans le lecteur.
- 2 - Par la fenêtre "FICHIER", "OUVRIR", chargez le programme "INSTALLE" de la disquette.
- 3 - Faites "EXEC", éventuellement en mode "SIMULATION", pour ne pas avoir à maintenir la gâchette enfoncée. Le programme démarre et affiche le logo ABB.
- 4 - Choisissez le type de communication avec l'automate: par réseau Profibus, FipIO ou par E/S parallèles.
- 5 - Sur l'invite du menu, choisissez les applications concernées par ce robot.



Les choix « SUIV » et « PREC » permettent de faire défiler la liste des process.

Pour chaque process demandé, une confirmation permet d'installer ou de désinstaller.

Lorsque le choix des process est terminé, le logiciel vérifie leurs dépendances et compatibilités. Si une erreur est détectée, un message s'affiche, proposant de recommencer l'opération.

Il est par exemple interdit de cumuler la soudure pneumatique et la soudure avec pince électrique.

Si les choix sont valides, le logiciel propose de configurer de façon plus fine chaque type de process installé. Par exemple, dans le cas de l'encollage Mastic, y a t'il 1 ou 2 pistolets ? Si une pince de soudure est déclarée, est elle avec ou sans basculement ? Simple course ou double course ? Etc...

Après avoir répondu à toutes ces questions embarrassantes, et ce pour chaque process, le programme d'installation crée un fichier "CONFIGUR.SYS" sur le Ramdisk, puis copie les fichiers minimaux de démarrage (PSA.prg, ACTMSITE.sys, ACTUSITE.sys, TRAJECT.sys) de la disquette vers le Ramdisk. Soit 5 fichiers en tout. Notez que vous n'avez jamais besoin de passer par le Gestionnaire de Fichiers pour faire cette copie à la main.

Le programme crée alors les fichiers de paramètres système *sys.cfg*, *mmc.cfg* et *eio.cfg*. Il ne peut toutefois pas les charger en mémoire. A l'invite du message "Chargez les paramètres système", allez dans la fenêtre "Paramètres système", validez "Charger paramètres". Sélectionnez le support:RAMDISK.



Puis les fichiers « **A1** », « **A2** » et « **A3** ».

(Ce nom leur est donné pour qu'ils apparaissent en premier dans la liste.)

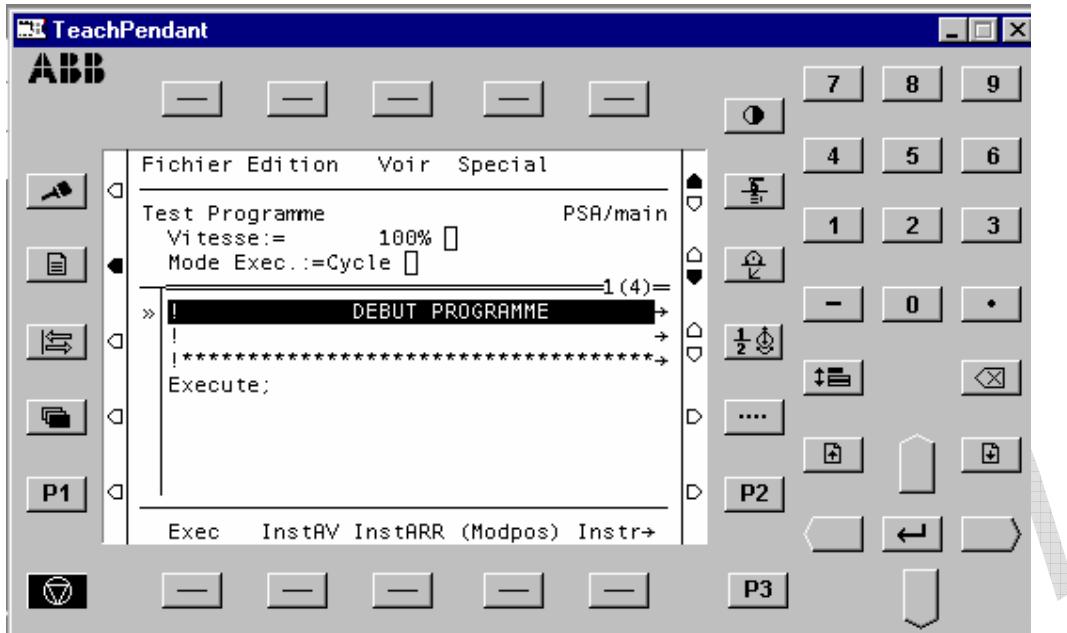
Faites alors un P-START (2 5 8 par le menu Maintenance).

La baie initialise les tâches et charge le programme d'arrière plan et le programme d'avant plan. S'il n'y a pas d'erreur, le message "Enlevez la disquette standard" s'affiche.

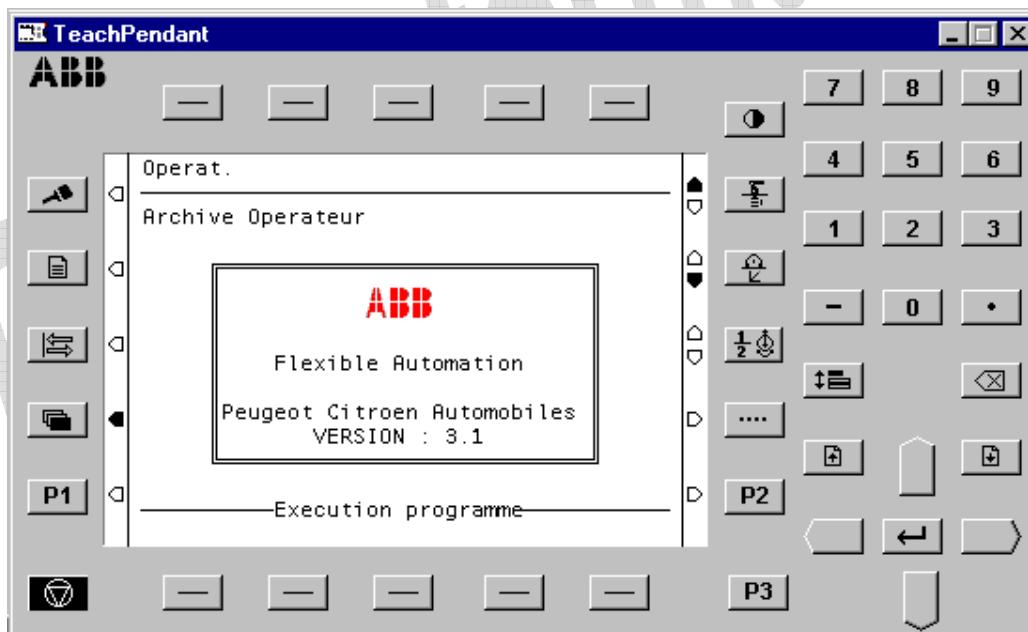
Retirez la du lecteur.

## 2 - Premier Démarrage du Programme:

Après le P-START, tous les modules de toutes les tâches sont chargés. L'écran doit alors afficher (après avoir enlevé la disquette):

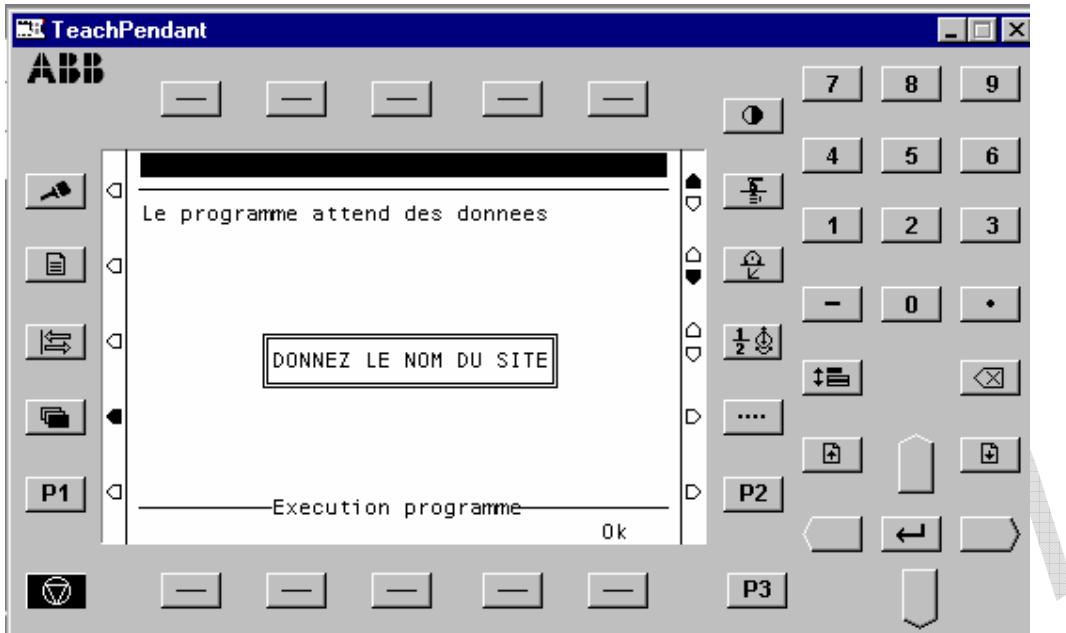


Appuyez sur Exec en maintenant la gâchette. Le programme se lance et affiche le numéro de version du Standard (3.1 dans cet exemple):

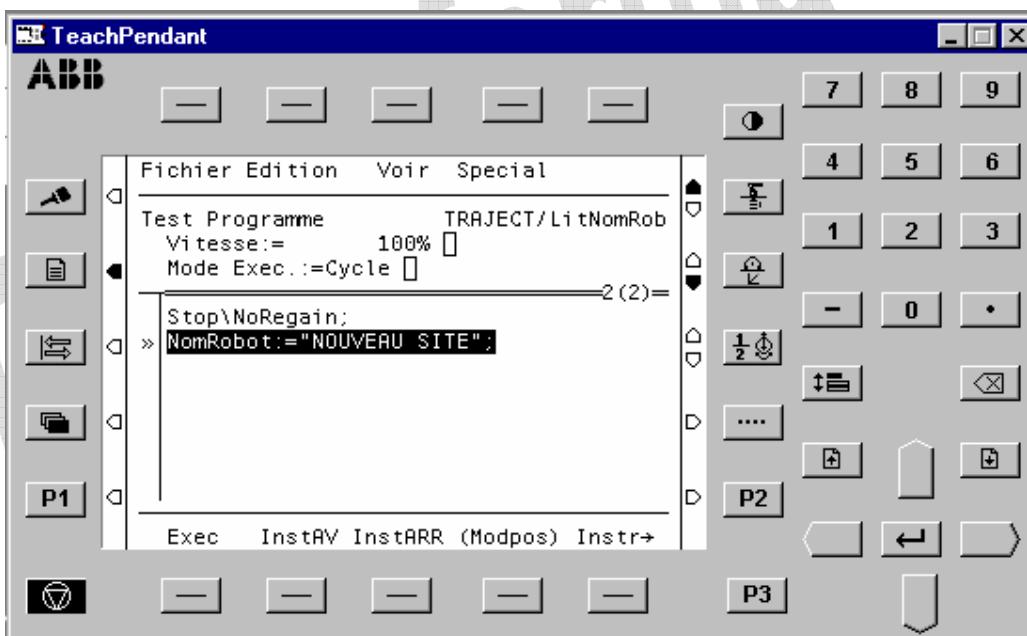


Le programme refuse de démarrer tant que la disquette du standard est dans le lecteur.

Le programme vous demande de donner un nom au site:



Appuyez sur OK. Le programme s'arrête sur la ligne:

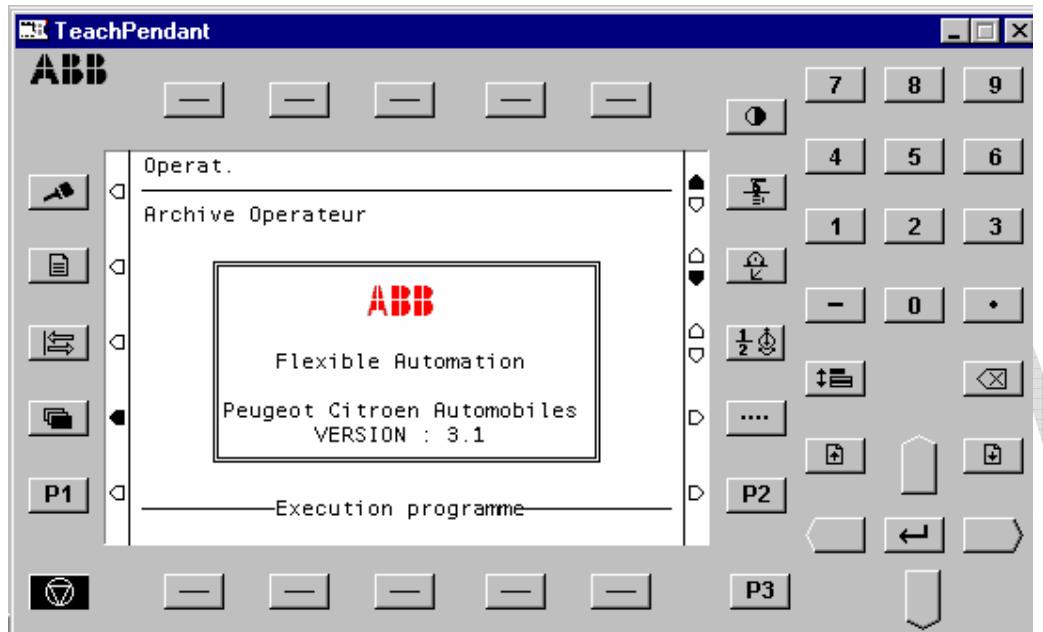


Déplacez le curseur vers la droite et appuyez sur RETURN. Vous pouvez alors entrer un nom pour la donnée « NomRobot ».

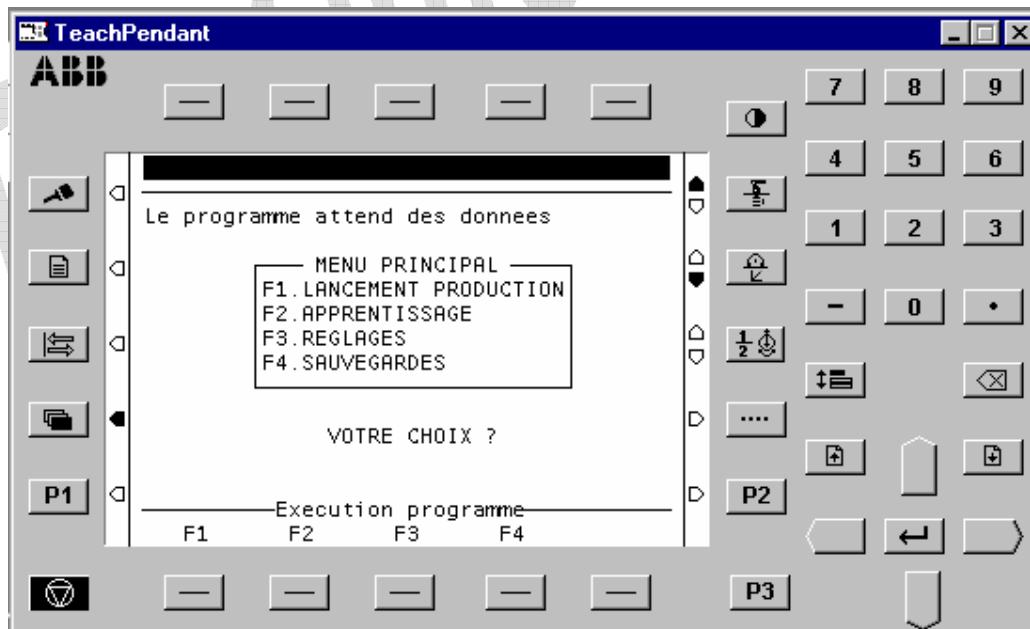
Ce nom permettra d'identifier le fichier Trajectoires, ainsi que le fichier de suivi. Comme il s'agira d'un nom de fichier, il est limité à 8 caractères sans caractères spéciaux et sans espaces. Le standard tronque automatiquement le nom s'il est trop long, mais il ne supprime pas les caractères interdits.

Il est judicieux de prendre comme nom le numéro du robot, par exemple « R\_2354 ».

Profitez en pour désactiver le mode Simulation, et relancez le programme en exécution: le logo de présentation s'affiche :



Puis le menu principal apparaît, sauf si vous avez fait un départ distant en mode Automatique. En effet, dans ce cas, le programme passe immédiatement en mode Production.



Si vous êtes ennuyé par les surveillances d'outil parce que ceux ci ne sont pas encore câblés, et qu'un défaut apparaît en permanence, repassez en simulation pour pouvoir lancer le programme, puis dans le menu "REGLAGES","SURVEILLANCES", désactivez les surveillances gênantes. La prise en compte est immédiate.

### **3 - Sauvegarde des programmes et données:**

Seuls 4 modules sont susceptibles d'être modifiés par l'utilisateur: les trajectoires et données des trajectoires (TRAJECT.SYS), la configuration du site (CONFIGUR.SYS), et les actions manuelles (ACTMSITE.SYS et ACTUSITE.SYS).

Vous pouvez très simplement sauvegarder l'ensemble de ces modules en validant le menu "Sauvegardes", "Sauvegarde des Modifs", à partir du menu principal:



Vous pouvez choisir de sauvegarder soit sur le ramdisk, soit sur une disquette, soit encore directement sur un PC distant si l'option Ethernet avec NFS est installée. Attention: le support NFS s'appelle "S4boot/" par défaut.. Vous pouvez modifier le nom du répertoire distant dans la constante « *chemin\_NFS* », par exemple « *Cote\_de\_Caisse*:».

#### **ATTENTION: Modification de structure depuis la version 3.68**

Le répertoire AppliXYZ du PC est maintenant le reflet exact de la disquette d'installation, il suffit donc de recopier la disquette dans le répertoire AppliXYZ, ce qui est plus simple que dans les versions précédentes.

Si la version porte le nom V3.68, le répertoire doit s'appeler Appli368 (**Pas de point**).

L'arborescence des répertoires sur votre PC doit être la suivante:

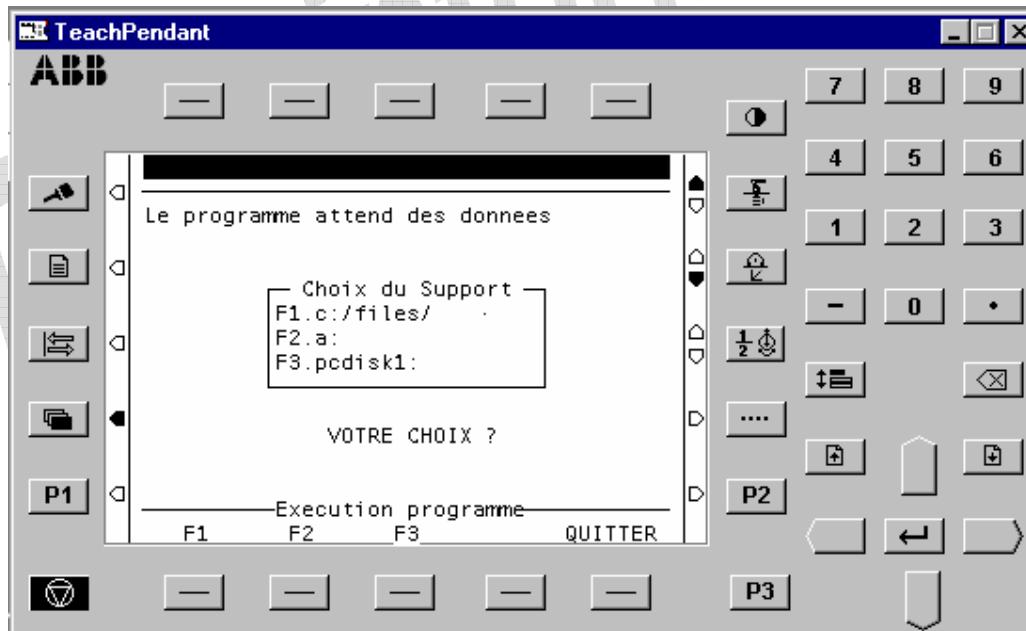
```
C:/S4boot
  /AppliXYZ
    → /Appli
      → modules du standard de la version X.Y.Z
    -> /Textes
      → /Français ou Anglais ou Espagnol
        → modules de textes de chaque langue
  /Robot_1
    → modules spécifiques de Robot_1.
  /Robot_2
    → modules spécifiques de Robot_2.
```

**Attention:** le nom du répertoire *Robot\_1* doit être identique au nom donné au robot (string *NomRobot*) lors de l'installation.

De plus, le contenu de la chaîne *Chemin\_Nfs* doit être **strictement identique** au nom déclaré dans les paramètres système dans le champ **LocalPath** pour la communication par Ethernet.

Ce nom doit, de plus, se terminer par le caractère ":". Ex: "Pcdisk1:", ou "PC\_Distant:"...

Le champ **ServerPath** déclaré dans les paramètres système doit contenir la chaîne "c:\S4boot" sans anti-slash à la fin.



Choisissez le support de sauvegarde. **Tous les fichiers propres au robot** sont sauvegardés. Si ces fichiers existaient déjà sur le support, ils sont écrasés.

(Nota : les noms de répertoires ci-dessus correspondent à une utilisation sous DESKWARE. Sur un robot réel, vous avez les choix Ramdisk, Flp1,Pcdisk1)

Depuis la version 3.1, une sauvegarde sur Ramdisk entraîne automatiquement une sauvegarde sur le réseau, et inversement. Ainsi, les deux supports sont systématiquement à jour. Le temps d'une sauvegarde complète est de 2 à 3 secondes, selon la taille du fichier Trajectoires.

Les modules sauvegardés sont:

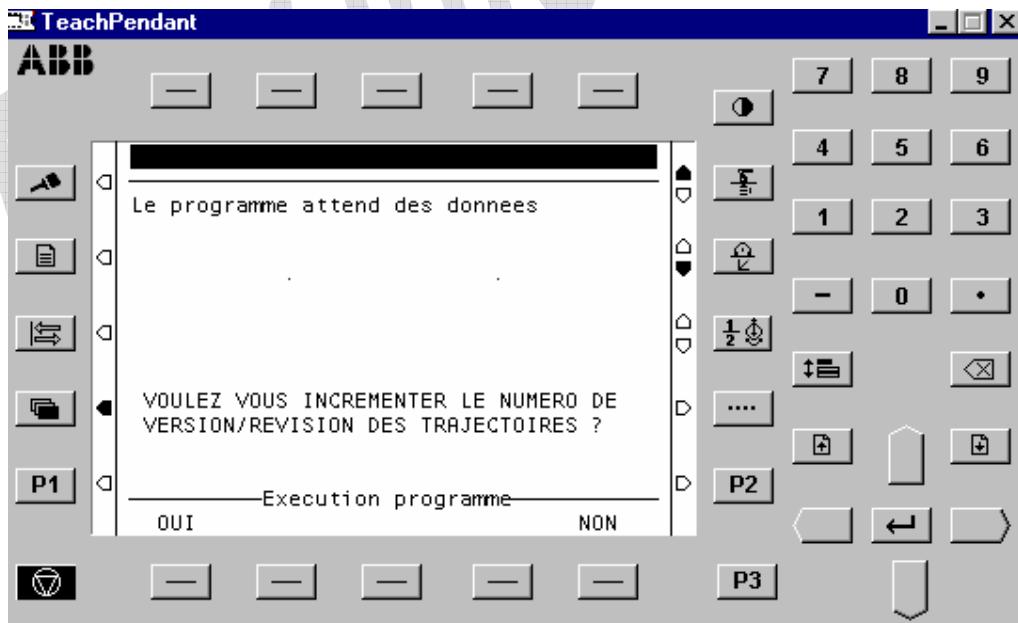
- PSA.prg
- SUPROB.prg
- CONFIGUR.sys
- TRAJECT.sys
- ACTMSITE.sys
- ACTUSITE.sys

Dans le cas d'une pince électrique, on trouve également les fichiers SWUSR.C.sys, SWUSR.F.sys et SWITDATA.sys, pour la commande de la carte swit.

**NOUVEAU 3.68:** Les paramètres de pinces sont également sauvegardés par cette opération, mais uniquement sur disquette.

Les autres modules du standard ne sont jamais sauvegardés, puisqu'ils sont identiques sur tous les robots.

Si une modification a eu lieu dans les trajectoires depuis la dernière sauvegarde, le standard le détecte et vous propose, avant de sauvegarder:



Si vous répondez « NON », le fichier TRAJECT conserve le même numéro de version / révision. Si vous répondez « OUI », celui-ci est incrémenté de 1. Par exemple, s'il était 3.2, il passe à 3.3. La date et l'heure de la modification sont également sauvegardées.

Si vous n'êtes pas sûr de vouloir conserver les modifications, vous pouvez passer outre. Ce mécanisme est ensuite utilisé pour vérifier que le fichier Trajectoire chargé est bien celui du dernier indice, et non pas une version antérieure.

En cas de recharge d'une version plus ancienne, un message d'alerte sera affiché, indiquant que la version chargée n'est pas la plus récente.

Ce mécanisme fonctionne même si vous rechargez entièrement le programme. (Magique !)

Si des modifications de trajectoires ont eu lieu et n'ont pas été sauvegardées, le programme affiche, à chaque lancement En Production :



Cet écran reste affiché pendant 3 secondes.

La sauvegarde de paramètres système ne peut pas actuellement se faire automatiquement. Vous devez donc les sauvegarder manuellement sur la disquette du robot (ou sur le PC distant).

Validez le menu "PARAMETRES SYSTEME", "SAUVER TOUT SOUS...". Donnez un nom de répertoire ou acceptez le choix par défaut. Tous les paramètres sont sauvegardés sur la disquette.

Si la liaison Ethernet est installée, vous pouvez sauvegarder les paramètres directement sur le PC distant.

Votre disquette de sauvegarde est maintenant prête. À chaque modification importante de programme, pensez à sauvegarder en Ramdisk, et faites une mise à jour de la disquette de temps en temps. Cela ne prends que quelques secondes.

Pourquoi sauvegarder en Ramdisk le fichier "PSA.prg" alors qu'il est strictement identique à tous les robots ?

Simplement pour une question de commodité. En effet, si vous faites "FERMER PROGRAMME", ce fichier est effacé de la mémoire. S'il n'était pas en Ramdisk, il faudrait aller chercher la disquette du standard pour le recharger. Il est plus rapide de le recharger à partir du Ramdisk.

#### **4 - Procédure de Rechargement à froid (Cold Start):**

Le rechargement du logiciel ne devrait normalement jamais se faire, sauf après une panne du système ayant entraîné la perte totale de la mémoire, et un redémarrage "à froid".

##### *a - Rechargement à partir de la disquette Robot:*

Après le rechargement du système de base, insérez la disquette des données du robot dans le lecteur. Celle ci doit contenir les fichiers suivants:

- PSA.prg
- TRAJECT.sys
- ACTUSITE.sys
- CONFIGUR.sys
- ACTMSITE.sys
- SUPROB.prg

Dans le cas d'une pince électrique, on trouve également les fichiers SWUSR.C.sys, SWUSR.F.sys et SWITDATA.sys.

Par la fenêtre Programme, faites « FICHIER », « OUVrir ». Chargez le programme principal : **PSA.prg**, et **CONFIGUR.sys**. Appuyez sur EXEC, éventuellement en simulation. Celui ci détecte qu'il s'agit d'une ré-installation, car les fichiers ne se trouvent pas sur le Ramdisk. Il les copie donc à partir de la disquette, puis vous invite à charger les paramètres système, à insérer la disquette standard (sauf si la liaison Ethernet est valide), et à faire un P-START. Tout se recharge et le programme se lance.

##### *b - Rechargement à partir d'un PC distant:*

Par le Gestionnaire de fichiers, recopiez le répertoire "*c:\\$4boot\systems\Nom\_robot*" sur le Ramdisk. Chargez les paramètres système et faites un P-START (menu Maintenance).

Le standard se recharge à partir du PC distant, avec la version/revision correspondant à la dernière ayant été exécutée sur ce robot.

#### **5 - Mises à jour du standard et redémarrage (P-START):**

Les mises à jour du programme standard s'effectuent très simplement:

Insérez la disquette "STANDARD PSA " contenant la nouvelle version du logiciel dans le lecteur et chargez le programme "**Installe**".

Si le robot avait déjà été équipé d'une version 3.6 ou supérieure, le menu

F1 : Mise à jour du standard 3.6x → 3.6y

est proposé.

Cela signifie que la mise à jour automatique est possible. Validez ce choix.

Si des évolutions de paramètres système ont eu lieu, le programme génère des fichiers B1, B2 ou B3 à **ajouter** (et **non pas à charger**, ce qui écrasera les anciens).

Faites alors un P-Start.

Si le robot était dans une version inférieure à 3.6, la mise à jour automatique n'est pas possible, il faut refaire une première installation complète et mettre à jour manuellement le fichier EIO.sys avec les entrées sorties propres au site, ce qui est évidemment beaucoup plus laborieux..

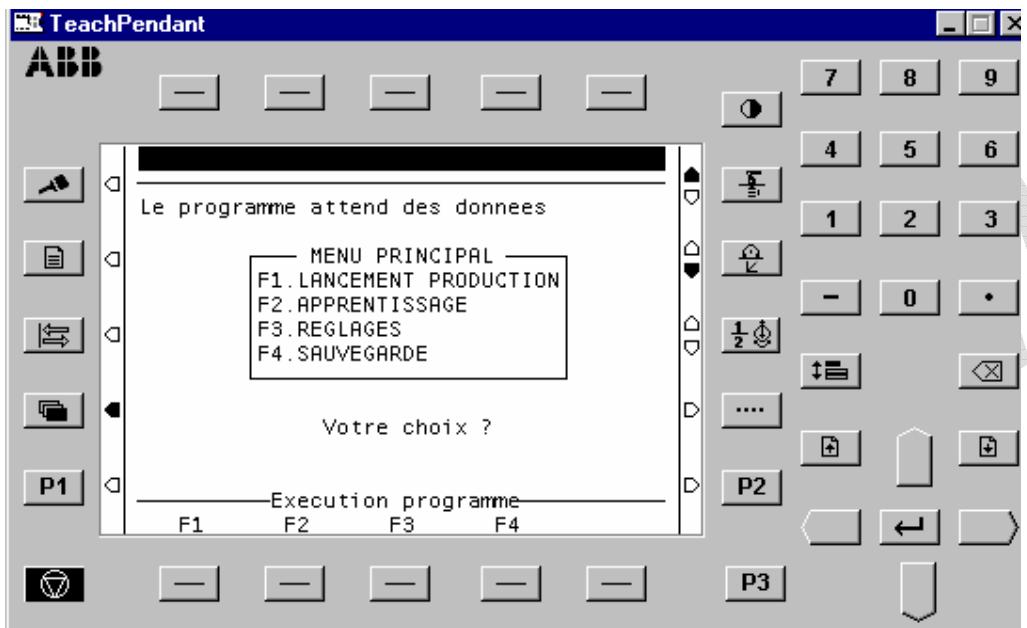
La disquette standard contient un fichier LISEZMOI.DOC, qui récapitule les principales évolutions du standard d'une version à l'autre. Ce fichier est éditable sur PC au moyen d'un éditeur de texte quelconque. Il est fourni en annexe de cette documentation.

**ATTENTION:**

Tout P-Start doit être suivi d'un redémarrage normal pour que les listes d'instructions (Commune 1, 2 et 3) soient remplies automatiquement avec les routines disponibles.

### III - PROGRAMMATION

Lancez le programme en exécution. Le menu principal apparaît:



Vous pouvez, sans n'avoir rien programmé, tenter de valider le menu "Lancement Production". Le robot tentera de rejoindre la position de Repli, qui par défaut est définie avec des valeurs impossibles à atteindre. Vous obtiendrez donc simplement une erreur d'exécution, sans conséquence. Mais vous serez "planté", avec redémarrage du programme au début obligatoire.

Pour gagner du temps, il est donc préférable de respecter l'ordre opératoire qui suit. Hormis la détermination de repères corrects, il est possible de réaliser un cycle de test complet, y compris avec reprises, en moins de 30 minutes.

#### 1 – Création des points remarquables

Les points dits "remarquables" sont des positions définies pour tous les robots, et **indépendantes de l'outil**. Cela signifie que les axes robot rejoindront toujours cette position absolue, quelquesoitoit l'outil (Tooldata) en cours. Il s'agit:

- Du REPLI.
- Du PEO.
- De la position de PURGE (Si Encollage).
- De la position de SERVICE.
- De la position de CHANGEMENT ELECTRODES (Si SR).

Ces positions, toutes prédéfinies, sont modifiables via le menu "APPRENTISSAGE","POINTS SPECIAUX".



Validez un choix. Pour chaque position choisie, l'algorithme est le même: un menu vous propose de rejoindre ou non la position demandée. Si vous répondez "OUI", le robot tente d'y aller, **par le chemin le plus court**. Si vous répondez "NON", le robot reste sur place. Le pupitre affiche alors:

"VOUS POUVEZ DEPLACER LE ROBOT, PUIS FAITES EXEC"

Notez que la première fois, le robot ne peut pas rejoindre la position demandée, puisque les points par défaut sont créés volontairement avec des valeurs inatteignables. Tant que le point n'a pas été modifié, le standard ne propose pas de rejoindre le point.

Amenez le robot sur la position voulue à l'aide du joystick, puis faites "EXEC".

Le programme vous demande alors de valider ou non la nouvelle position. Si vous répondez "OUI", celle ci est immédiatement enregistrée. Sinon, elle est ignorée.

Faites ainsi pour toutes les positions remarquables disponibles. Notez bien que celles ci dépendent des applications déclarées: vous ne pouvez pas modifier une position de changement électrodes si il n'y a pas d'application soudage! D'ailleurs le menu proposé ne vous le permet pas. (Les menus sont dynamiques en fonction des applications installées et des circonstances).

Vous remarquerez qu'aucun "MODPOS" n'est nécessaire. (ni possible, d'ailleurs).

NB: Pour des raisons de sécurité évidentes, le programme refuse de déplacer le robot vers la position voulue si celui-ci est en mode automatique. Un passage en mode Manuel est obligatoire.

Une fois toutes vos positions remarquables créées, vous pouvez passer au menu "Trajectoires de Service".

## **2 – Création des trajectoires de Service**

Les trajectoires dites "de Service" regroupent:

- La trajectoire de Lancement (Du Repli au PEO).
- La trajectoire de Repli (Du PEO au Repli).
- La trajectoire de Service (Du PEO au point de Service et retour).

Puis, selon les applications:

- La trajectoire de Purge (Du PEO au point de purge et retour).
- La trajectoire de Changement Electrodes.
- La (les) trajectoires de rodage(s).
- La trajectoire de contrôle outil (cible).

Ces trajectoires sont appelées soit automatiquement (Lancement, Repli, Purge, Rodage...) en cours de production, soit sur demande de l'opérateur en agissant sur le BP "DEMANDE DE SERVICE". Elles sont donc appelées Trajectoires de Service au sens large, car elles ne participent pas directement à la production. (Elles ne soudent pas, elles ne collent pas, elles ne clinchent pas, etc...).

La trajectoire de contrôle (cible) permet de vérifier que l'outil et le robot ne sont pas décalés. Elle doit comprendre un point de soudure sur la cible spécial. Cette trajectoire est exécutée automatiquement par le standard **a la fin** du cycle en cours, selon la fréquence définie dans le paramètre **NbCycle\_Av\_Cont**. La valeur 0 permet d'inhiber la fonction.

Cette trajectoire peut être utilisée dans d'autres contextes: par exemple en Rivetage, si l'on souhaite effectuer des éprouvettes. C'est une trajectoire de contrôle au sens large, qui existe pour tous les process.

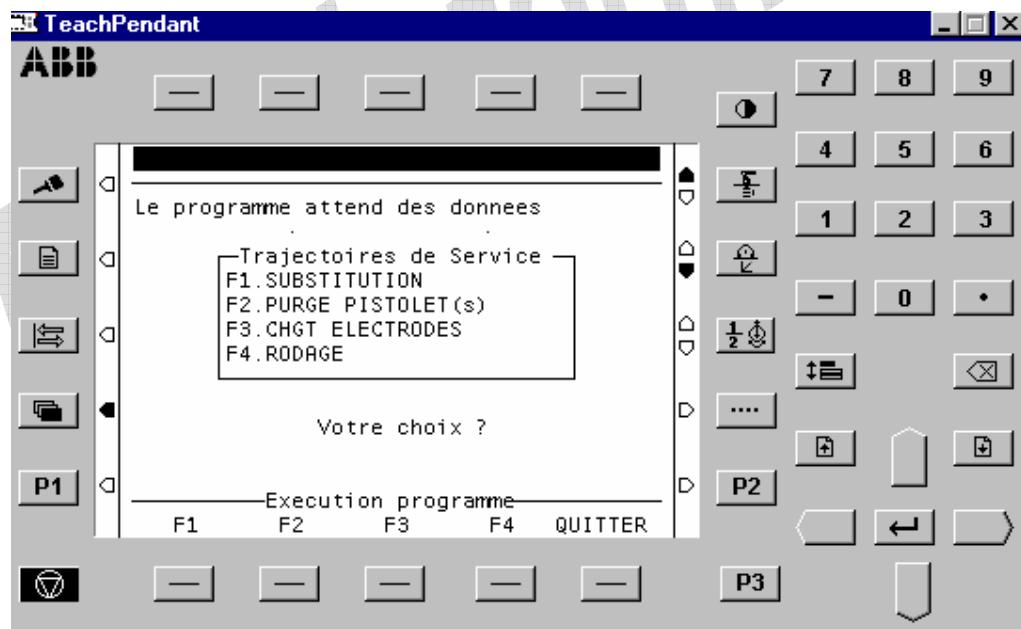
Elle est notamment appelée systématiquement (depuis la V 3.7\_2) à la fin de la trajectoire de Changement Electrodes.

Si un décalage existe, la soudure est effectuée en dehors du trou de la cible, donc sur une partie isolante. Ceci déclenche donc un défaut soudure, permettant de vérifier le décalage.

Toujours dans la rubrique « Apprentissage », validez le menu « Trajectoires de service ».

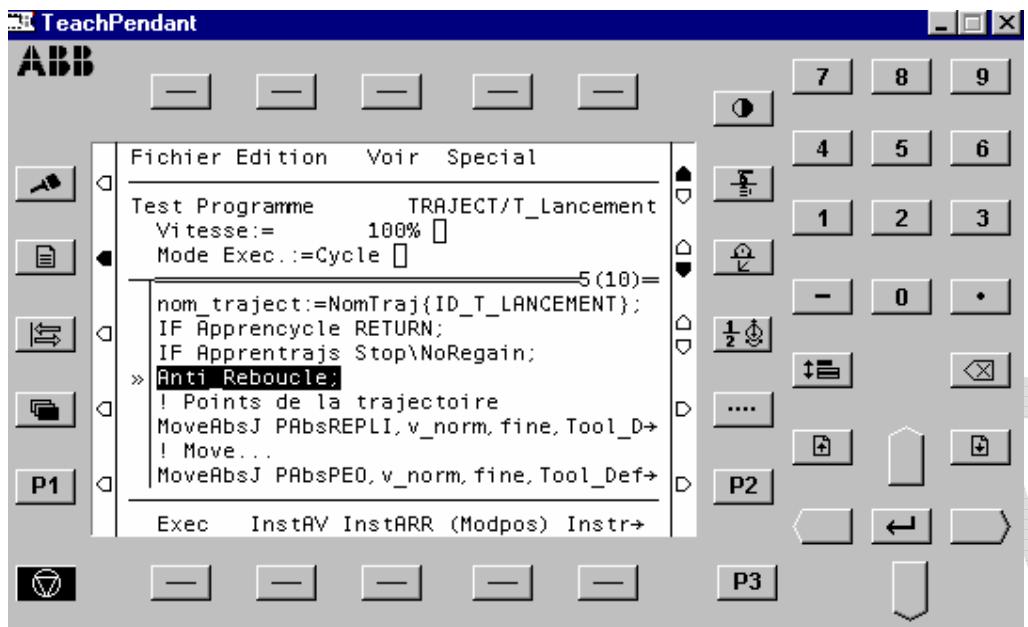


Si vous validez «AUTRES», c'est le menu des trajectoires de service spécifiques aux applications installées qui apparaît:



Cet exemple corresponds à un cas soudure + encollage.

Validez le choix de la trajectoire souhaitée en répondant au menu. Le programme s'arrête alors automatiquement sur la première ligne "utile" de la trajectoire voulue. Vous êtes déjà "dans" la trajectoire. C'est le standard qui vous y a conduit. Il vous reste à créer les points, si nécessaire. Par défaut, toutes les trajectoires contiennent déjà les positions remarquables nécessaires.



Par exemple, la trajectoire de lancement ci dessus contient déjà les mouvements:

- Aller au Repli.
- Aller au PEO.

Il vous suffit d'insérer les éventuels points intermédiaires pour éviter un obstacle entre les deux.

Notez que l'outil courant (tooldata) a été automatiquement positionné par le standard: il s'agit de l'outil *Tool\_Defaut*.

La trajectoire de purge contient, par exemple:

- Aller au PEO.
- Aller au point de Purge.
- Purger le pistolet Debit s'il existe.
- Purger le(s) pistolet(s) Mastic s'il(s) existe(nt).
- Retourner au PEO.

Lorsque vous avez fini de créer vos points, faites "EXEC". Le système vous propose de redémarrer à l'endroit du curseur ou à l'endroit où le programme s'était arrêté.

### **ATTENTION:**

Si vous répondez "Programme", la trajectoire sera exécutée depuis le début, donc risque de casse possible.

Si vous répondez "Curseur", l'exécution sera stoppée par le Standard, pour permettre un éventuel retour en arrière. Un nouvel appui sur "EXEC" permet de revenir au menu de choix des trajectoires.

Vous pouvez alors choisir une autre trajectoire de service à créer, ou passer à l'apprentissage des trajectoires de travail.

### **3 – Crédation des trajectoires de Travail**

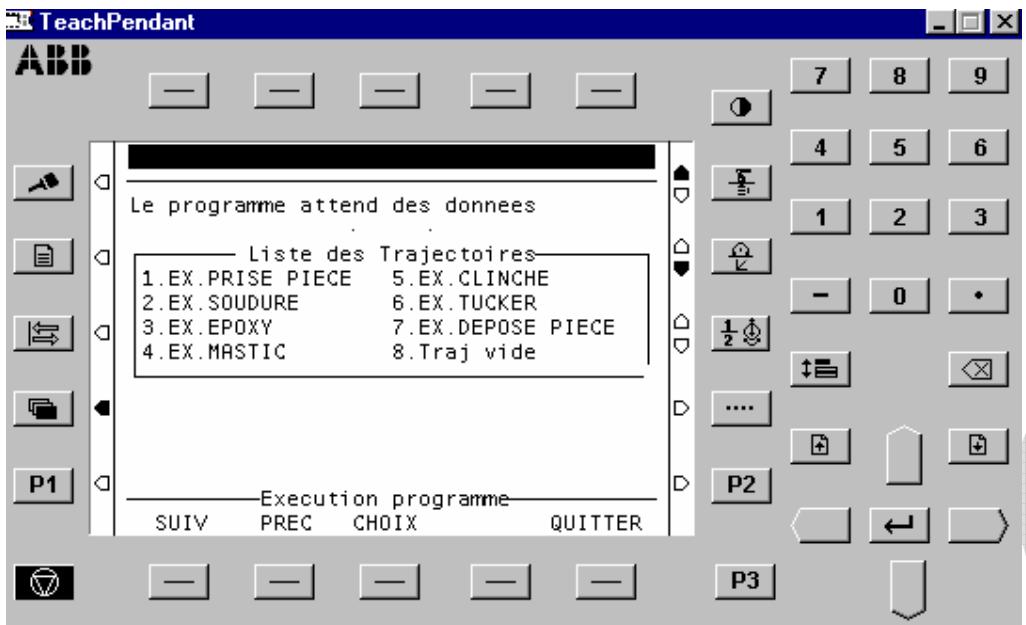
En règle générale, une trajectoire est un ensemble de positions robot correspondant à une zone gérée par l'automate. Par exemple: prise d'une pièce dans un montage et dégagement. Bien que cette règle ne soit pas absolue, son application permet une programmation simple et rapide.

Admettons qu'un cycle robot soit constitué des opérations suivantes:

- 1 - Approche d'un montage.
- 2 - Prise d'une pièce.
- 3 - Encollage sous un pistolet fixe.
- 4 - Dépose de la pièce sur un convoyeur d'évacuation.
- 5 - Retour au DPO.

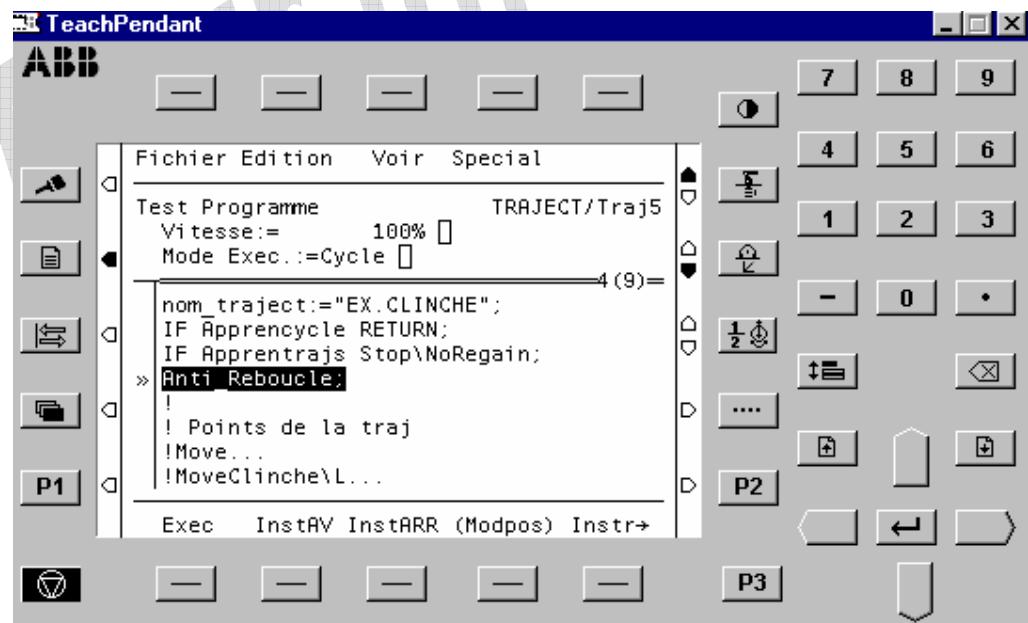
On considère que seuls les montages de prise et dépose sont surveillés par l'automate, la zone d'encollage étant totalement dégagée. Pour gagner du temps de cycle, on considère que l'approche du montage est possible dès que le code automate est reçu, même si le montage n'est pas encore prêt. L'approche sera donc une trajectoire complète, mais sans zone affectée.

Validez le menu "APPRENTISSAGE", "TRAJECTOIRES DE TRAVAIL". Choisissez un numéro quelconque dans la liste proposée. Vous pouvez faire défiler la liste page par page avec les choix "SUIV" et "PREC". Au départ, des exemples de trajectoires de chaque type sont fournis. Prenez par exemple la trajectoire n° 5.



Le programme s'arrête alors dans la routine "Traj5", sur la première ligne où vous pouvez créer des points.

Une trajectoire "vide" se présente ainsi:



**ATTENTION**  
les 4 premières lignes de la trajectoire contiennent des instructions indispensables qu'il est  
**interdit de supprimer.**

Si vous le faites, le Standard le détectera et interdira totalement de s'exécuter, en affichant

#### « ERREUR DE PROGRAMMATION »

Vous serez alors contraint de ré-écrire manuellement les lignes supprimées.

ATTENTION: Dans ce cas, le robot peut avoir le temps de bouger plus ou moins avant que le défaut ne se déclenche !

La première ligne vous permet de donner un nom à votre trajectoire. C'est ce nom qui s'affichera ensuite dans la liste des trajectoires, de même qu'à l'écran en cours de production. Vous pouvez donner un nom absolument quelconque, incluant des espaces ou des caractères spéciaux. Par exemple:

"Approche du montage N 4".

Bien que le nom puisse en théorie faire 80 caractères, il est préférable de se limiter à 23, pour que celui-ci puisse apparaître en entier à l'écran de production.

Les points de la trajectoire doivent être créés après la ligne "Anti\_Reboucle". Ne vous préoccupez pas des gestions de zone pour l'instant. Dans notre exemple d'une approche, le premier point de la trajectoire devra pouvoir être rejoint directement à partir du PEO, et le dernier point devra être le plus prêt possible du montage, mais **Hors interférence** avec celui-ci.

Si vous souhaitez que le robot **ne s'arrête pas** si l'autorisation de prise de pièce est déjà présente, il vous suffit de créer le dernier point en point de passage.

Une fois vos points créés, faites "EXEC". Le programme s'arrête sur la dernière ligne de la trajectoire, ce qui vous permet, si vous le souhaitez, de la retester depuis le début.

Sinon, le programme revient au menu donnant la liste des trajectoires. Notez que le programme vous affiche maintenant le nom en clair que vous avez donné à votre trajectoire.(Magique!).

Vous pouvez maintenant créer la trajectoire de prise de pièce. Choisissez par exemple, la trajectoire N° 6. Le premier point de cette trajectoire sera cette fois à l'intérieur de la zone gérée par l'automate. En effet, le programme standard teste les autorisations d'accès **AVANT** d'entrer dans une trajectoire, et libère la zone juste **APRES** celle-ci. Donc, le premier point d'une trajectoire doit être dans la zone, et le dernier doit être en dehors.

ATTENTION: Si le dernier point de la trajectoire est en point de passage, il y a une anticipation de la sortie hors zone par rapport au point appris. 3 méthodes peuvent être utilisées pour pallier à ce problème:

- 1 - Créer le dernier point de la trajectoire **très** en dehors de la zone. C'est le plus simple lorsque la zone suivante est relativement éloignée.
- 2 - Utiliser l'argument \NoConc sur le dernier point de la trajectoire. Cela résoudra le problème, mais le robot risque de marquer un léger arrêt sur le point, le temps de tester le hors zone suivant s'il y en a un.

3 - La solution idéale est d'utiliser la routine du standard:

**Fin\_De\_Traj** *pt\_vise,vitesse,zone,outil,objet.*

Cette routine, à appeler en sortie de zone, amène le robot au point visé et active la sortie hors zone exactement au centre de la sphère définie pour le point de passage. Il n'y a ni anticipation, ni retard.

Par défaut, le mouvement s'effectue en mode articulaire. Le paramètre optionnel \L permet d'effectuer un mouvement linéaire.

NOTA : l'appel à **Fin\_de\_traj** ne se fait pas obligatoirement à la fin de la trajectoire. Si, par exemple, des points d'approche de la trajectoire suivante sont nécessaires, mais sans zone surveillée, il est tout à fait possible de mettre ces points après l'instruction **Fin\_de\_Traj**.

De plus, l'instruction **Fin\_de\_Traj** provoque la vérification de l'état du process que l'on vient d'effectuer, en affichant un message d'erreur et un choix "Continuer", "Refaire", et ne libère la zone que si l'on a validé "Continuer".

De même que pour la trajectoire d'approche, donnez un nom en clair à votre trajectoire, par exemple:

"Prise pièce sur montage N 4"

Créez ainsi toutes les trajectoires qui constitueront votre cycle. La trajectoire de dépose possède une particularité: ce sera la dernière du cycle. Son dernier point sera donc très probablement le PEO, et nous aurons l'instruction

MoveAbsJ PabsPEO,vmax,...

Si le retour du robot vers cette position à partir du convoyeur s'effectue hors zone d'interférence, il serait dommage d'attendre que le robot ait rejoint le PEO pour solliciter un nouveau code cycle de l'automate, ce qui est le fonctionnement par défaut du standard. Ce dialogue pourrait en effet être anticipé et s'effectuer en temps masqué, pendant le retour du robot au PEO. Mais comment faire ? C'est bien entendu très simple...

Il vous suffit d'insérer un appel à la routine "**Fin\_De\_Cycle**" dans votre trajectoire, en créant le point à l'endroit exact où le robot se trouve hors zone du montage, en fin de dépose. Vous pouvez déclarer ce point en passage. La routine "**Fin\_De\_Cycle**" libère la zone (de la trajectoire) et monte la demande de code **exactement au centre** de la sphère que vous avez définie pour le point de passage. Il n'y a ni anticipation, ni retard.

Ainsi, si un nouveau code peut être envoyé par l'automate, son acquisition et son contrôle seront faits en temps masqué.

Si, de plus, le dernier point de votre trajectoire (PabsPEO) est déclaré en point de passage, le robot pourra repartir directement sur la trajectoire d'approche du montage de prise, **sans s'arrêter au PEO**.

On voit que si la programmation est correcte, un robot **ne s'arrête jamais si toutes les autorisations sont présentes.**

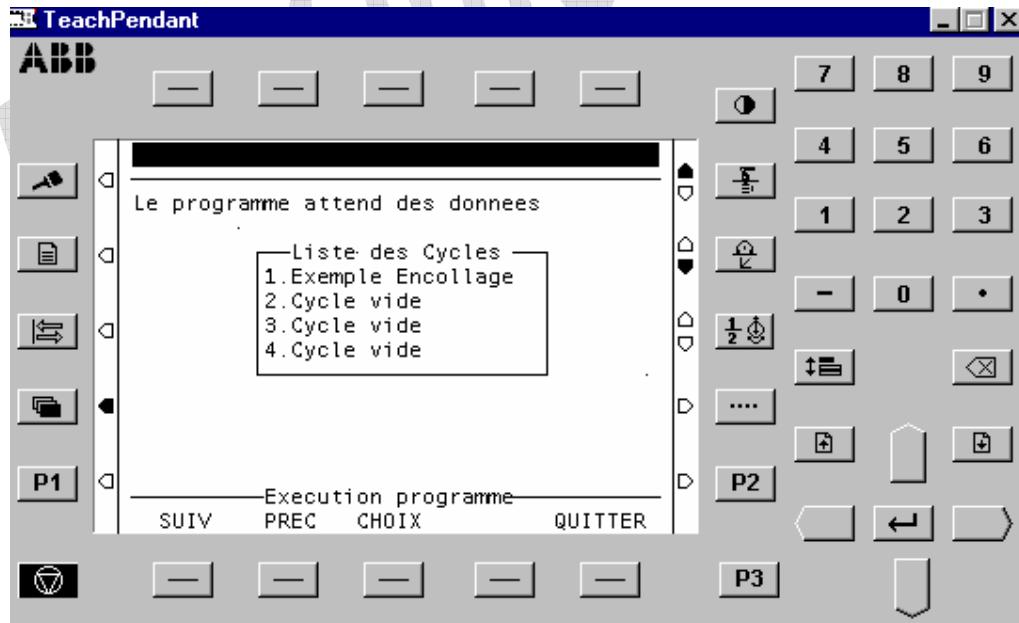
Une fois toutes vos trajectoires créées, mémorisez ou notez sur un papier les numéros des trajectoires que vous avez utilisées, dans l'ordre où elles devront être exécutées. C'est la petite contrainte du standard. Dans notre cas, nous aurons, par exemple:

- Traj 5: Approche.
- Traj 6: Prise pièce.
- Traj 12: Encollage.
- Traj 15: Dépose pièce et retour.

Vous pouvez maintenant créer votre cycle complet.

#### **4 – Crédation et test des Cycles, gestion des Hors Zone**

Toujours dans le menu "APPRENTISSAGE", choisissez "MODIF CYCLES". De même que pour les trajectoires, le programme vous affiche la liste des cycles existants, avec leur nom en clair. Choisissez le numéro de cycle correspondant au code automate pour ce travail, par exemple le 10. Notez que comme 31 codes peuvent être reçus, les 31 cycles correspondants sont pré-créés, mais vides.



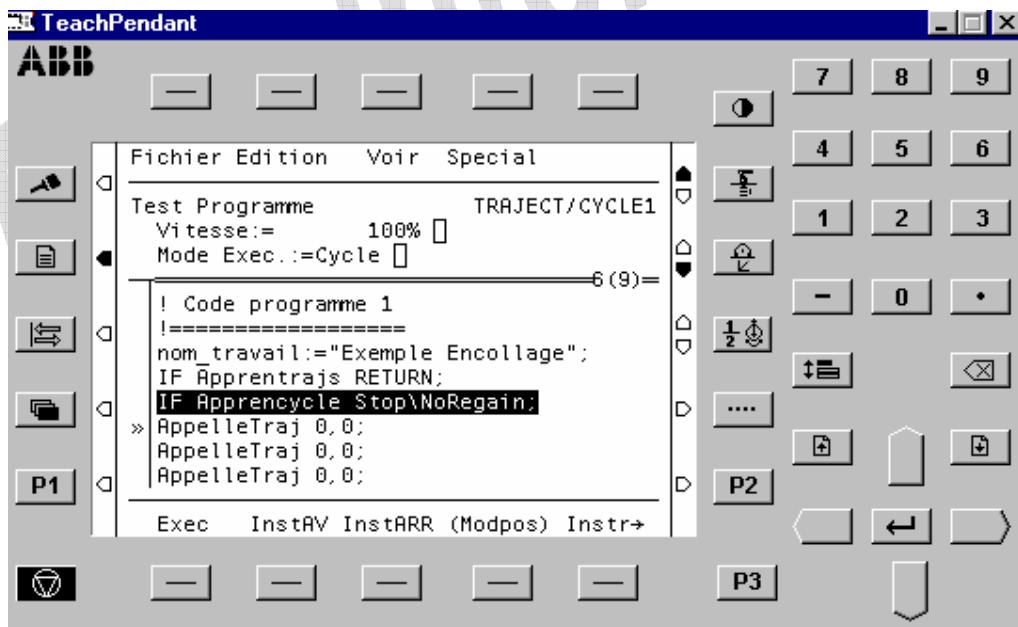
Choisissez un numéro, en accord avec l'automaticien.

Un ultime menu vous propose le choix suivant:



Le mode TEST vous permettra de tester votre cycle complet, avec ou sans pièce. Mais il faut d'abord le créer. Validez donc CREATION / MODIF.

Le programme s'arrête automatiquement sur la première ligne "utile" du cycle. Vous devez commencer à avoir l'habitude du système...



Comme pour les trajectoires, vous pouvez donner un nom en clair à votre cycle, en documentant la première ligne.

**Les deux lignes suivantes ne doivent pas être supprimées.**

Ensuite, en accord avec l'automaticien, définissez les numéros de zone affectés aux montages de prise et dépose, par exemple 1 et 2. Il vous reste à écrire l'appel des trajectoires et des zones en utilisant la procédure "**AppelleTraj**".

Votre cycle complet ressemblera alors à ceci:

```
PROC CYCLE10()
    nom_travail:="PAVILLON Z8";
    IF Apprentrajs RETURN;
    IF Apprenecycle Stop\NoRegain;
    AppelleTraj 5,0;
    AppelleTraj 6,1;
    AppelleTraj 12,0;
    AppelleTraj 15,2;
ENDPROC
```

Comme vous le voyez, la routine **AppelleTraj** a besoin de deux données: le numéro de trajectoire à appeler, et le numéro de zone à gérer. Si aucune zone n'est à tester, mettez simplement 0.

#### ATTENTION:

- L'utilisation de la routine AppelleTraj est INDISPENSABLE. Il est interdit d'utiliser un appel direct à la routine trajectoire elle-même.
- Un cycle ne doit pas contenir autre chose que des "AppelleTraj". En effet, si vous mettez du calcul ou des tests dans un cycle, la gestion automatique des reprises pourra ne pas fonctionner.
- Il est rigoureusement interdit de placer des instructions de mouvement dans un cycle, car elles seront exécutées lors d'une reprise.

Une fois que vous avez écrit tous les "AppelleTraj" nécessaires pour réaliser votre cycle, vous pouvez faire "EXEC". Le programme revient alors au menu "CREATION/MODIF","TEST", en ayant sauté l'exécution de toutes les trajectoires de votre cycle, pour éviter de vous surprendre avec des mouvements robot non souhaités.

Vous pouvez maintenant, si vous le souhaitez, valider le choix "TEST" du cycle. Celui-ci est alors immédiatement exécuté. Ceci est possible aussi bien en Manuel qu'en Auto. Le code Automate n'est évidemment pas testé dans ce mode. Par contre, toutes les autorisations d'accès zone sont testées, de même que les interverrouillages robot, exactement comme en production.

### Peut on gérer les autorisations d'accès à l'intérieur d'une trajectoire ?

Oui bien sûr. Le Standard propose un fonctionnement "par défaut": si l'on n'écrit rien, le standard attend l'autorisation d'accès zone spécifié, puis lance la trajectoire tout en surveillant en permanence l'autorisation de zone, puis libère la zone. Normalement, une trajectoire ne contient donc que des instructions de mouvement et de process.

Il est toutefois possible d'ajouter des zones intermédiaires, successives ou imbriquées. On fera pour cela appel aux procédures "**Att\_Acces\_Zone X**" à l'entrée d'une zone, et "**Fin\_de\_Zone**" à sa sortie. La gestion de la zone spécifiée sur l'appel de la trajectoire sera tout de même assurée.

La routine **Fin\_de\_Zone** permet de monter le hors zone voulu de façon précise sur point de passage. Mais comme elle contient un mouvement, elle ne peut être utilisée dans un cycle. Si l'on est contraint de gérer le hors zone dans un cycle, il faut utiliser l'instruction **Hors\_zone**, mais qui n'est pas précise.

Naturellement, une attente d'autorisation de zone ou sa libération doivent être programmées **en dehors** de la zone en question.

Rappelons que le fait de mettre 0 dans le 2ième paramètre de la routine AppelleTraj permet de ne déclarer aucune zone pour cette trajectoire.

### Peut on avoir deux zones "imbriquées" ?

La version 3.3 du standard apporte sur ce point une évolution majeure:

1 - Toutes les zones peuvent être imbriquées, ce qui n'était pas le cas avant. On a donc 10 niveaux au lieu de 2.

2 - Toutes les zones peuvent être "prises" et "libérées" à n'importe quel moment et dans n'importe quel ordre. Elles n'ont plus besoin d'être libérées dans l'ordre inverse de leur prise.

3 - Toutes les zones "prises" sont surveillées et provoquent un arrêt robot en cas de perte.

4 - Les zones peuvent être "prises" et "libérées" aussi bien dans un cycle que dans une trajectoire.

5 - Elles peuvent être libérées de façon précise sur point de passage (*Fin\_de\_Zone*, *Fin\_de\_Traj*, *Fin\_de\_Cycle*), ou de façon approximative (*Hors\_Zone*).

6 - La même zone peut être prise plusieurs fois de suite sans avoir été libérée, et libérée plusieurs fois de suite sans avoir été prise. (Att\_zone 1 + Att\_zone 1 + Att\_zone 1 autorisé).

Par exemple, les écritures suivantes sont désormais autorisées:

<i>Instruction</i>	<i>Action</i>	<i>Zones surveillées</i>
PROC CYCLE1()		
....		0
Att_acces_zone 10;	Prise de la zone 10	10
AppelleTraj 1,1;	Appel trajectoire 1 avec zone 1. Libération automatique de la zone 1 en fin de traj 1	10 et 1
AppelleTraj 2,2;	Appel trajectoire 2 avec zone 2. Libération automatique de la zone 2 en fin de traj 2.	10 et 2
Hors_Zone 10;	Libération zone 10.	0
Att_acces_zone 9;	Prise de la zone 9	9
AppelleTraj 3,3;	Appel trajectoire 3 avec zone 3 Libération automatique de la zone 3 en fin de traj 3	9 et 3
Hors_zone 9;	Libération zone 9.	0
ENDPROC		

Et par exemple, dans la trajectoire 1 (en tenant compte de l'écriture du cycle ci-dessus):

<i>Instruction</i>	<i>Action</i>	<i>Zones surveillées</i>
PROC TRAJ1()		
....		10 et 1
Att_acces_zone 4;	Prise de la zone 4	10 et 1 et 4
MoveJ...;	Mouvement	10 et 1 et 4
Att_Acces_zone 5;	Prise de la zone 5.	10 et 1 et 4 et 5
Depalettise...;	Process..	Idem
Hors_zone 4;	Libération zone 4.	10 et 1 et 5
Att_acces_zone 8;	Prise de la zone 8.	10 et 1 et 5 et 8
MoveJ...;	Mouvement	Idem
Hors_zone 5; Fin_de_Zone 8,*,... Fin_de_traj *,...	Libération zone 5 (non précise) Libération zone 8 (précise) Vérification des défauts process et Libération de la zone 1 (précise).	10 et 1 et 8 10 et 1 10 (car prise dans le cycle)
MoveJ... MoveJ...	Mouvement	
ENDPROC		

#### NOUVEAU DEPUIS LA VERSION 3.4:

En cas d'utilisation de la routine Fin\_de\_Traj avant la fin physique de la trajectoire (car il reste des points de mouvement pur à faire), si l'on abandonne le programme après que la zone a été effectivement libérée, alors la reprise se fait automatiquement sur la trajectoire suivante.

Autrement dit, le standard considère que comme le Hors Zone de la trajectoire est monté, le process a été terminé, et qu'il n'y a donc pas lieu de reprendre la trajectoire abandonnée, mais la suivante.

Une extension de ce principe a été appliquée à la routine Fin\_de\_zone qui accepte maintenant un paramètre optionnel \Fin\_Process. Celui ci indique de tester immédiatement les défauts process mémorisés lors de la trajectoire, et de considérer que la trajectoire est terminée (en cas de reprise), même si un arrêt se produit avant la fin physique de celle ci.

L'utilisation de Fin\_de\_Traj est donc identique à Fin\_de\_zone\Fin\_Process.

### **Bifurcations de trajectoires:**

Dans certains cas, il peut être nécessaire de réaliser une bifurcation de trajectoire. Par exemple, déposer la pièce sur le montage 4 s'il est libre, ou sur le montage 6. (Exemple : poste de contrôle).

Le choix de l'endroit où déposer la pièce est du ressort de l'automate, qui devra envoyer un numéro d'autorisation d'accès zone différent pour chaque montage.

Côté robot, cette bifurcation est gérée automatiquement par la routine AppelleTraj (encore elle !), grâce à ses deux paramètres optionnels:

- "Ou\_traj"
- "Si\_zone"

Dans notre cas, il suffira d'écrire, dans le cycle:

```
AppelleTraj 15,2Ou_traj := 16Si_zone := 4;
```

Vous avez deviné: le programme effectuera la trajectoire de dépose 15 si la zone 2 est autorisée, et la trajectoire 16 si c'est la zone 4.

ATTENTION: la cohérence des autorisations d'accès est vérifiée: dans notre cas, il est interdit d'avoir simultanément les zones 2 et 4. Si cela se produit, le programme affiche un message d'erreur bloquant:

"ERREUR AUTOMATE: Zones 2 et 4 SIMULTANEES !!!"

Actuellement, le Standard ne gère que les bifurcations "simples", c'est à dire à 2 branches de 1 trajectoire chacune. C'est généralement suffisant.

Il peut toutefois arriver des cas complexes où cette gestion ne suffit plus. Exemple: enchaîner 2 trajs dans un cas de la bifurcation, et seulement 1 dans l'autre.

La seule solution possible, autre que de concaténer les deux trajs en une seule, est d'utiliser la variable du standard traj\_encours, qui contient toujours le numéro de la dernière traj exécutée, et d'écrire le cycle comme suit:

```
PROC CYCLE1()
```

```
Appelletraj 1,1\ou_traj:=2\Si_zone:= 2;  
If traj_encours= 1 OR traj_encours= 3 Appelletraj 3,3;  
If traj_encours= 3 OR traj_encours= 4 Appelletraj 4,4;
```

```
!deuxième branche de la bifurcation  
If traj_encours= 2 OR traj_encours= 5 Appelletraj 5,5;  
ENDPROC
```

Le OR est nécessaire pour la reprise: si on abandonne la traj 3, la variable traj\_encours vaut 3, et il faut donc le traiter si l'on veut ré-exécuter la traj 3.

Voilà, c'est déjà fini ?

Presque: il reste à créer les trajectoires de reprise, si toutefois vous souhaitez utiliser cette possibilité.

A ce niveau, il est utile de penser à sauvegarder votre travail en validant "SAUVEGARDES", « SAUVEGARDER LES MODIFS ». Vous pouvez sauvegarder en ramdisk, sur disquette ou directement sur le PC distant si NFS est installé.

Si vous faites "FERMER PROGRAMME", volontairement ou par erreur, pas de panique: vos trajectoires sont toujours en mémoire, car déclarées dans un module "système". Vous les récupérez en faisant "FICHIER","OUVRIR","PSA" (à partir du Ramdisk), ou n'importe quel autre programme principal.

## **5 - Interverouillages entre robots:**

Lorsque plusieurs robots peuvent être amenés à travailler dans la même zone, il est nécessaire d'assurer la sécurité anti-collision par des infos d'interverouillage.

Le programme gère jusqu'à 16 interverouillages, mais ils ne sont pas forcément tous valides, selon la configuration hardware:

- En dialogue PROFIBUS, on peut avoir soit 3 interverouillages déclarés sur la carte ACTION, soit 16 déclarés sur la carte API. Dans ce cas, c'est l'automate qui sert de "boîte au lettres".
- En dialogue FIPIO, on est limité à 7 interverouillages sur la carte API.
- En dialogue par E/S logiques, on est limité à 3.

Avant d'entrer dans la zone dangereuse, chaque robot doit appeler la routine *Entree\_Interverr*, en précisant le numéro d'interverouillage. Les points suivants et précédents peuvent être en points de passage, ainsi le robot ne s'arrêtera pas, sauf bien sûr si l'info est absente. Dans ce cas, le message "Point de passage dégénéré en point d'arrêt" apparaît dans les archives, sans autre conséquence.

Au cas où deux robots peuvent se présenter dans la zone exactement au même moment, il est prudent d'utiliser le paramètre optionnel \Maitre, qui permet de définir un robot prioritaire. Celui considéré comme *non maître* fera alors une double lecture de ses entrées, à un intervalle de temps paramétrable défini dans la constante *Tps\_Interver*.

La sortie de la zone se fait par appel à la routine *Sortie\_Interverr*, qui contient le point, ce qui permet de monter la sortie exactement sur le point de passage (routine triggée).

Pour information

## 6 – Création des trajectoires de Reprise

Une trajectoire de reprise est utilisée pour se recycler lorsqu'on a abandonné un programme "sauvagement", en plein milieu.

Admettons qu'il y ait eu un problème sur l'encollage. L'opérateur a arrêté le robot, est passé en manuel, puis l'a ramené près de lui au joystick pour vérifier la pièce. Il s'agit maintenant de repartir, avec la pièce, et sans se décybler avec l'automate. Deux cas sont alors possibles:

1 - L'opérateur n'a pas fait un "reset" du programme, en validant "SPECIAL", "DEBUT PROGRAMME". Le programme est donc toujours à l'endroit où l'arrêt a eu lieu. Si l'opérateur valide "EXEC", le robot rejoint par le plus court chemin (après confirmation de l'opérateur) la position où l'arrêt initial a eu lieu, à vitesse réduite, puis poursuit son cycle. Dans ce cas, aucun traitement particulier de la reprise n'est à faire.

2 - L'opérateur a fait un "Reset" du programme. Celui-ci ne sait donc plus à quel endroit l'arrêt a eu lieu. Toutefois, le programme sait quel **cycle** et quelle **trajectoire** ont été abandonnés, ainsi que les **zones** qui étaient attendues et surveillées.

Au moment du redémarrage en production, et après avoir rejoint le REPLI à vitesse réduite, le robot teste si le code automate présent sur les entrées n'a pas varié. S'il est identique, un menu propose "REPRISE DU CYCLE" ou "RAZ MEMOIRE DE TRAVAIL".



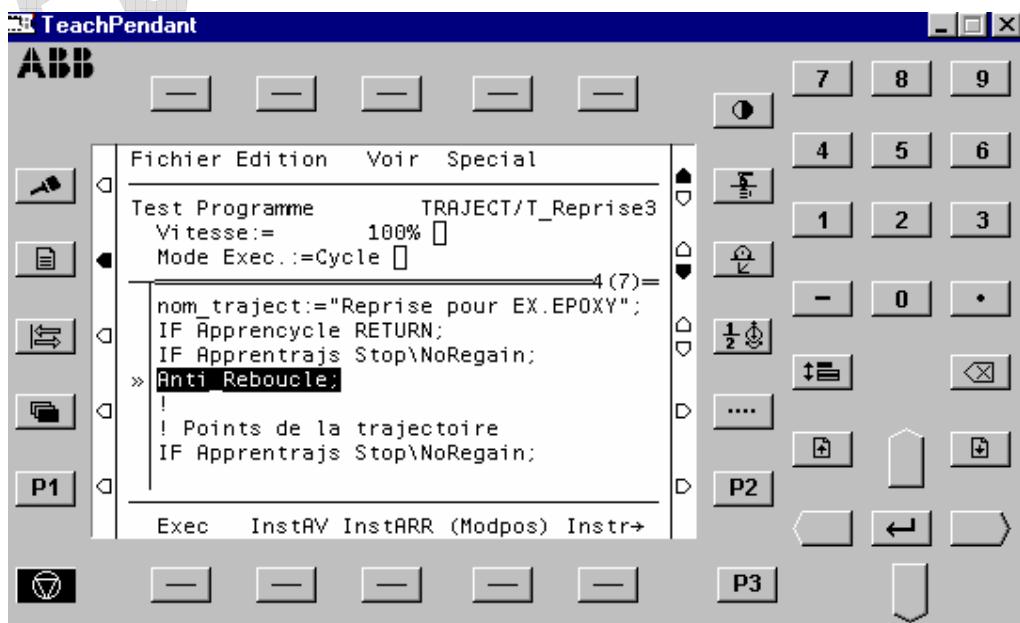
Si le choix "Reprise" est validé, le robot **attends que toutes les autorisations de zone** qui étaient présentes avant l'arrêt le soient, puis exécute la trajectoire de reprise **de la trajectoire abandonnée**, puis poursuit son cycle normalement. Pour cela, il est donc nécessaire d'avoir,

au préalable, créé le petit tronçon qui permet d'aller, avec la pièce, du Repli au début de la trajectoire d'encollage, donc en ayant "sauté" la prise pièce.

Pour créer ces trajectoires, rien de plus simple: A partir du menu principal, validez "APPRENTISSAGE","MODIF TRAJECTOIRES","TRAJECTOIRES DE REPRISE".



Le Standard vous affiche alors la liste des trajectoires de TRAVAIL existantes. Choisissez la trajectoire pour laquelle vous souhaitez créer la reprise. Par exemple: "Encollage Pavillon". Comme pour les trajectoires "normales", le programme s'arrête automatiquement dans une routine, qui ressemble en tous points à une trajectoire. Si vous êtes attentif, vous remarquerez toutefois que le programme est cette fois dans une routine qui s'appelle "T\_RepriseX", où X est...simplement le numéro de la trajectoire de Travail que vous avez sélectionnée !



Vous pouvez donner un nom à votre trajectoire de reprise (en documentant la variable nom\_traject:="...."), mais ce nom n'apparaîtra que fugitivement à l'écran, uniquement pendant l'exécution de la trajectoire.

En résumé, chaque trajectoire de travail possède sa propre trajectoire de reprise, qui porte le même numéro. Ainsi, si la trajectoire n°12 est abandonnée, c'est la trajectoire de reprise n° 12 qui sera exécutée AVANT la trajectoire 12 proprement dite. Naturellement, cette gestion est totalement transparente pour le programmeur: A aucun moment, il n'a besoin de savoir que c'est la trajectoire 12 ou 45 qui sera appelée. Il ne connaît que le **nom** du travail. (Ex: "Encollage").

Attention: pour la bonne gestion des Hors zone, une trajectoire de reprise doit se terminer **avant** le premier point de la trajectoire à reprendre, donc Hors Zone.

#### **Une trajectoire de reprise va du REPLI au dernier point de la trajectoire qui précède celle à reprendre.**

NOTA: Il n'est pas nécessaire de créer la position de REPLI dans la trajectoire de reprise: le Standard assure que le robot est bien dans cette position pour lancer la reprise.

Le contenu d'une trajectoire de reprise n'a rien de particulier: on ne trouve que des instructions de mouvement élémentaire (MoveJ, MoveL...) SAUF:

Si un traitement particulier est nécessaire, comme dans le cas de la Manutention. En effet, dans le cas d'une prise pièce par exemple, le travail a pu être interrompu AVANT la prise ou APRES la prise. Comment savoir ou redémarrer ?

Dans ce cas, il faut introduire l'appel aux fonctions de reprise spéciales, comme suit:

```
PROC T_Reprise1()
nom_traject:="Reprise pour Prise pièce";

If Rep_Av_Prise(di_Pres_Piece1) then
  MoveJ...
  MoveJ...
endif
ENDPROC
```

La fonction **Rep\_Av\_Prise** vérifie si le(s) détecteur(s) de présence pièce spécifié est à 1, puis, selon le cas, propose le choix "Reprendre avant la prise pièce?" ou « Reprendre après la prise pièce ? ».

Si tous les présence pièce spécifiés sont à 0 et que l'opérateur réponds "OUI", la fonction renvoie VRAI. Ce sont donc les points créés après le IF Rep\_Av\_Prise() qui seront exécutés, c'est à dire la prise pièce.

Si tous les présences pièce spécifiés sont à 1, il faut donc effectuer une reprise APRES prise pièce, celle-ci étant déjà dans le préhenseur. Mais en fait, cette reprise correspond exactement à la reprise...de la trajectoire suivante dans le cycle ! Dans ce cas, c'est donc la reprise de l'encollage, par exemple, qui sera réellement exécutée !

On voit que dans ce cas, les points à créer dans la trajectoire de reprise de la prise pièce ne concernent que le cas "AVANT PRISE". Le cas "APRES PRISE" se fera "tout seul".

Si l'état des présences pièce est incohérent (il en manque 1 sur 2...), le redémarrage n'est possible qu'après être revenu dans un état « normal » : tous détecteurs à 1 ou tous détecteurs à 0.

On trouve la même chose pour une trajectoire de dépose, avec la fonction "**Rep\_Av\_Depose**".

La fonction **Rep\_Av\_Process()** peut être utilisée pour vérifier la cohérence des présences pièce avant la reprise d'un process quelconque.

Ces trois fonctions reçoivent comme paramètre obligatoire le nom du présence pièce à tester. Deux détecteurs supplémentaires peuvent être déclarés en paramètres optionnels.

#### Gestion des Hors\_zone sur reprise "après" ou "avant":

Si l'on fait une reprise "Avant":

C'est le cas simple, toutes les zones qui étaient attendues le restent.

Si des attentes de zones existent dans le cycle AVANT la traj à reprendre, elles sont ignorées.

Si l'on fait une reprise "Après":

Toutes les zones prises par Appelletraj de la traj abandonnée **et par la trajectoire elle-même** sont libérées.

Si une zone a été prise dans un cycle (zone globale), elle est libérée si et seulement si la trajectoire qui sera réellement reprise est en dehors de la zone en question.

On voit que, mis à part ce petit appel de fonction pour les cas "spéciaux", **aucune ligne de programme n'est à écrire** pour assurer la programmation complète d'un site.

Il existe tout de même un ensemble de procédures et fonctions du standard que le trajectoriste peut utiliser dans ses trajectoires. Leur liste exhaustive est donnée en annexe. Toutefois, leur utilisation devrait être limitée.

#### Reprise sur bifurcation de trajectoire:

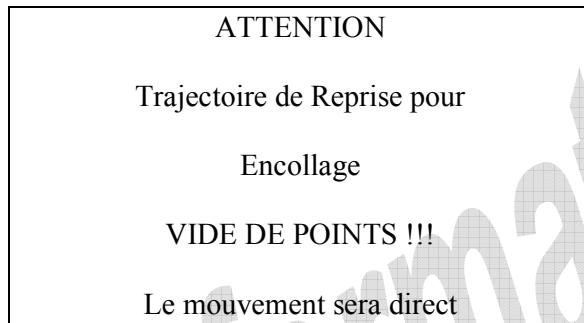
En cas de bifurcation de trajectoire, c'est bien la trajectoire abandonnée qui sera reprise. Dans ce cas, la trajectoire de reprise réellement exécutée sera celle correspondant à la première trajectoire déclarée dans l'appel de "Appelletraj".

Il n'est donc pas utile de créer une reprise pour la deuxième trajectoire de la bifurcation, puisque le point d'entrée des deux trajectoires est identique.

### **Et si la trajectoire de reprise est vide ?**

Il peut tout à fait arriver qu'une trajectoire de reprise n'aie pas été programmée: soit par oubli, soit volontairement, parce que la reprise peut être directe à partir du Repli.

Le Standard gère ce cas: si la trajectoire de reprise est vide de points ou inexistante, un message de confirmation est proposé sous la forme:



En cas de validation Opérateur, le robot rejoint directement le début de la trajectoire à reprendre. Sinon, le programme s'arrête.

### **7 – Crédation des trajectoires de substitution**

Cette fonction est utilisée essentiellement pour les robots dits "de ligne". Elle consiste à faire exécuter par un robot tout ou partie du travail d'un autre robot, en panne. Si l'on souhaite réaliser une substitution, il suffit d'insérer dans le cycle l'appel de la routine:

Substitution 2;

A l'endroit du cycle où l'on souhaite effectuer le travail d'un autre robot.

En cas "normal", cette routine ne fait rien. Par contre, si vous avez validé le choix "SUBSTITUTION" par le menu "REGLAGES", cette routine appellera la trajectoire T\_SubstX, selon votre choix. En effet, 10 cas de substitution sont prévus, mais dont seulement 3 sont créés par défaut:

- Robot 1.
- Robot 2.
- Robot 3.

Depuis la version 3.68, ces noms peuvent être modifiés (manuellement) en changeant le contenu du tableau de strings **NomRobSubst**. Vous pouvez mettre, par exemple, "R\_624" ou "Toto"...

Cette notion permet de mettre un robot en mode "substitution" de façon simple: au moment de la création des trajectoires de substitution, par le menu "APPRENTISSAGE", le standard vous demande pour quel robot vous souhaitez créer la substitution: pour le robot "R 1", "R 2", ou "R 3". Validez votre choix, le programme se retrouve automatiquement dans la routine T\_Subst1, 2 ou 3 selon le cas. Vous pouvez créer vos points.

Lors de l'exploitation, le simple fait de valider le choix "MODE SUBSTITUTION" du menu "REGLAGES" provoquera l'exécution de la trajectoire du robot "R 1", "R 2", ou "R 3", selon le choix que vous aurez fait.

Si vous décidez de créer les trajectoires de substitution par "copier/coller" direct des trajectoires d'un autre robot (sous réserve que les repères Objet soient corrects), sachez que:

- T\_Subst1 corresponds au robot "R 1" (*Resp le nom 1 du tableau Nomrobsubst*).
- T\_Subst2 corresponds au robot "R 2". (*Idem pour le nom 2*).
- T\_Subst3 corresponds au robot "R 3". (*Idem pour le nom 3*).

Et ainsi de suite jusqu'à 10.

ATTENTION: les trajectoires T\_Subst4 à T\_Subst10 ne sont pas créées d'origine, pour ne pas encombrer inutilement la mémoire. Si l'on souhaite les utiliser, il faut donc les créer par le menu Routine / Dupliquer.

Dans l'appel de la routine SUBSTITUTION, le paramètre envoyé **n'est pas le numéro de trajectoire**. Il s'agit du **numéro de zone**.

ATTENTION: Chaque robot ne peut exécuter qu'une seule substitution à la fois, à choisir parmi les 10 choix ci-dessus. On ne peut pas déclarer un robot comme se substituant à la fois au robot d'en face, de droite et de gauche.

Par contre, chaque robot peut à tout moment se substituer à n'importe lequel des 10 choix possibles.

## **8 – Exécution de routines sur point de passage**

Les procédures de base *MoveJSync*, *MoveLSync*, *MoveCSync* permettent d'appeler une routine lorsque le robot passe sur la projection du centre de la sphère définie sur le point.

La routine étant appelée par une interruption, elle ne doit pas comporter de traitement « lourd ». En particulier, elle ne doit pas être bloquante, en attente d'événements. Sinon, il y a génération du défaut « **Interrupt Queue Full** » avec coupure de la baie obligatoire et quelques fois, re-démarrage à froid.

La routine appelée sera généralement utilisée pour monter des sorties.

Nota : Plusieurs bogues ayant été relevés dans l'utilisation de ces routines système, il est plutôt conseillé de les ré-écrire soit même, en déclarant une interruption connectée sur une sortie , elle-même commandée par une instruction TriggIO...

## **9 – Mise au point et Optimisation**

### *a - Mise au point:*

Il est bien pratique, en phase de mise au point de l'installation, de pouvoir tester l'ensemble des mouvements robot ainsi que le dialogue automate, mais sans produire (ni abîmer) de pièces réelles. (mode Déverminage).

Ce mode est prévu mais n'est pas accessible par un menu direct, car d'utilisation restreinte par rapport à la durée de vie de l'installation.

Pour l'activer, il suffit de forcer le booléen *Sans\_Piece* à True.

Dès que ce flag est activé, le robot peut être lancé en fausse production, avec le comportement suivant:

- L'état des clés Avec/Sans process n'a aucune influence sauf pour le soudage par points. Les modes Avec / sans piece et Avec/sans soudure sont indépendants.
- Les process nécessitant la présence d'une pièce réelle sont simulés (la pince à clincher se ferme, mais ne clinche pas, le pistolet d'encollage n'est pas piloté, etc)...
- En soudure par points, il est possible de souder réellement selon l'état de la clé.
- En Clinchage, la pince se ferme, mais sans amplification de pression.
- En Rivetage, une simple tempo de 5 s remplace la commande de la pince.
- Les présences pièce sont vérifiées à 0.
- Tous les mouvements préhenseur sont exécutés, et les contrôles correspondants **sont effectués**. Attention, car cela peut gêner dans le cas où un contrôle est présent avec pièce, mais absent sans pièce...Mais la sécurité impose de vérifier l'état réel du préhenseur.
- Tout le dialogue Automate se passe exactement comme en mode normal.

### **ATTENTION:**

Une détection automatique du passage au mode *Avec Piece* est prévue: Si le robot tournait en mode *Sans Piece* et qu'une pièce réelle se présente, il y a détection de la pièce réelle dans les applications:

- Soudure (grâce au contrôle DEC, s'il existe et qu'il est activé).
- Manutention (grâce aux présences pièce).

Ceci pour éviter qu'une pièce réelle soit évacuée sans avoir été traitée, du fait que le robot était resté en mode *Sans\_Piece*.

Dans ce cas, un menu s'affiche permettant de confirmer le passage au mode *Avec\_Piece*. Si aucune pièce réelle n'est présente, cela signifie un défaut de programmation des trajectoires ou de conception de l'outillage, à savoir:

- La pince touche l'outillage lors d'une soudure, entraînant un défaut DEC. (modifier le point ou l'outillage).

- Un présence pièce passe à 1 alors qu'il ne devrait pas → Revoir réglage préhenseur ?

On voit que le mode *Sans\_Piece* (Déverminage), permet non seulement de tester l'ensemble des cycles et trajectoires et la bonne gestion des Hors Zone, mais également de détecter des défauts de programmation ou de conception quelquefois difficiles à détecter autrement. (Point de soudure difficile d'accès et où l'on ne voit pas l'électrode inférieure...)

#### *b - Optimisation de la mémoire:*

Il est judicieux de minimiser la mémoire utilisée par les actions suivantes:

- Si le robot n'a pas de pince électrique, supprimer tous les fichiers "SWxxx" du Ramdisk.
- Supprimer les fichiers archive paramètres.
- Vous pouvez diminuer sensiblement la taille du fichier TRAJECT standard en supprimant toutes les trajectoires inutilisées, notamment:
  - Les trajectoires de changement d'outil s'il n'y a pas de changeur (T\_Prise, T\_Depose).
  - Les trajectoires de purge s'il n'y a pas d'encollage (T\_Purge).
  - Les trajectoires de rodage et de changement électrodes s'il n'y a pas de soudure.
  - Les trajectoires de Contrôle et de Nettoyage si ces fonctions ne sont pas utilisées.
  - Les trajectoires de substitution si cette fonction n'est pas utilisée. (T\_SubstXX).
  - Les points de début et de fin de container s'il n'y a pas de dépalettisation.

Vous pouvez également supprimer toutes les déclarations de trajectoires et de cycles non utilisés, p.e. cycles 4 à 31 puisqu'en utilise que 3. Le standard sait gérer les appels à des routines inexistantes. Cela améliorera sensiblement le temps de réponse du programme, et permettra de plus de faire disparaître les faux défauts système lors d'une sauvegarde sur disquette.

#### *c - Optimisation des temps de cycle:*

Il n'y a pas grand chose à faire pour optimiser les temps de cycle. Si les derniers points de toutes les trajectoires sont en points de passage, le robot ne s'arrête jamais tant que les autorisations sont présentes. Si par contre une autorisation manque, le robot s'arrête sur le point et attend. Le message "Point de passage dégénéré en point d'arrêt" apparaît alors dans les archives, sans autre conséquence.

Dans ce cas, seul un travail sur l'automate peut éventuellement remédier au problème.

Toutes les attentes d'événements "standard" possèdent le même fonctionnement (Ex: test Présence Pièce). En auto, l'attente s'interromps dès que l'événement attendu est présent. Passé un délai, un message d'erreur est affiché, avec notification dans le fichier de suivi. Tant que l'on reste en Auto, il n'est pas possible de "sauter" l'événement. Par contre, en Manuel, un menu s'affiche permettant d'abandonner l'attente et de poursuivre l'exécution, ou de re-tester.

Avec ce mode de fonctionnement, l'attente est nulle si l'événement attendu est déjà présent. Il est donc possible de tester les présences pièces, par exemple, sur un point de passage: si les contrôles sont ok, le robot ne s'arrêtera pas. Si l'un vient à manquer, le robot s'arrêtera jusqu'à ce que celui-ci devienne correct.

**Attention:**

Dans ce cas, le message "Point de passage dégénéré en point d'arrêt" apparaît dans les archives, ce qui est normal.

Un point toutefois doit être examiné avec la plus grande attention: il s'agit de la documentation des données de masse et d'inertie de l'outil embarqué. En effet, la baie S 4 possède un modèle dynamique qui adapte les vitesses et accélérations des axes, en temps réel, en fonction des contraintes définies et mesurées. Si vous déclarez un outil avec une masse de 100 Kg alors que l'outil réel n'en pèse que 50, vous serez pénalisé en temps de cycle. A l'inverse, vous obtiendrez des défauts d'exécution.

Pour plus de détail sur la définition des masses et inerties, on se reportera au manuel du robot, selon son type. (L'IRB 6400 possède, par exemple, une mesure automatique de ces données, ce dont ne bénéficient pas les autres modèles de la gamme).

## **10 – Gestion des cames de protection Opérateur.**

Ces cames servent à protéger l'opérateur lorsqu'il y a un poste de chargement / Déchargement manuel. Cette protection est assurée par le câblage, mais le Standard effectue une vérification anti-shunt. En outre, il contrôle le battement des détecteurs à chaque cycle.

Pour une question de sécurité, l'entrée robot est à 0 lorsque le robot est dans la zone opérateur et à 1 le reste du temps.

Le trajectoriste doit insérer dans ses trajectoires l'instruction Verif\_Came Sur\_CameX lorsque le robot est dessus, et Verif\_Came Hors\_CameX lorsqu'il la quitte.

La persistante num NB\_CAMES\_CYCLE permet de déclarer le nombre de cames que le logiciel doit surveiller (de 0 à 3). La prise en compte de cette valeur est immédiate (pas de redémarrage à faire).

**ATTENTION :** Les trois cames sont prédéclarées dans les entrées sorties, mais seules celles déclarées au moment de l'installation du logiciel sont affectées sur une carte physique. Les autres sont déclarées virtuelles. Si vous augmentez le nombre de cames à surveiller, pensez à redéclarer les signaux d'entrée en physique.

En cours de cycle, si l'état d'une came ne correspond pas à l'état attendu (lors de l'exécution de l'instruction Verif\_Came), un défaut est généré.

A la fin du cycle, le standard vérifie si l'état de toutes les cames déclarées a bien été vu à 1 et à 0. Si ce n'est pas le cas, le défaut « Incohérence Cames » est généré.

Si un cycle ne passe pas par toutes les cames déclarées, on évite de générer ce défaut en déclarant la routine Verif\_Came SUR\_CAME\_SIM et Verif\_Came HORS\_CAME\_SIM.

Il faut tenir compte de ces cames dans les trajectoires de reprise : Admettons qu'un cycle comporte les trajectoires 1, 2 et 3, et que la traj 1 passe sur la came 1. Si l'on fait une reprise de la traj 2, on ne passera pas sur la came, d'où erreur à la fin du cycle. Il faut dans ce cas écrire Verif\_Came SUR\_CAME\_SIM dans la trajectoire de reprise de la traj 2.

Pour information

## IV – COMMANDES MANUELLES

Les actions « manuelles » sont une aide à l’exploitation du site. Elles permettent de réaliser une fonction plus ou moins complexe, sans avoir à forcer les sorties. Une action ne peut être effectuée qu’en mode manuel et moteurs coupés (gâchette relâchée), sauf pour les actions sur pince électriques où les asservissements sont nécessaires. Elles sont autorisées En Production.

Il existe un grand nombre d’actions manuelles possible. Malheureusement, seules les 5 touches du pupitre P1 à P5 peuvent être interceptées par programme lorsque celui-ci est à l’arrêt. Le fonctionnement suivant a donc été défini :

### **1 – Si le robot est en mode Manuel et Hors asservissements :**

Un appui sur la touche P1 permet de configurer l’action qui sera exécutée lors d’un appui sur les autres touches. Un menu apparaît, donnant la configuration actuelle :



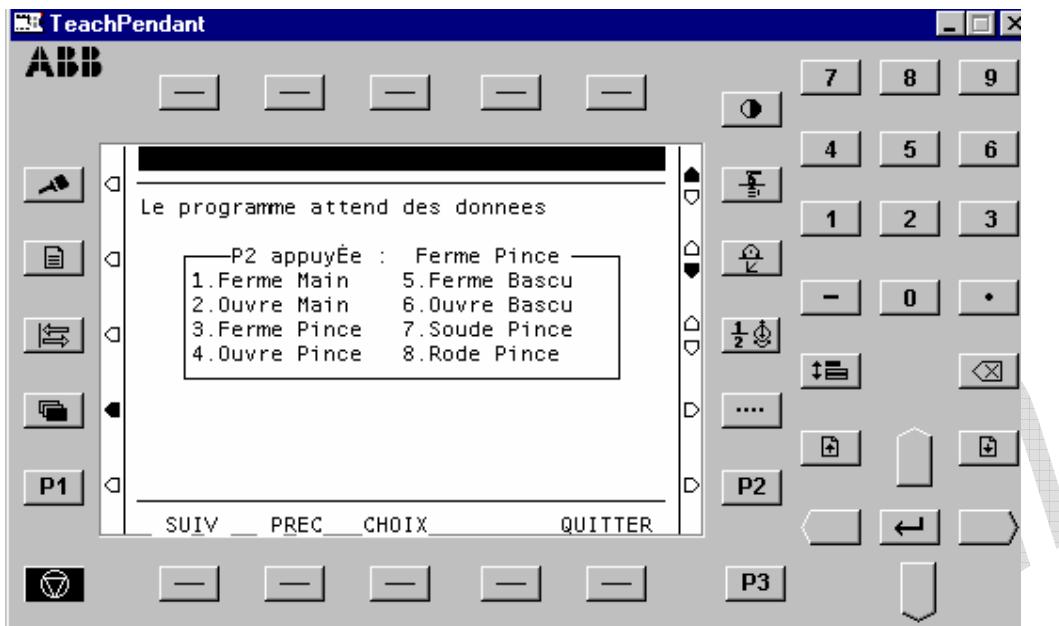
Validez le choix de la touche à configurer. Vous pouvez alors sélectionner le type d’action à réaliser pour cette touche :



Selon le type d'action, certains paramètres doivent être fournis. Par exemple, pour l'action « Ferme Pince », si plusieurs pinces existent, il faut préciser le numéro de pince. Il faut également préciser le n° de programme à envoyer à la CPS pour la consigne de pression.



Le choix ACTION permet de changer l'action affectée à la touche :



Si vous affectez une action à une fonction non existante, par exemple la purge Debit alors qu'il n'y a pas d'application Debit installée, un simple message avec acquittement apparaît indiquant que la fonction n'est pas disponible. (lors d'une tentative d'exécution).

Une fois la configuration terminée, un appui sur l'une des touches P 2 à P5 exécute, après confirmation, la fonction demandée.

## **2 – Si le robot est en mode Automatique :**

Les actions manuelles sont donc interdites. Toutefois, et si les asservissements sont montés, un appui sur la touche P1 permet de visualiser les éventuelles surveillances inhibées, ainsi que le mode de marche des différents process (Sans Piece, Sans Soudure...), et les alarmes en cours .

*Depuis la version 3.68, cette fonction affiche également la version du Standard et l'état de la supervision de mouvement( Active, Inactive, et sa valeur).*

Si une application Encollage Mastic ou Debit est présente, cette fonction affiche également le temps restant avant la prochaine purge, et la durée de la dernière purge effectuée.

Lorsque l'on passe en Automatique mais que l'on est Hors asservissements, le message "APPUYEZ SUR DEPART" s'affiche en permanence. Cet affichage peut être gênant si l'on souhaite, à ce moment, accéder à d'autres menus (par ex pour modifier la vitesse générale). Dans ce cas, un appui sur P1 permet d'inhiber cet affichage pendant 10 secondes.

NOTA : L'intégrateur doit écrire lui même les routines qui seront appelées lors de demande d'exécution manuelle pour les cas non standard, notamment la gestion des préhenseurs, trop variables selon les sites.

Ces routines existent déjà, mais elles ne font rien. Elles s'appellent *Ferme\_Main1*, *Ouvre\_Main1*, *Ferme\_Main2* et *Ouvre\_Main2*, *Ferme\_Main3* et *Ouvre\_Main3*. Elles sont dans le module ACTUSITE.

Le module ACTMSITE contient la déclaration des appels aux routines exécutables en mode manuel. Il est modifiable. Toutefois, les modifications à apporter devraient être minimales. Celui-ci contient déjà en effet la quasi totalité des actions possibles, à savoir:

- Ferme\_Main x
- Ouvre\_Main x
- Ferme\_Pince x
- Ouvre\_Pince x
- Ferme\_Bascu x
- Ouvre\_Bascu x
- Soude\_Pince x
- Rode\_Pince x
- Purge\_Debit
- Purge\_Pisto x
- Teste\_Mastic x
- Clinche
- Soude\_Tete x
- Avance\_Tete x
- Recule\_Tete x
- Programme\_Tuck x
- Avance\_Masse
- Recule\_Masse
- Avance\_Fil
- Arret\_Fil
- Debut\_Decrasse
- Fin\_Decrasse
- Teste\_Gaz
- Premiere Init Pince (Electrique).
- Init Pince.
- Rivetage.
- Avance broche rivetage.
- Recul broche rivetage.
- Test Lampes

La fonction Test Lampes (nouvelle depuis V 3.4) permet d'allumer les voyants Repli, Depart Cycle et Verrine pendant 2 secondes, puis de restituer leur état précédent.

ATTENTION: si vous modifiez ce module, faites attention au cas du changeur d'outil ! En effet, dans le cas du changeur, la cohérence entre l'action demandée et l'outil réellement présent à l'axe 6 est vérifiée par le programme. Regardez attentivement le début de la routine **am\_Action\_Exec** avant de la modifier, ainsi que les indices des actions dans le tableau...

Il est préférable de modifier ce module par ajout d'appels supplémentaires, plutôt que par suppression des appels existants.

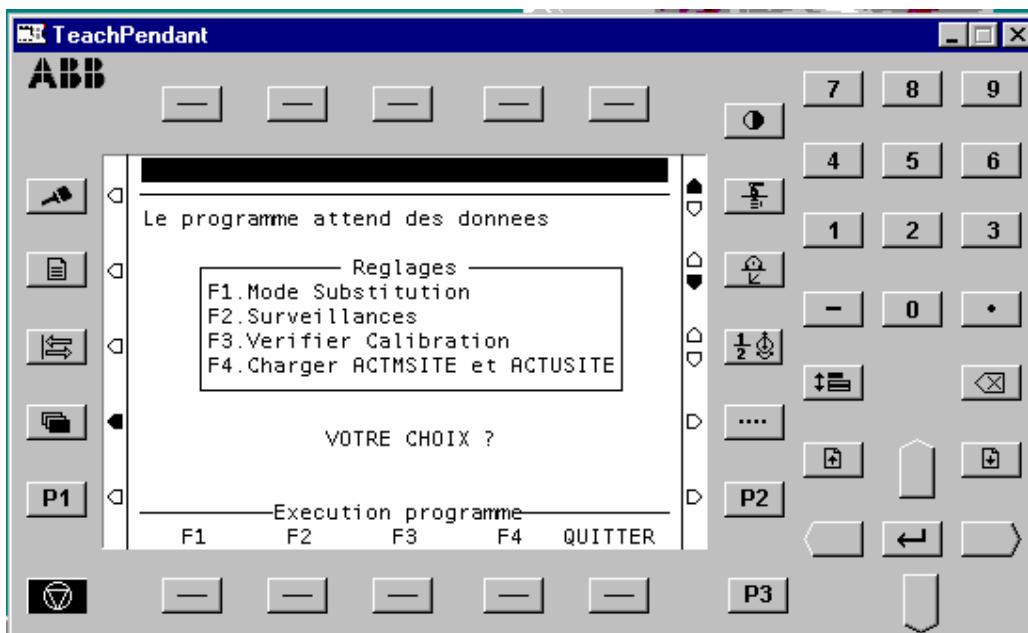
Si vous modifiez la **configuration** des actions manuelles (par appui sur la touche P1), pensez à sauvegarder les modifs par le menu "SAUVEGARDES","SAUVEGARDER LES MODIFS".

Un appui sur la touche P2 alors que le robot est En Production inhibe le rafraîchissement de l'écran de production pendant 60 secondes ou jusqu'à un nouvel appui sur P2.

Pour information

## V – MENU REGLAGES

Ce menu, accessible à partir du menu principal, en Auto ou en Manuel, permet les accès suivants:



### 1 - Substitution:

Le choix 1 (SUBSTITUTION), vous permet de déclarer que le robot sur lequel vous êtes doit dès maintenant remplacer un robot de la ligne déclaré en panne. 10 choix sont possibles, dont seulement 3 sont déclarés par défaut: "R 1", "R 2", "R 3". Ces choix correspondent évidemment à ceux du mode Apprentissage. Si vous validez un choix, et si la trajectoire correspondante a été créée, le robot exécutera la trajectoire de remplacement à l'endroit où l'instruction **Substitution** a été placée, dans un cycle ou une trajectoire. Une ultime confirmation est demandée pour valider ce mode.

Lorsque le mode Substitution est en cours, le menu se modifie pour ajouter le choix

### SUPPRIMER SUBSTITUTION

Une validation de cette action permet de revenir au mode "normal". C'est à dire que les instructions **Substitution** placées dans un cycle ou une trajectoire n'auront plus d'effet.

En résumé, une instruction **Substitution** placée dans un cycle exécute la trajectoire correspondant au robot déclaré en panne grâce à ce menu.

## **2 - Surveillances:**

Le deuxième sous menu du menu Réglages permet d'activer ou de désactiver les surveillances, de façon rapide mais globale à chaque application. Il sera utilisé notamment en phase de mise au point, si le câblage n'est pas terminé, ou en production si un détecteur est en panne. Pour chaque application installée, l'écran affiche l'état des surveillances: Active ou Inactive.



Le choix d'une option permet d'activer / de désactiver **toutes** les surveillances liées à l'application en question. C'est une bascule. Si par exemple on inhibe les surveillances de la soudure, les entrées suivantes ne seront plus testées:

- le Présence Pince.
- le Présence Air.
- le Présence Eau.

Par contre, les détecteurs de P.O et G.O continueront d'être testés.

S'il s'agit d'une application Encollage Debit ou Mastic, le Présence Pistolet ne sera plus vérifié. Par contre, le fonctionnement du groupe ou du comptage continueront à l'être.

Si une surveillance est inhibée, tout lancement En Production provoque l'affichage d'un message bloquant indiquant cette inhibition, avec validation obligatoire.

On voit donc que l'utilisation de ce menu permet de se débarrasser facilement et rapidement d'un problème de panne ou de câblage, mais ne peut rester que **provisoire**, car contraignant à chaque lancement En Production.

Si l'on souhaite désactiver momentanément une surveillance particulière, il faut modifier manuellement l'état des flags (booléens) par le menu système "Voir", "Données", choisir le

type "bool", puis modifier le flag souhaité. Leurs noms sont suffisamment explicites pour ne pas nécessiter d'explications supplémentaires ici. (Ex : sr\_SURV\_EAU pour le circuit d'eau).

#### NOTAS:

- Le fait de valider un choix dans le menu écrase l'état des flags qui auraient pu être positionnés manuellement auparavant. Ex: si vous positionnez le flag "sr\_CTRL\_AIR1" à FALSE parce que le contrôle d'air ne marche plus, et que vous validez le menu Activation Surveillance de la soudure, le flag en question sera remis à TRUE, ce qui provoquera un défaut si le contrôle n'est toujours pas opérationnel. En fait, ce menu est "brutal": c'est tout ou rien, il n'y a pas de demi mesure.

- Dans le cas de l'application SR 2 pinces et s'il y a plusieurs débit-litres et plusieurs présence Air, toutes les surveillances de toutes les pinces sont activées / désactivées simultanément.

Notez bien que ce menu ne fait qu'inhiber les surveillances: les actions sont effectuées quand même, mais sans contrôle. Ex: la mise en marche de l'eau sera faite, mais le débit litre ne sera pas testé.

L'état des surveillances inhibées peut être visualisé à tout moment, en production, par un appui sur la touche P1. (Voir *Actions Manuelles*). Si l'on est Hors Production, seul le menu Réglages permet de visualiser cet état, d'une façon globale à chaque process.

### **3 - Calibration / Definition des Repères:**

#### *a- Calibration*

Le troisième choix du menu Réglages permet de vérifier la calibration du robot. Il ne peut être utilisé qu'en mode Manuel. Après confirmation, le robot rejoint à vitesse réduite la position 0 de tous ses axes, ce qui permet de contrôler si un décalage a eu lieu, grâce aux repères placés sur chaque axe.

Ce point est maintenant modifiable pour tenir compte des sites où le robot ne peut physiquement rejoindre le 0 de tous les axes. Le point est de type JointTarget et se nomme

*PabsCAL\_POS*

Et est déclaré dans le module TRAJECT.SYS.

#### *b - Définition des Repères:*

Non spécifié à ce jour, utiliser les routines standard du système.

### **4 - Rechargement des actions manuelles:**

Le fichier ACTMSITE ne peut être modifié que sur PC, et jusqu'à présent son rechargement exigeait de faire un P-START.

La version 1.8 du standard pallie à cet inconvénient : le dernier choix du menu REGLAGES permet de provoquer le rechargeement des modules ACTMSITE et ACTUSITE par l'arrière-plan, et ACTUSITE seul par l'avant plan. Ces deux fichiers doivent être présents sur la disquette. Ils ont pu y être transférés auparavant par le menu « Sauvegarde des modifs ».

Après confirmation, le programme **copie** ces deux fichiers sur le Ramdisk (qui est donc automatiquement à jour), puis les charge en mémoire d'exécution.

Il n'est donc plus du tout nécessaire de faire un P-START, ce qui exigeait de disposer de la disquette standard.

Toutefois, si une erreur de syntaxe est détectée dans l'un des modules, il y a malheureusement arrêt de la tâche de fond avec obligation de faire un P-Start avec un fichier corrigé.

Si on l'utilise, ce nouveau mécanisme entraîne qu'il ne faut **jamais** faire « Fichier », »Ouvrir » du module ACTUSITE. Sinon, il serait chargé deux fois, ce qui entraîne une erreur. Le Standard intercepte cette erreur en vous indiquant :

« Ne peut charger ACTUSITE, faites Fichier, Ouvrir »

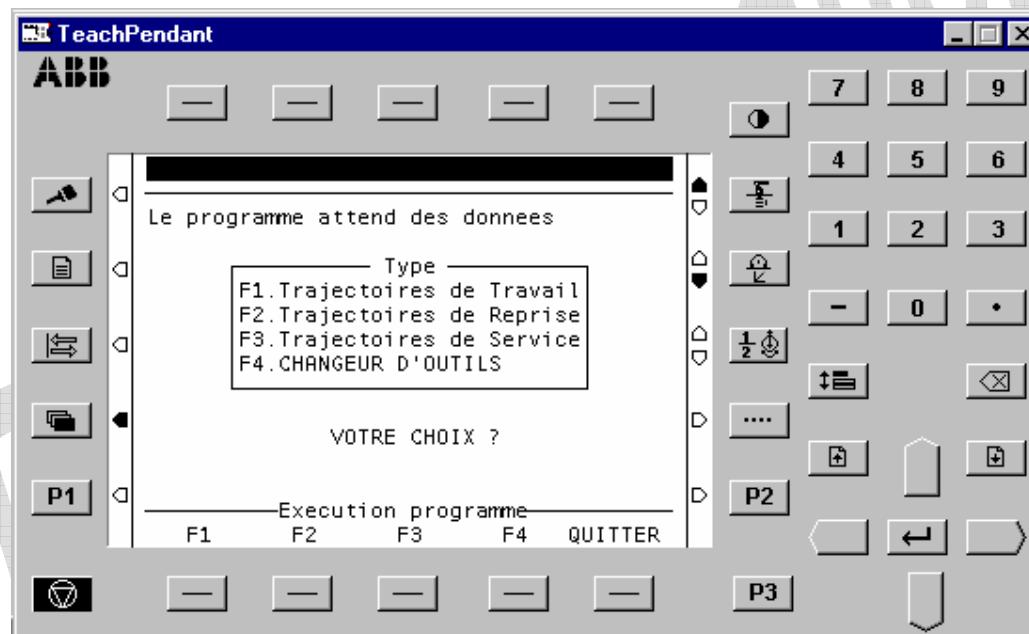
Conclusion : si vous faites, ne serait-ce qu'une fois, un chargement manuel de ce module, vous devrez ensuite **toujours** procéder de la même façon. Ceci n'est valable que pour l'avant plan. Pour l'arrière plan, il n'y a pas de problème car on ne peut pas charger manuellement des modules.

## VI - CAS DU CHANGER D ' OUTILS

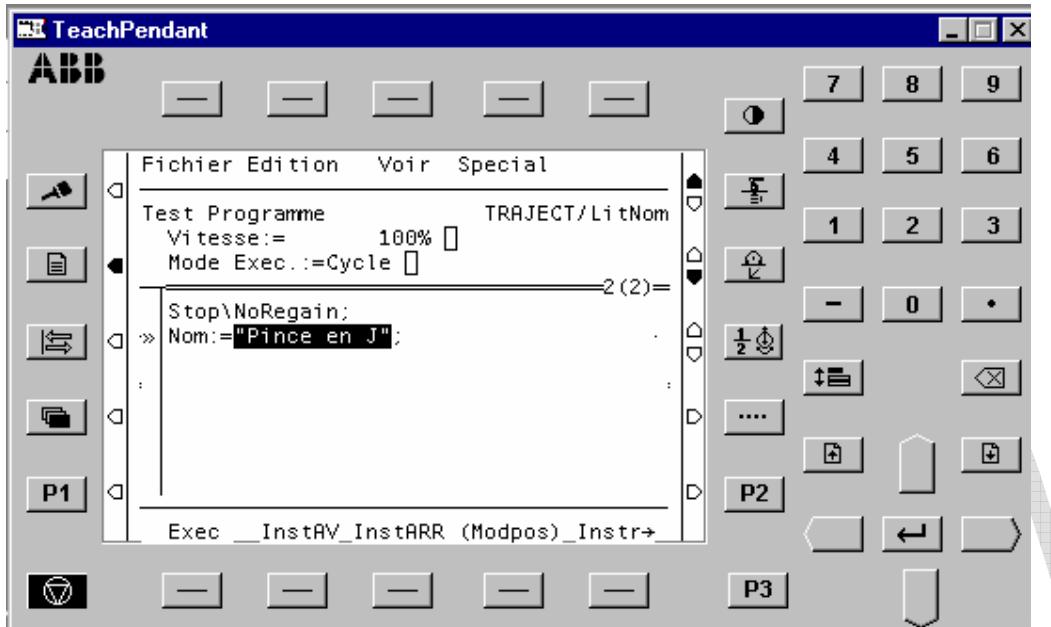
Le changeur d'outils constitue une application un peu particulière. L'application est capable d'accepter jusqu'à 7 outils de n'importe quel type. Pinces à souder, pinces à clincher, préhenseurs, pistolets d'encollage....

ATTENTION: 7 outils peuvent être gérés, mais seules les trajectoires de prise / dépose des outils de 1 à 4 existent par défaut. Si vous utilisez plus de 4 outils, il faut recopier les trajs de prise / dépose existantes.

Leur création / modification s'effectue très simplement par le menu "APPRENTISSAGE", "TRAJECTOIRES","CHANGEUR D'OUTIL". Ce menu n'existe que si l'application Changeur a été déclarée lors de l'installation.



Ce menu vous propose d'abord de donner un nom à chacun de vos outils. Validez le choix "Nom de l'Outil", puis donnez un numéro de 1 à 7. Le programme s'arrête. Vous pouvez alors donner un nom en clair à votre outil n° 1. Par exemple:



Ensuite, c'est ce nom qui apparaitra en clair dans tous les menus. Validez "EXEC". Le programme vous demande alors le type d'outil:



Choisissez le type d'outil. Cette information est capitale pour la surveillance des outils: en effet, l'application doit activer et désactiver des surveillances différentes selon le type d'outil présent à l'axe 6.

On ne peut pas quitter ce menu tant que le type d'outil n'a pas été donné.

Vous pouvez ensuite "créer" les trajectoires de prise et dépose de chaque outil. Choisissez le nom de l'outil dans la liste. Le programme vous affiche:



Une fois votre choix fait, le programme s'arrête directement dans la trajectoire demandée. (Comme d'habitude !). Vous n'avez plus qu'à créer les points.

Dans le principe, une trajectoire de prise commence un peu au dessus du ratelier et se termine au PEO. Pour une trajectoire de dépose, c'est l'inverse. Attention toutefois de vérifier que le robot peut physiquement aller directement, sans outil, de la fin d'une trajectoire de dépose au début de toutes les trajectoires de prise.

Les actions de prise et dépose (verrouillage / déverrouillage changeur, test du numéro d'outil codé sur les entrées, présence outil, dégagement du robot dans l'axe du tool0 de 100 mm) sont gérés par les routines "*Prends\_Outil*" et "*Lache\_Outil*", déjà écrites dans les trajectoires.

#### NOTA:

Dans le cas des pinces électriques, la carte Swit impose que la pince soit fermée lors de la prise ou de la dépose de la pince. Si l'opérateur a mal programmé ce point, le Standard s'en charge et force la fermeture de la pince lors de la dépose. Cela prends toutefois un peu plus de temps que si le point de dépose a bien été appris pince fermée.

Il est possible d'inhiber le dégagement automatique du robot dans le Z- du tool0 en validant le paramètre optionnel **\Sans\_Recul**. Dans ces cas, s'il s'agit d'une dépose, le standard ne peut pas tester que les contrôles sont bien tombés à 0, car la routine se termine alors que le robot est encore en contact. Il faut donc tester les entrées à 0 un peu plus loin dans la trajectoire, en appelant la routine "*Teste\_Absence*".

Avant la prise d'un outil, il est possible de s'assurer que le changeur est bien déverrouillé en appelant la routine "*DeverChangeur*", qui déverrouille le changeur et teste le contrôle.

Il est évidemment strictement interdit d'appeler cette routine dans une trajectoire de dépose, sous peine de retrouver l'outil par terre...

Ensuite, il est nécessaire d'indiquer, au début de chaque trajectoire de travail, quel outil doit être utilisé. Cela se fait simplement en insérant la routine

```
Chge_Outil "Nom de l'outil";
```

au début de la trajectoire. La variable *Nom de l' Outil* contient le nom de l'outil voulu. Par exemple:

```
Chge_outil "Pince en J";
```

provoquera la dépose de l'outil courant si ce n'est pas le bon, puis ira chercher la pince en J. Comme il s'agit d'un type Pince A Souder, la puissance soudage sera activée, de même que l'eau et l'air.

Si c'est déjà la pince en J qui est présente, rien ne se passe.

Toutefois, pour assurer une meilleure maintenabilité, il est préférable d'utiliser cette routine en faisant appel non pas au nom de l'outil lui-même, mais au nom de l'outil présent dans le tableau Nom\_Outil{7}. Ainsi, si le nom de l'outil est modifié, il ne sera pas nécessaire de modifier le programme. Si par exemple, il s'agit de l'outil n°2, on écrira:

```
Chge_outil Nom_Outil{2};
```

Et l'indice 2 du tableau de string Nom\_Outil contiendra "Pince en J".

De plus, cette écriture permet de s'affranchir des erreurs d'écriture, notamment Minuscules / Majuscules, auxquelles le programme est sensible:

Pince en j  $\Leftrightarrow$  pince en J

**ATTENTION:** Si la routine "Chge\_Outil" est appelée à l'intérieur d'une trajectoire, le changement d'outil ne se fera qu'après la réception de l'autorisation d'accès zone de la trajectoire en question.

Il est donc autorisé d'appeler cette routine depuis le cycle, donc avant la trajectoire:

```
PROC Cycle1()
```

```
    Chge_Outil "Pince en J";
    AppelleTraj 4,2;
    Chge_Outil "Prehenseur";
    AppelleTraj 5,1;
```

```
ENDPROC
```

*Avantage de cette méthode:*

Le changement d'outil n'est pas "lié" à la trajectoire elle-même. Une même trajectoire pourra donc être éventuellement exécutée avec des outils différents.

#### *Inconvénient de cette méthode:*

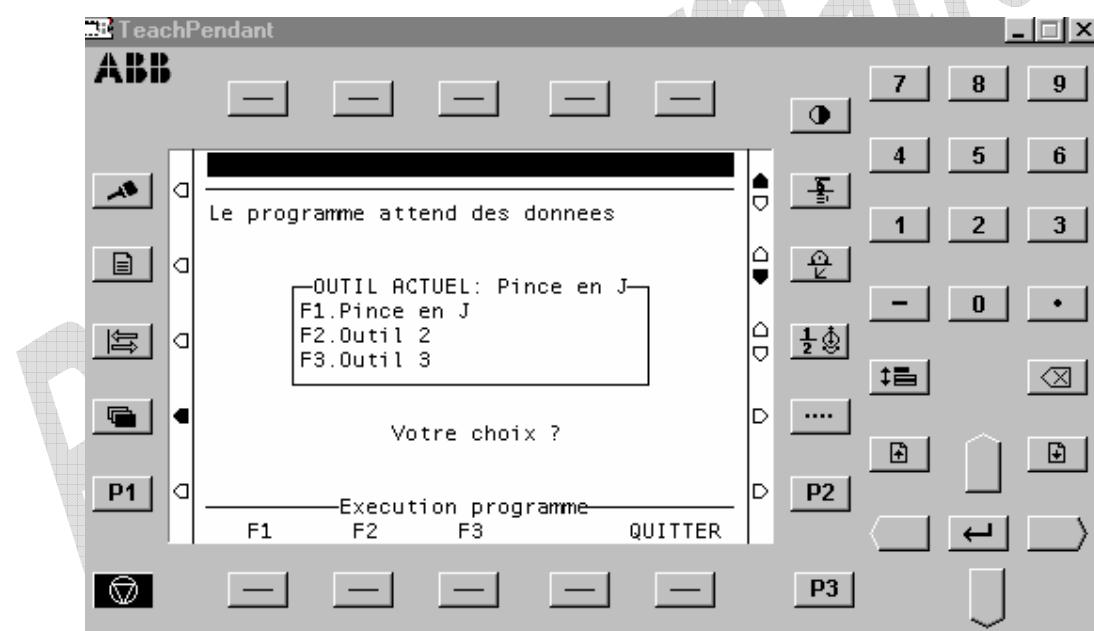
L'appel de la trajectoire en apprentissage, par le menu du standard, ne garantira pas que c'est le bon outil qui est présent à l'axe 6. Avec l'instruction dans la trajectoire, le programme va forcément chercher le bon outil.

Toutefois, la supervision de mouvement minimise ce risque.

#### **Trajectoires de service, demandes de rodage:**

##### *1- Sur demande de l'opérateur*

A tout moment, une demande de service peut être effectuée en appuyant sur le BP correspondant. Dans ce cas, à la fin du cycle en cours, un menu apparait, permettant de choisir sur quel outil on souhaite faire une opération. Exemple:



Le choix "AUTRE" n'apparaît que si une autre application, non gérée par le changeur, est présente. Exemple: Pistolet d'encollage fixe. Dans ce cas, un sous menu propose le choix "classique": voir chapitre "**Demandes de service**".

Si l'outil demandé n'est pas l'outil présent, le robot dépose l'outil en cours puis prend celui demandé. S'il s'agit d'une pince à souder, la puissance soudure n'est pas activée, ni le circuit d'eau, puisqu'on sait que l'on va effectuer du service sur cette pince.

Un menu permet ensuite de sélectionner le type de service voulu: par exemple, et si c'est une pince à souder, on aura le choix "Changement Electrodes", "Service". Si le flag

ROD\_AVT\_CHG est mis à TRUE, le robot effectuera un rodage de la pince avant d'appeler la trajectoire de changement électrodes.

S'il s'agit par exemple d'un pistolet d'encollage, le choix sera "Purge Pistolet", "Service".

Il est possible de redemander plusieurs opérations successives sur un même outil: le même menu reste affiché jusqu'à ce que l'on valide "QUITTER".

A ce moment, le robot redépose l'outil si ce n'est pas le même qu'au départ, puis va rechercher l'outil qu'il avait avant la demande de service, et l'active. (Puissance soudure, air, eau...). A la fin de l'opération, le robot se retrouve exactement dans les conditions initiales.

## 2 - Automatique

Admettons qu'une demande de rodage ou de changement électrodes venant de la CPS parvienne pendant la soudure. Oui, mais le robot doit encore déposer sa pince pour prendre une autre pince à souder, puis un préhenseur et effectuer divers travaux de manutention.

Pas de problème: le nom de l'outil courant est mémorisé par le standard au moment où la demande arrive. Ainsi, à la fin du cycle, le robot sait qu'une demande de changement électrodes sur la Pince en X, par exemple, a été faite. Il va alors chercher la pince en X, et change les électrodes.

Par contre, il garde cet outil et l'active: en effet, c'est probablement celui-ci qu'il faudra justement aller chercher au début du cycle suivant. Il y a en effet beaucoup de chances pour que l'outil nécessaire en début de cycle soit différent de celui utilisé en fin de cycle. Comme seuls 3 outils sont possibles, on a 1 chance sur 2 pour que l'outil pris soit le bon. Autant donc le garder. Si ce n'est pas le bon, le robot ira de toutes façons rechercher le bon outil après avoir reçu son nouveau code programme.

**NOTA:** Si deux ou trois pinces à souder différentes sont utilisées, deux ou trois trajectoires de rodage distinctes doivent être programmées, une pour chaque pince. Le choix de la trajectoire de rodage se fait de façon simple: Pince n°1 => T\_Rodage1, Pince n° 2 => T\_Rodage2, Pince n° 3=> T\_Rodage3. Ce choix est transparent pour le programmeur. Il doit simplement indiquer de quelle pince il souhaite créer la trajectoire de rodage.

Si une seule pince à souder existe et qu'elle porte le numéro 3, c'est la trajectoire T\_Rodage3 qui sera utilisée.

La trajectoire de changement électrodes est unique, donc commune à toutes les pinces. Il faut donc s'assurer, au moment de sa création, que le point est atteignable sans casse quelquesoït la pince présente. Il s'agit d'une position absolue: elle ne dépend pas du Tooldata courant.

## 3 – Programmation

Il faut savoir que dans le cas du Changeur d'outil, comme en fait on n'a toujours qu'une seule pince à la fois, ce sont les process unitaires qui sont utilisés. Donc Sr1p en pneumatique, SrEl en pince électrique, SrAro en pince électrique avec séquence ARO externe.

Le robot ne "connaît" qu'une seule pince à un moment donné. Ceci est différent lors d'un rodage ou d'un changement d'électrodes. En effet, il est nécessaire de razer le compteur de points de la bonne pince, et pas d'une autre. Pour cette raison, les programmes CPS de rodage sont affectés de la façon suivante:

127 → pince n° 1  
126 → pince n° 2  
125 → pince n° 3  
etc jusqu'à la pince n° 7.

Pour une pince électrique, le maximum est de 4, puisqu'une carte swit ne peut gérer que 4 compteurs au maximum.

#### ATTENTION:

Comme le numéro de programme CPS à appeler dépend du **numéro d'outil**, il est impératif que la pince 1 soit déclarée comme outil n° 1, la pince 2 comme outil n° 2, etc...Les autres outils possibles (préhenseur, tucker, etc..) doivent donc avoir des numéros **supérieurs**.

Il faut utiliser autant de **Trajectoires** de rodage qu'il y a d'outils, puisque ces trajectoires peuvent être différentes. (T\_Rodage1 = pince 1, T\_Rodage2 = pince 2, etc...).

Par contre, dans l'instruction de rodage, on utilisera toujours

Rode\_pince 1;

Quelquesoit l'outil réellement présent, puisque le robot n'en "voit" qu'une à la fois, mais "sait" de laquelle il s'agit.

Autrement dit , dans ce cas le "1" signifie, pour le programme, "pince courante".

De même pour la routine Change\_Elec: il n'est pas nécessaire de préciser le numéro de pince. Si la pince 3 est présente, c'est le programme CPS 125 qui sera appelé, et c'est le compteur n° 3 de la swit qui sera razé.

#### **Détections de changement d'outil:**

Lors d'un Special 3, le programme vérifie que l'outil présent est toujours le même, même après coupure secteur. Si celui ci a changé, (par exemple après un échange d'outil "manuel"), un message d'alerte s'affiche:

Detection Changement d'outil

L'outil present est bien "Pince en J" ?

Si les capteurs indiquent que plus aucun outil n'est présent, le message devient:

## ATTENTION

Confirmez la perte de l'outil "Pince en J".

L'opérateur doit alors vérifier visuellement si la situation réelle corresponds à celle annoncée. Si oui, il peut valider "OUI". Si non, il faut valider "NON" et vérifier les capteurs, au moins l'un d'entre eux est en panne.

### **En cas de panne d'un outil :**

En cas de panne, un outil peut être déclaré inactif en modifiant l'état du booléen *Outil\_Actif{no\_outil}* .

Dans ce cas, toutes les trajectoires utilisant cet outil sont ensuite **ignorées** lors de l'exécution du cycle.

Par contre, les trajectoires utilisant un outil différent **seront exécutées**.

Le programmeur doit donc prendre garde aux raccordements entre trajectoires dans ce cas précis.

### **Reprises de trajectoire:**

La version 3.69 du standard **permet** les reprises de trajectoire même dans le cas du changeur d'outils. Cette fonction n'était pas totalement opérationnelle dans la version 3.68, et strictement interdite dans les versions précédentes.

La gestion devient alors assez complexe. Prenons le cas du cycle suivant:

PROC Cycle1()

```
Chge_outil nom_outil{1};  
AppelleTraj 1,0;  
Chge_outil nom_outil{2};  
Appelletraj 2,0;
```

ENDPROC

Cet exemple simple permet d'analyser les différents cas de figure possibles:

#### *1 – Abandon AVANT la prise de l'outil 1:*

Lors de la reprise, le robot va au repli, puis exécute *T\_Lancement*, puis retourne chercher l'outil 1, puis poursuit le cycle normalement. Ceci parce que les trajets de prise d'outil démarrent du PEO, et non du repli.

## *2 – Abandon APRES la prise de l'outil 1:*

Lors de la reprise, le robot va au repli, puis exécute la **traj de reprise** de la traj 1 (puisque l'outil 1 est déjà présent), puis reprends la traj 1 et le cycle complet. On voit que le fonctionnement n'est pas du tout le même que dans le cas précédent.

En fait, c'est "comme si" on avait abandonné la traj 1, et non la traj de prise de l'outil.

## *3 – Abandon PENDANT l'exécution de la traj 1:*

Si l'outil présent est toujours le même, fonctionnement classique (Repli, T\_Reprise). Si non, voir plus loin.

## *4 – Abandon PENDANT la traj 1 mais APRES l'instruction Fin\_de\_traj:*

La traj 1 n'est donc PAS reprise: le robot va au repli, puis exécute T\_Lancement, puis dépose l'outil 1, prends le 2, et poursuit le cycle avec la traj 2.

## *5 – Abandon AVANT la prise de l'outil 2 (dans la traj de prise):*

Idem que le cas 1: on exécute T\_Lancement, puis on va chercher l'outil 2 (après avoir éventuellement déposé l'outil présent incorrect), puis on poursuit le cycle normal, **donc sans exécuter** la traj de reprise de la traj 2.

## *6 – Abandon APRES la prise de l'outil 2:*

Idem au cas 2.

Après un spécial 3 suivi d'un lancement, l'outil physiquement présent est toujours testé et comparé à ce qu'il devrait être compte tenu des circonstances dans lesquelles le Spécial 3 a été fait.

Si l'outil a disparu (dépose "sauvage" en manuel), ou est différent (suite à passage en apprentissage ou tests divers), le Standard va automatiquement déposer l'outil en cours si ce n'est pas le bon, puis prendre celui demandé dans la routine Chge\_Outil, et ceci quelque soit le moment où l'abandon programme a été fait.

Le fait d'exécuter la traj T\_Lancement ou la traj de reprise de la traj abandonnée dépend donc uniquement de la cohérence entre l'outil présent et celui attendu.

## VII - FONCTIONNEMENT EN PRODUCTION

Le passage "En Production" peut s'effectuer de deux façons:

1 - En mode Manuel, en validant le choix "LANCEMENT PRODUCTION" du menu principal.

2 - En mode Automatique, de la même façon **ou** en appuyant sur le bouton "DEPART CYCLE". Dans ce cas, le départ n'est possible que si:

- Le programme était dans le menu principal, OU
- On vient de le relancer au début (Spécial 3).

Par conséquent, une action sur le bouton "DEPART CYCLE" n'a aucun effet si le programme est dans le sous menu "Services", par exemple.

Le Départ à Distance est de plus interdit dans certains cas, notamment lors de l'exécution d'une trajectoire de présentation Colle ou de Changement Electrodes. Dans ce cas, seul un Départ Local est autorisé.

Le programme commence par vérifier si le robot est au Repli. S'il n'y est pas, un menu s'affiche, proposant de rejoindre le repli par le chemin le plus court. Si le robot est vraiment trop éloigné du repli, repassez en Manuel et amenez le au joystick dans une position d'où il pourra rejoindre le repli en direct. Relancez alors le programme et validez "OUI": le robot rejoint le repli à vitesse réduite.

Il active ensuite son outil, sauf si un changeur est déclaré et qu'aucun outil n'est présent. Dans ce cas, un simple message "Pas d'Outil Détecté" s'affiche.

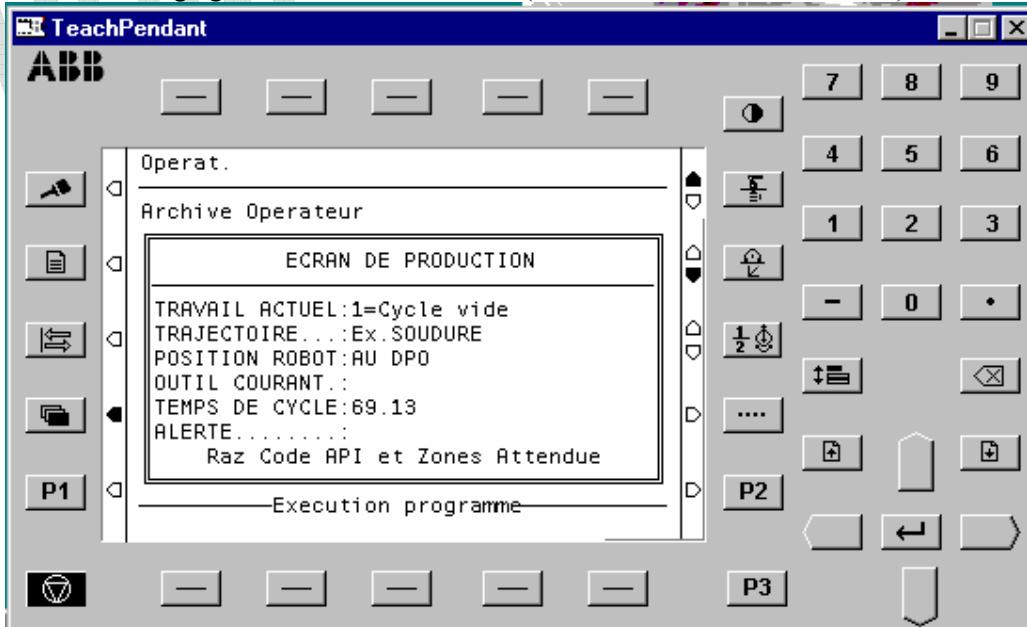
Le programme vérifie ensuite les surveillances de tous les process installés. Si une surveillance est inhibée, il faut confirmer:



Le robot exécute ensuite la trajectoire de Lancement, ce qui le place sur la position PEO, en attente de Code Programme.

Pour un robot encolleur, les temps de déclenchement de purge automatique sont lancées. L'écran dit "de production" s'active. Il indique:

- Le nom et le numéro du cycle en cours ou du dernier effectué.
- Le nom de la trajectoire en cours ou de la dernière effectuée.
- La position du robot (AU DPO).
- Le dernier temps de cycle mesuré.
- Une alerte éventuelle (Demande de purge...).
- Un message général sur l'état du robot (Attente Demande de Travail...).



Cet écran est ré-actualisé automatiquement dès que l'un de ses constituants a changé. Il n'est donc pas rafraîchi de façon cyclique et continue, ce qui serait gourmant en temps CPU.

Dans certains cas, cet écran est rafraîchi trop souvent et l'accès aux autres fenêtres du système devient difficile, par exemple les entrées sorties.

Dans ce cas, appuyez sur P2: le rafraîchissement est alors bloqué pendant 60 secondes, ou jusqu'à un nouvel appui sur P2.

Il est également ré-actualisé si on change de mode: c'est le cas notamment lorsqu'on passe en Manuel: il faut bien arrêter cet écran pour récupérer la main, mais il faut le ré-afficher si l'on repasse en Auto, alors que son contenu n'a pas changé.

Par ailleurs, un appui sur P1 permet de visualiser:

- Le numéro de Version/Révision du standard.
- L'état de la supervision de mouvement.
- Les éventuels contrôles inhibés.
- Les alertes qui font clignoter la verrine rouge (alarme groupe encollage, alerte Décrassage torche...)

Certaines données de cet écran sont modifiables par le programmeur:

#### *1 - La position du robot:*

Le programmeur peut écrire

AffPosition " Point de contrôle";  
quelque part dans sa trajectoire, pour que ce message s'affiche dans l'écran de production,  
dans le champ « Position Robot ».

NOTA: Pour tous les points "Process", ex: pt de soudure, le Standard affiche automatiquement le NOM du point s'il est explicite (ex Ps123), ou simplement une indication "Point de soudure", si celui ci est en point \*. Cette fonction n'est disponible que si l'option "Fonctions Developpeur" a été installée. Sinon, le standard le détecte et affiche un message simple "Point Process".

#### *2 - Un message d'alerte:*

AffAlerte "Bac presque vide";  
Le texte apparaitra dans le champ "Alerte". Celui ci est limité à 23 caractères.

#### *3 - Un message d'information:*

AffMessage "ATTENTE DE GOUJON";  
Le texte apparaitra dans le champ "Message" de l'écran de production. Il est limité à 38 caractères.

Attention: ces textes ne sont pas mémorisés: ils peuvent donc être "écrasés" par un autre message venant du standard, par exemple "Demande de purge Debit".

Cet écran reste actif tant que le robot est "En Production", c'est à dire apte à recevoir un code programme ou en train de l'exécuter.

Le passage "Hors Production" s'effectue d'une seule manière: en appuyant à n'importe quel moment sur le bouton "REPLI". Le voyant se met alors à clignoter, pour indiquer que la demande a été prise en compte. A la fin du cycle en cours, le robot attend que le code programme venant de l'automate passe à 0, puis rejoint le Repli par la trajectoire de repli.

Le menu principal est alors ré-activé.

Si on ne bouge pas le robot, un appui sur "Depart Cycle" le relance immédiatement en production. Les asservissements ne sont pas coupés à la fin de la trajectoire de repli. Le voyant Repli s'allume en fixe, et la sortie do\_SA\_Robot passe à 1. Si l'on déplace légèrement le robot au joystick, le voyant Repli s'éteint et la sortie do\_SA\_Robot retombe.

A la fin de l'exécution de chaque trajectoire, le "Mode Process" est automatiquement vérifié: si par exemple on vient de faire une trajectoire d'encollage et que l'on est passé au moins une fois en mode "SANS COLLE" avant le début d'un cordon, le standard affiche:

"Pièce mal encollée, Continuer ou Refaire ?"

Si on vient de faire une soudure, le message s'adapte:

"Pièce mal soudée, Continuer ou Refaire ?"

Si l'opérateur valide "Continuer", le cycle se poursuit. Si par contre il choisit "REFAIRE", le programme est automatiquement ré-initialisé. C'est alors le mécanisme des trajectoires de Reprise qui prendra la main lors du re-lancement du programme.

Cette gestion est transparente pour le programmeur. Elle s'adapte automatiquement au type de process qui vient d'être effectué. Il est toutefois interdit, pour que cela fonctionne correctement, de mélanger les process à l'intérieur d'une même trajectoire. On créera donc une trajectoire par process. (Exemple: soudure et encollage dans deux trajectoires différentes).

Une exception toutefois: il est possible de mélanger Colle Debit et Mastic, bien que ces applications soient différentes. En effet, le standard considère ces deux process de la même façon: c'est de l'encollage au sens large.

## **1 - Surveillance Communication Automate :**

Depuis la version 3.0 du standard, le réseau Profibus ou FipIO est surveillé par le robot dès qu'il est en mode Production.

Dès que cette entrée robot retombe, cela signifie une perte de communication sur le réseau, ce qui a pour effet d'arrêter immédiatement le robot, ainsi que le process en cours (encollage, soudage arc...), d'afficher un défaut bloquant, puis de réinitialiser complètement le programme.

Toute perte de dialogue se traduit donc par un redémarrage du programme au début et un retour au repli.

Si l'on ne souhaite pas activer cette fonction, il suffit de déclarer l'entrée *di\_ComOk* inversée.

Depuis la version 3.68, la réaction du programme sur perte communication automate est paramétrable au moyen du boolean ***Repli\_Sur\_Perte***:

- Si TRUE: une perte de communication provoque un Reset du programme, il faut appuyer 2 fois sur Depart pour repartir, le robot retourne au repli.
- Si FALSE: le robot repart de là où il est, après recalage sur traj, si la communication est rétablie lors de l'appui sur Depart.

Attention: toute modification de ce booléen nécessite un P-Start pour être prise en compte.

## **2 – Procédure d’Arret Cycle**

Depuis la version 3.7, une fonction Arrêt Cycle a été implémentée. Si l'entrée *di\_Dmd\_Arret* venant de l'automate passe à 1, le robot se met en attente et monte sa sortie *do\_Rob\_Arrete* en affichant le message « Arret Cycle Demandé ».

Les asservissements ne sont pas coupés. Le robot peut donc repartir spontanément si l'entrée *di\_Dmd\_Arret* retombe.

Mais généralement, cette fonction est utilisée pour faire un accès ilôt, ce qui provoque la perte des asservissements.

La mise en attente du robot se fait uniquement à des moments particuliers, à savoir :

- En attente de code cycle.
- Avant toute attente d'une autorisation de zone.
- Pendant toute attente d'une autorisation de zone.
- Avant toute trajectoire.

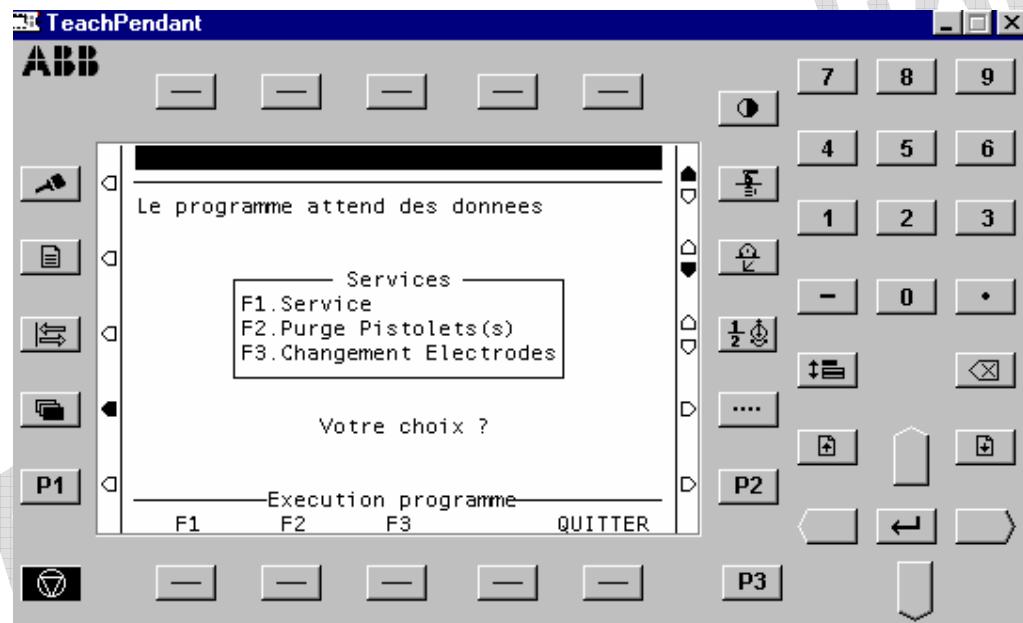
Elle ne peut pas se faire en cours de process ou de trajectoire.

## VIII - DEMANDES DE SERVICE

En cours de production, il est possible à tout moment de faire une demande de service en appuyant sur le BP "Demande de Service". Ce bouton n'est pas lumineux, la seule façon de s'assurer que la demande a bien été prise en compte est de regarder l'écran pour y lire le message

"SERVICE DEMANDE"

A la fin du cycle en cours, le robot propose un menu dépendant du type d'application installée, et notamment si un changeur d'outils est présent (voir chapitre "Changeur d'Outils").



Ce menu permet d'exécuter le service demandé. A la base, seule la trajectoire de service est commune à toutes les applications. Elle rejoint la position dite "de service", puis les asservissements sont coupés. Un appui sur le BP "DEPART CYCLE" est alors nécessaire. Puis le robot rejoint le PEO.

Les autres choix sont variables. Si une application Encollage est installée, on trouvera:

Fx : Purge Pistolet(s)

Si une application Soudure est installée, on trouvera:

Fx : Changement Electrodes

Si ces deux applications coexistent, les deux choix sont disponibles

Dès validation d'un choix du menu, la trajectoire correspondante est exécutée. Le robot peut rester en Auto.

Après validation du choix "QUITTER", l'écran de Production est réactivé, le robot est au PEO, en attente d'un nouveau code programme.

Un code programme PEUT être reçu et accepté **pendant** l'exécution d'une traj de service.

NOTA : Pour certains sites, une trajectoire de nettoyage du robot peut être utilisée. Son exécution est possible à partir de ce menu. Toutefois, ce cas étant rare, cette trajectoire, dénommée « T\_Nettoie », n'est pas accessible en mode Apprentissage. C'est l'exception qui confirme la règle : elle ne peut être modifiée que par un appel à « Autre Routine ».

Pour information

## IX - DETAIL DES DIFFERENTS PROCESS

### **1 - Fonctions communes à tous les process:**

Tous les process ont un mode de fonctionnement similaire. Si une perte du présence outil intervient (quelque soit l'outil), un défaut bloquant est signalé si le robot est sous asservissements et que le programme est en exécution. Ceci provoque un arrêt du programme. Ce défaut réapparaît ensuite dès que l'on tente de relancer le programme en exécution, et sous réserve que le contrôle soit toujours absent.

Par contre, il est possible, même en l'absence du contrôle, de déplacer le robot au joystick pour le dégager. Dans ce cas, le défaut n'est donc signalé qu'une fois.

Si l'on est gêné par cette surveillance parce qu'on est en phase de mise en route et que certains contrôles manquent, on peut désactiver les surveillances d'un process par le menu "Réglages", "Désactiver surveillances". Il est nécessaire de passer en mode Simulation pour accéder à ce menu, sinon le défaut apparaît en permanence.

#### *a- Gestion du mode Avec/Sans Piece et des clés Avec/Sans Process:*

Si l'on a mis le flag SANS\_PIECE à true, pour faire des essais, aucun process n'est exécuté, quelquesoitoit l'état de la clé (sauf soudure SR). Le clinchage est simulé, la colle ne se dépose pas, etc...

En soudure par points, la soudure est réalisée selon l'état de la clé Avec / Sans Soudure. On peut donc souder même sans pièce. Si l'on est Sans Soudure, les mouvements pinces sont exécutés, mais la puissance soudure est inhibée. Les compteurs de points pour le rodage et le changement électrodes sont gérés, ainsi que les trajectoires associées.

Les mouvements du préhenseur sont exécutés. Si un présence pièce est vu à 1, le mode AVEC PIECE est remis, après confirmation de l'opérateur.

Si l'on est AVEC PIECE, le mode process est forcé sur AVEC à chaque début de cycle, quelque soit l'état de la clé. On démarre donc forcément Avec Colle, Avec Clinchage, Avec Soudure.

Ensuite, lors de chaque **instruction** process (DebitL, PistoL, Soude...), l'état de la clé est vérifié. Si elle est sur Sans Process, le point est effectué sans soudure, le cordon de colle est interrompu. Une mémoire monte, indiquant que la pièce n'est pas correctement réalisée.

A la fin de la trajectoire, un menu apparait, permettant de refaire ou d'évacuer la piece.

#### *b - Gestion du redémarrage après arrêt:*

Lors d'un arrêt robot en cours de process (perte autorisation de zone, AU, stop...), celui-ci est arrêté s'il est de type continu: Debit, colle, soudure arc.

A ce moment, si l'on passe en Manuel et que l'on fait Instruction Avant ou Instruction Arrière, le standard considère que la trajectoire a été modifiée. Un message apparaîtra donc au prochain redémarrage en production, proposant de sauvegarder les éventuelles modifications.

Lors de la relance du robot en **exécution continue**, deux cas sont possibles:

1 - Reprise de l'exécution en mode Manuel:

- Les process continués sont inhibés (sauf en soudure arc). Une mémoire est Montée pour affichage du défaut en fin de trajectoire.

2 - Reprise en mode Automatique:

- Si la clé est passée SANS PROCESS avant la reprise, la trajectoire se termine ainsi. La colle n'est pas réactivée, ni le cordon de soudure.

- Si la clé est restée en position AVEC PROCESS, et qu'un process était en cours lors de l'arrêt, (stop au milieu d'un cordon...), un menu apparaît, permettant de poursuivre le cycle Avec ou Sans Process. Si l'on choisit Avec, le process est réactivé immédiatement après le recalage sur trajectoire (le pistolet s'ouvre), puis le robot part.  
On a une chance que le raccord soit acceptable.

Si l'on choisit SANS, la trajectoire se termine **sans process**.

*NB: Le process Encollage Doseur n'accepte pas de reprise de cordon.*

- Si aucun process n'était en cours (stop entre deux cordons), le robot redémarre normalement, l'état de la clé sera pris en compte à la prochaine instruction de process de même nature.

Dans tous les cas, (sauf arrêt Hors Process), le standard considère que la pièce est peut être mauvaise, et l'on aura le fameux message "Continuer, Refaire" à la fin de la trajectoire.

## **2 – Encollage Debit (contrôle par Volucompteur)**

L'application Encollage Debit permet de gérer 2 pistolets différents **non simultanés**. Le pistolet 1 doit être refermé avant (ou en même temps) d'ouvrir le pistolet 2. Les deux ne peuvent être ouverts en même temps.

Deux instructions principales permettent de gérer le process Encollage Debit:

- ColleDebitL.
- ColleDebitC.

Deux autres permettent des fonctions "annexes":

- Purge\_Debit

- Presente\_Debit.

Ces 4 fonctions sont directement disponibles sous le menu "LstSys2", "Commune 2".

La première permet d'encoller selon une ligne droite, la deuxième selon un arc de cercle. Elles sont proches dans leur utilisation des routines de mouvement normales "MoveL" et "MoveC".

Toutefois, il est nécessaire de leur transmettre des paramètres supplémentaires, propres à l'encollage Debit.

Le système d'encollage est doté d'un volucompteur, qui mesure la quantité de produit réellement déposé. Ce volucompteur a besoin d'être initialisé à l'aide d'une sortie robot "Debut Comptage". Environ 200 ms après avoir activé cette sortie, le robot doit vérifier que le compteur est en marche, grâce à l'entrée "Comptage en cours". Un défaut est généré si ce n'est pas le cas. Ce contrôle ne doit toutefois être bloquant que pour le **premier cordon** d'une série, c.a.d. d'une trajectoire.

A la fin du **dernier cordon**, il est nécessaire d'arrêter le volucompteur, et de vérifier si la mesure est correcte, donc si le volume de colle déposé corresponds à celui prévu. Il faut également s'assurer que le comptage est arrêté.

Le volume attendu est codé à l'aide d'un numéro de programme Encollage, sur 4 bits.

On voit donc:

- 1 - Qu'il est nécessaire de traiter différemment le premier cordon et les suivants.
- 2 - Que le dernier cordon constitue également un cas spécial.
- 3 - Que tous les tests se font "à la volée", donc sans arrêt robot, sauf sur défaut.
- 4 - Qu'un numéro de code programme doit être transmis au groupe d'encollage.

Il faut noter que le comptage ne s'arrête pas entre deux cordons. Seul le pistolet se ferme.

Les deux routines DebitL et DebitC possèdent donc, en sus des paramètres standard (vitesse, zone, outil, objet), les paramètres obligatoires suivants:

- 1 - Numéro de Cordon.
- 2 - Numéro de Programme.
- 3 - Anticipation.

Puis les paramètres switch optionnels suivants:

- 4 - P1On / P1Off **ou** P2On / P2Off.
- 5 - PremierCordon / DernierCordon.

Le premier paramètre (1) permet simplement d'identifier les cordons: ce numéro apparaît à l'écran de production.

Le deuxième (2) permet de sélectionner un numéro de programme à envoyer au groupe d'encollage, et donc d'attendre un certain volume de colle déposé.

Le troisième (3) permet de modifier l'endroit exact où l'ouverture / fermeture pistolet va se produire par rapport au point appris. Avec une valeur nulle, cette action est exécutée exactement au centre de la sphère définie dans le paramètre de zone, donc sans retard ni anticipation. Une valeur positive ou négative permet de déplacer le point d'ouverture / fermeture du pistolet le long de la trajectoire, exprimée en millimètres.

Si une anticipation en distance ne donne pas le résultat escompté (notamment en cas de cordons rapides très proches les uns des autres), le paramètre optionnel \Temps permet de modifier le comportement du paramètre Anticip: dans ce cas, la valeur d'anticipation n'est plus exprimée en millimètres, mais en secondes (avance ou retard).

Le quatrième paramètre, optionnel, permet de déclarer si l'on souhaite ouvrir ou fermer le pistolet. Ce paramètre sera donc utilisé à chaque début de cordon et à chaque fin de cordon. Pour les points intermédiaires d'un cordon, il n'est pas utile.

Enfin, le dernier permet, comme son nom l'indique, de spécifier s'il s'agit du premier cordon ou du dernier. En effet, lors du premier cordon, une demande de comptage doit être envoyée, puis un test de l'entrée comptage doit être fait. Pour le dernier cordon, une demande d'arrêt comptage doit être faite, puis un test des différentes informations possibles: dose trop faible, trop forte... Ces mesures sont effectuées sur l'ensemble des cordons, entre le premier et le dernier.

Sur un cordon intermédiaire, seule la présence de l'entrée "Comptage en cours" est vérifiée.

Ce paramètre est intimement lié à PxOn / PxOff:

"PremierCordon" ne doit être utilisé qu'en présence de PxOn, et "DernierCordon" ne le doit qu'en présence de PxOff.

Le programme vérifie d'ailleurs cette cohérence.

Ceci permet, entre autres, de ne créer qu'un seul cordon dans une trajectoire, qui est donc à la fois le premier et...le dernier !

Le numéro de programme est envoyé au groupe d'encollage avant le début du cordon. Or, il n'est pas garanti que le groupe d'encollage sache gérer des programmes successifs différents au sein d'un même groupe de cordons, c'est à dire tant que le comptage est actif.

Il est donc recommandé de ne pas changer de numéro de programme d'un cordon au suivant, sauf si on a utilisé PremierCordon / DernierCordon, qui gèrent le comptage.

Avant le début du premier cordon, le programme vérifie les défauts groupe encollage, et l'entrée comptage qui doit être à 0.

Il n'est pas prévu, dans la version actuelle du logiciel, de piloter et de contrôler les 2 pistolets **simultanément**. On prendra donc garde à s'assurer qu'un pistolet est fermé avant de demander l'ouverture de l'autre.

### *Purge à tout moment*

Dans certains cas, on peut être amené à effectuer une purge au milieu d'un cycle. Par exemple, sur un poste de mariage, il faut encoller juste avant le mariage. Si l'autorisation de zone arrive après un temps trop long, il faut pouvoir purger.

Le standard gère également ce cas. Il suffit d'insérer dans la trajectoire, où dans le cycle, à l'endroit souhaité, l'appel de la fonction

Ep\_A\_Purger()

qui renvoie VRAI si le temps écoulé depuis la dernière **fermeture pistolet** est supérieur à la valeur paramétrée dans la variable "ec\_TPS\_AVT\_PURGE".

On pourra donc écrire, par exemple:

```
IF Ep_A_Purger() THEN
    MoveJ...
    ..MoveJ...
    ...T_Purge;
    ...MoveJ...
ENDIF
```

Dans cet exemple, la trajectoire de purge "T\_Purge" sera appelée si nécessaire, après avoir effectué les points d'approche éventuels de la trajectoire de purge.

Si cette fonction est utilisée dans un cycle, pas de problème: le cas des reprises de trajectoire est prévu. La fonction renvoyant toujours FAUX tant qu'une reprise est en cours, ni les points ni la purge ne seront effectués.

Par contre, si la fonction est utilisée dans une trajectoire, la purge sera effectuée dans tous les cas.

### Pourquoi une fonction plutôt qu'une routine gérant directement l'appel à la purge ?

Pour pouvoir insérer d'éventuels points intermédiaires entre l'endroit du test et le début de la trajectoire de purge.

Il n'est d'ailleurs pas obligatoire d'appeler la **trajectoire** de purge: on peut se contenter d'appeler la procédure de purge seule, qui ne contient pas de mouvement robot. Exemple:

```
IF Ep_A_Purger() Purge_Debit 2;
```

provoque la purge du pistolet pendant 2 secondes, **sans déplacer** le robot.

Le trajectoiriste pourra choisir l'une ou l'autre méthode, selon la configuration particulière du site. (Pistolet fixe ou embarqué).

Le standard permet ces deux modes.

Toutefois, d'un point de vue logique, cette fonction devrait être appelée au début d'une trajectoire d'encollage, donc après réception de l'autorisation d'accès zone de la trajectoire en question, et devrait également appeler directement la procédure de purge. (Exemple ci-dessus).

En résumé, la partie visible de l'application Debit présente les routines suivantes:

- |                 |   |
|-----------------|---|
| - DebitL...     | Réalise un cordon linéaire.                                 |
| - DebitC...     | Réalise un cordon circulaire.                               |
| - Ep_A_Purger() | Indique qu'il faut purger le pistolet.                      |
| - Purge_Debit X | Purge le pistolet pendant X secondes, sans mouvement robot. |

Les données modifiables sont:

- |                           |  |
|---------------------------|--|
| - <i>ep_TPS_AVT_PURGE</i> | : Donne le temps au bout duquel une demande de purge est générée. Le temps est celui depuis la dernière fermeture du pistolet. |
| - <i>ep_TEMPO_DEF_GRP</i> | : Donne le temps au bout duquel il faut ré-afficher le défaut groupe encollage.  |

### **3 - Encollage Billes de Verre et Pulvérisation (contrôle par Doseur)**

Ce process ressemble à l'encollage Debit. On lira donc utilement le chapitre précédent. Toutefois, une différence importante: le n° de programme est envoyé avant le début du premier cordon, au moyen de la routine *Code\_Piece X*. Ceci permet de remplir le doseur avec la quantité de colle qui sera nécessaire sur l'ensemble de la trajectoire.

Il n'y a pas de comptage: l'antenne d'encollage envoie simplement une information Defaut Bloquant ou non bloquant à la fin de la trajectoire (c'est à dire quand le robot remet à 0 sa sortie *do\_Dose\_Encours*, qui autorise l'ouverture pistolet).

Les instructions ColleDoseL et ColleDoseC permettent de gérer le pistolet. Elles sont directement disponibles dans la liste commune 2.

Le numéro de programme correspondant à la quantité de colle **totale** à déposer sur la pièce est un paramètre obligatoire de la routine *Code\_Piece*. Il doit également être spécifié à la routine *ColleDose*, mais dans ce cas il ne sert qu'à identifier le numéro de cordon élémentaire en cours (en cas de défaut notamment)..

Depuis le projet A8, le débit peut être modifié dynamiquement en cours de trajectoire avec le paramètre \Code\_Debit.

Lors de l'exécution du dernier cordon, un traitement particulier est opéré (raz de la sortie *do\_Dose\_Encours*, attente et contrôle de l'antenne d'encollage). Il est donc nécessaire de le spécifier à la routine ColleDose au moyen du paramètre optionnel \DernierCordon.

Une trajectoire d'encollage se présente donc ainsi:

```
PROC Traj1()
Nomtraj:="Encollage Doseur";
...
Code_Piece 1;      → Remplissage du doseur avec la quantité nécessaire à la pièce 1.
MoveJ...
MoveJ...
MoveJ...
ColleDoseL\P1On,*,1,speed,zone....          → Démarrage cordon n°1
MoveL (ou Colle DoseL...)
ColleDoseL\Code_debit := 4,*,speed,zone;
ColleDoseL\P1Off,*,1,speed,zone...           → Changement de débit
                                                → Fin cordon n°1
MoveJ
MoveL
ColleDoseL\P1On,*,2,speed,zone...           → Démarrage cordon n°2
ColleDoseL\P1Off,*,2\DernierCordon,speed,zone...   → Fin dernier cordon de la
                                                       Trajectoire.
Fin_de_traj *,...
MoveJ...
MoveJ...
ENDPROC
```

La position précise de l'ouverture et de la fermeture du pistolet (et le changement de code débit dans le cas A8) peut être ajustée au moyen du paramètre optionnel \Anticip. Ceci permet de tenir compte du temps de réponse de l'antenne d'encollage et du pistolet.

Par défaut, cette valeur est exprimée en millimètres d'anticipation par rapport au point appris.

Attention: si cette valeur devient supérieure à la distance qui sépare le point appris du point précédent, l'anticipation en distance **ne fonctionne plus**.

Pour cette raison, le paramètre optionnel supplémentaire \Temps permet de réaliser une anticipation en **secondes**, et non plus en millimètres.

Toutefois, l'anticipation en temps est plus difficile à maîtriser, car elle dépend de la vitesse, et semble moins performante en terme de répétabilité.

L'expérience a montré que l'anticipation en distance est préférable, tant que le point qui précède chaque ouverture ou fermeture de pistolet peut être créé suffisamment loin de celui ci. (à une distance au moins égale à la valeur donnée à *Anticip*).

A l'inverse des autres procédés d'encollage, toute interruption de l'encollage en cours d'exécution entraîne la perte de la pièce: le procédé ne permet pas de tenter de faire un raccord. La pièce se termine donc obligatoirement sans colle avec le message d'erreur classique en fin de trajectoire: "Piece mal encollée, Continuer ou Refaire ?". (Reprise de la traj entière).

Comme pour les autres process d'encollage, la colle n'est réellement déposée que si l'on est en AUTO et En Production (ou en Apprentissage, Auto et mode Test Colle), et Avec Colle. Tout passage en mode Sans Colle en cours d'encollage se traduit par l'arrêt de la colle dès le cordon suivant.

En cas d'arrêt du programme (Stop, AU, Perte autorisation de zone...), le pistolet est automatiquement coupé, le redémarrage se faisant toujours sans colle.

La fonction permettant de tester si une purge en cours de trajectoire est nécessaire s'appelle `do_A_Purger()` et fonctionne comme pour les autres process Encollage.

De même pour la présentation, qui s'appelle `Presente_Doseur`.

*Données modifiables (dans Configur.sys):*

<code>do_TPS_AVT_PURGE:</code>	Temps avant la purge.
<code>do_Surv_Pisto:</code>	Activation surveillance Présence Pistolet.
<code>mode_pulve:</code>	Activation du mode Pulvérisation.

La seule différence en mode Pulvérisation est que dans ce cas, c'est la sortie `do_dose_encours` qui monte lors du gavage, et non pas la sortie `do_Gav_Dose`.

#### **Evolutions à partir du projet A 8 (version 3.6+ du standard):**

- 2 pistolets Doseur peuvent être pilotés simultanément, ou indépendamment.
- L'instruction `Gavage` est remplacée par l'instruction `Code_Piece`.
- Il n'est plus nécessaire d'appeler la routine `Code_Piece` longtemps avant le premier cordon.
- Il est possible de modifier le débit de colle en cours de cordon avec le paramètre `\Colle_Dose`. Le changement de débit peut être anticipé ou retardé par rapport au point appris à l'aide du paramètre `Anticip`.
- Le paramètre `\PremierCordon` n'est plus utilisé.
- La purge n'est plus gérée par le robot, mais par l'antenne d'encollage, qui envoie l'info `di_Purge_Dem`, le robot envoyant quant à lui la sortie `do_Acq_Purge` pour provoquer la purge réelle.

#### **4 – Encollage Mastic (contrôle par Pressions)**

L'application Encollage Mastic peut gérer 2 pistolets **non simultanés**: l'encollage se fait soit avec un pistolet, soit avec l'autre.

La déclaration du deuxième pistolet (le premier existe forcément) se fait soit au moment de l'installation du logiciel, soit en modifiant la variable booléenne `ec_Pisto2_Existe`. Pensez à faire une sauvegarde pour que la modification devienne permanente. (menu Sauvegardes).

Les instructions maîtresses de l'encollage Mastic sont:

- `PistoL` pour effectuer un cordon linéaire.

- PistoC pour effectuer un cordon circulaire.

Avant le premier cordon, un appel à la routine **CtlAvantPisto** permet de vérifier les entrées de pression Min et Max.

Si cet appel est omis, il est de toutes façons exécuté par le standard.

Après le dernier cordon, un appel à **DefautsPisto** permet d'afficher les éventuels défauts qui ont pu se produire au cours de la session d'encollage. Si l'appel à cette fonction est oublié par le trajectoiriste, il est fait automatiquement par le standard à la fin de la trajectoire.

Toutes ces routines ont besoin d'un paramètre: P1 ou P2 pour spécifier de quel pistolet il s'agit.

Comme pour tous les autres process, le standard propose automatiquement le choix **Continuer / Refaire** si un défaut a été détecté au cours de la session d'encollage. Il n'est donc pas nécessaire de rajouter un traitement.

Les surveillances des seuils de pression peuvent être inhibés de façon globale sur un robot en modifiant la valeur de la variable *ec\_CTLAVT\_Px* (contrôles avant ou pendant encollage), ou sur un cordon particulier en indiquant les paramètres optionnels SansCtlAvt (contrôle avant encollage), et SansCtl (contrôle pendant encollage) dans l'appel de la routine.

Les routines **PistoL** et **PistoC** permettent d'ouvrir / fermer le régulateur et le pistolet sur point de passage, en spécifiant éventuellement une distance entre le point appris et le point où l'action sera réellement exécutée.

De plus, des valeurs globales de température à l'ouverture / fermeture du régulateur par rapport au pistolet peuvent être définies dans les variables *ec\_T\_DEB\_Px* et *ec\_FIN\_Px*. Ces températures sont à 0 par défaut.

Si l'on utilise ces paramètres, le régulateur s'ouvre après le pistolet en début de cordon, et se ferme avant la fermeture pistolet.

Les paramètres des deux routines **PistoL** et **PistoC** sont:

- |   |   |
|---|---|
| <ul style="list-style-type: none"><li>- <b>P1On   P1Off</b></li><li>- <b>P2On   P2Off</b></li><li>- \NC</li><li>- \SansCtlAvant</li><li>- \SansCtl</li><li>- <b>ToPoint</b></li><li>- \Dist_Antici</li><li>- <b>Speed</b></li><li>- <b>Zone</b></li><li>- <b>Tool</b></li><li>- <b>Wobj</b></li></ul> | <ul style="list-style-type: none"><li>- Ouverture / fermeture du pistolet 1.</li><li>- Ouverture / fermeture du pistolet 2.</li><li>- Numéro du cordon.</li><li>- Inhibe les contrôles avant début encollage.</li><li>- Inhibe les contrôles pendant encollage.</li><li>- Point de trajectoire appris.</li><li>- Distance d'anticipation (en mm) pour l'ouverture pistolet.</li><li>- Vitesse du robot.</li><li>- Zone pour un point de passage.</li><li>- Nom de l'outil</li><li>- Repère objet.</li></ul> |
|---|---|

*Purge à tout moment :*

Se référer un peu plus haut au cas de l'Debit. Le fonctionnement est identique, mais la fonction s'appelle *ec\_A\_PurgerX()*, où X est le numéro de pistolet.

*Fonction Test:*

Cette fonction s'utilise via les Commandes Manuelles. Elle doit être affectée à l'une des touches programmables du pupitre (voir chapitre *Commandes Manuelles*).

Elle peut également être appelée depuis une trajectoire quelconque, par exemple la trajectoire de purge. Son nom est **Teste\_Mastic** (original !).

Elle permet de vérifier le bon fonctionnement des capteurs de pression min et max. Elle s'effectue en cinq phases:

- 1 – Attente que les défauts soient absents (Seuils min et max à 1).
- 2 - Ouverture du régulateur avec pistolet fermé. Attente du déclenchement Seuil Maxi pendant 5 secondes.
- 3 - Fermeture du régulateur.
- 4 - Ouverture du pistolet avec régulateur fermé. Attente du déclenchement Seuil Mini pendant 5 secondes.
- 5 - Fermeture du pistolet.

La fonction de test est appelée automatiquement lors de la première purge de chaque équipe de travail. Les heures de début d'équipe doivent être renseignées dans la variable string **Heure\_Equipes** sous la forme « hh :mm » .

*Constantes modifiables:*

NOM	TYPE	VALEUR PAR DEFAUT	ACTION
Ec_Tps_Avt_Purge	Num	600	Durée depuis la dernière fermeture du pistolet pour déclencher la demande de purge.
Heures_Equipes	String	""	Heures des changements d'équipe pour l'autotest.
Ec_T_Ctl_Px	Num	0.35	Temps au bout duquel on contrôle les seuils de pression après ouverture pistolet x.
Ec_T_Deb_Px	Num	0.2	Temps de retard de l'ouverture régulateur sur l'ouverture pistolet x.
Ec_T_Fin_Px	Num	0.2	Temps de retard de la fermeture régulateur sur la fermeture pistolet x.
Ec_CtlAv_Px	Num	1	Type de contrôle effectué avant ouverture du pistolet x: 0: pas de contrôle. 1: contrôle pression max

			2: contrôle pression min
Ec_Ctl_Px	Bool	true	Activation des contrôles pendant cordon pistolet x.

### **Fonctions communes aux applications Pression, Volucompteur et Doseur:**

#### Limitations:

Le standard PSA V 3.69 peut gérer les différentes configurations suivantes:

- Mastic 1 pistolet.
- Debit 1 pistolet.
- Mastic 2 pistolets (non simultanés).
- Debit 2 pistolets (non simultanés).
- Mastic 1 pistolet et Debit 1 pistolet.
- Doseur 1 pistolet.
- Doseur 1 pistolet et Debit 1 pistolet.
- Doseur 1 pistolet et Mastic 1 pistolet.
- Doseur 2 pistolets simultanés (depuis A 8).
- 

En conclusion, l'application peut gérer au maximum deux pistolets de n'importe quel type .

Si deux pistolets sont déclarés en Debit, il est supposé qu'il existe deux groupes d'encollage différents. Par conséquent, les surveillances Groupe Bloquant et Groupe non Bloquant sont doublées, de même pour le Code Cordon.

Les Entrées Sorties de mesure du volucompteur sont également doublées. Dans le cas où un seul groupe serait présent, ou un seul volucompteur partagé entre les deux pistolets, il suffit de déclarer les entrées sorties du pistolet 2 aux mêmes adresses que celles du pistolet 1.

#### *Defaut Groupe Non Bloquant:*

Si l'entrée *di\_Grp\_Alarme* ou *di\_Grp\_AlarmeEPO* tombe à 0, un message est affiché dans le champ Alerte de l'écran de production. Cet affichage revient périodiquement tant que le défaut est présent. Il réapparaît également à chaque début de cycle et à chaque fin de cycle.

Si c'est la première fois qu'il survient, le robot s'arrête en fin de cycle et il est nécessaire d'acquitter le message. Cela n'empêche toutefois pas de continuer à produire. C'est une alarme légèrement "renforcée".

Cette alarme peut correspondre par exemple à l'atteinte du niveau bas de la colle dans la réserve, indiquant qu'il faut prévoir de changer celle ci.

#### *Defaut Groupe Bloquant:*

Le perte de l'information *di\_Grp\_Bloq* ou *di\_BloqEPO* indique qu'un défaut majeur est intervenu sur le groupe d'encollage, éventuellement pendant l'encollage.

Dans ce cas, le robot s'arrête à la fin de la trajectoire d'encollage en proposant le menu classique "Piece mal encollée, Continuer ou Refaire ?". Un choix est nécessaire.

Toute nouvelle exécution d'un code programme est bloquée tant que le défaut persiste. Un dépannage du groupe est donc indispensable pour repartir.

#### *Présence Pistolet(s)*

Le capteur de présence pistolet est testé en permanence si le flag *ec\_SURV\_PISTO* est à true (ou *ep\_SURV\_PISTO* pour l'Debit). Ces valeurs sont à TRUE par défaut, mais peuvent être modifiées via le menu "Réglages", "Surveillances".

En cas de perte de ce contrôle, un message bloquant s'affiche, quelque soit le mode courant du robot. Ce message doit être acquitté. Tant que le défaut est présent, le message ré-apparaît dès que l'on tente de faire EXEC du programme. Par contre, il n'apparaît pas si l'on déplace le robot en manuel au joystick.

Ceci permet de se sortir de situations embarrassantes, et d'assurer la sécurité en interdisant toute exécution de programme. (en manuel ou en auto, en production ou pas). L'exécution du programme en mode SIMULATION reste toutefois autorisée.

NOTA: Toutes les fonctions de surveillance de Présence Outil fonctionnent sur le même modèle, quel que soit le process.

#### *Trajectoires de Presentation*

On peut souhaiter vérifier visuellement la qualité de l'encollage avant d'évacuer la pièce. Pour cela, la routine *Presente\_Mastic* ou *Presente\_Debit* ou *Presente\_Doseur* doit être incluse dans la trajectoire d'encollage, après la fin du dernier cordon.

La présentation de la pièce peut se faire soit automatiquement, soit sur demande expresse de l'opérateur. Si la donnée numérique *Nb\_Cycles\_Av\_Pres* est supérieure à 0, par exemple 10, la trajectoire de présentation sera appelée automatiquement tous les 10 cycles.

De plus, si la donnée numérique *Nb\_Cycles\_Av\_Cli* est supérieure à 0, par exemple 3, la verrine rouge se met à clignoter 3 cycles **avant** l'appel de la trajectoire de présentation (donc ici après 7 cycles). Naturellement cette donnée doit être inférieure à *Nb\_Cycles\_Av\_Pres*.

De plus, cette routine teste systématiquement l'état du commutateur *di\_Dmde\_Visu*. S'il est à 1, c'est qu'une demande de contrôle est faite par l'opérateur.

Sinon, elle appelle la trajectoire *T\_Pres\_Mastic* ou *T\_Pres\_Debit* ou *T\_Pres\_Doseur*, qui doivent amener le robot à une position telle que le contrôle visuel soit possible.

Le point de contrôle est matérialisé par l'appel à la routine « **Presente** », déjà écrite dans les trajectoires de présentation.

Sur l'appel de cette routine, le robot s'arrête **après** avoir indiqué les éventuels défauts survenus pendant l'encollage, allume la verrine rouge, puis coupe les asservissements et affiche un message.

Un départ distant est rendu **impossible** : seul un départ local est autorisé. Ceci pour éviter qu'un opérateur relance le robot à distance alors qu'un autre est en train de vérifier le cordon.

NOTA : Si vous faites un Special 3 à ce moment, le prochain redémarrage en Auto ne sera possible qu'en Local.

Lors de l'arrêt du robot, la sortie **do\_Robot\_Ok** retombe.

La création des trajectoires de présentation est accessible via le menu « Apprentissage », « Trajectoires de Service » ; La suite du menu varie si le robot est également soudeur, ou pas.

Pour traiter les différentes diversités, trois routines différentes existent (*Presente\_Mastic*, *Presente\_Debit* et *Presente\_Doseur*), qui permettent d'appeler 3 trajectoires différentes. On peut en effet avoir le cas où l'on ne veut faire qu'une seule présentation à la fin de tout le process d'encollage (mastic ET Debit), ou seulement à la fin du Mastic, ou seulement à la fin du Debit, ou encore à la fin de chaque process élémentaire. Dans cette dernière hypothèse, on écrira :

```
PROC TrajXX()
Nomtraj := »Encollage Debit » ;
...
...
DebitL...
Presente_Debit ;
ENDPROC

PROC TrajYY()
Nomtraj := »Encollage Mastic » ;
...
...
PistoL...
Presente_Mastic ;
ENDPROC
```

NOTA : Il n'est pas prévu de trajectoires de reprise pour les trajectoires de présentation. Tout abandon en cours d'une trajectoire de présentation se soldera donc par une proposition de reprise de la trajectoire d'encollage qui précède.

*Mode Test Colle:*

Dès qu'une application Colle est présente, tout accès au menu Apprentissage Trajectoires de Travail est précédé de la question:

Passage en mode TEST COLLE ?

Si l'on répond OUI, l'exécution des trajectoires en mode Automatique est autorisé, ce qui permet de tester les trajectoires d'encollage.

Dans ce mode, les autorisations d'accès zone sont **ignorées** et les hors zone ne montent pas.

Il s'agit donc d'un mode "risqué" à manier avec précaution.

Le flag Teste\_Colle peut être utilisé judicieusement dans les trajectoires pour effectuer un rebouclage entre le dernier et le premier point:

```
PROC Traj1()
...
...
Anti_reboucle;
DebitL...
MoveJ...           → fin de la trajectoire "normale".
```

```
If Teste_colle then
  Stop;
  MoveJ...           → points de retour en début de trajectoire.
Endif
ENDPROC
```

Il devient ainsi facile de vérifier la trajectoire à vitesse réelle, sans risque.

Ce flag est remis à False dès que l'on quitte le mode Apprentissage, ou si l'on fait un reset du programme.

#### **4 - Clinchage**

Le Clinchage est un mode d'assemblage sans soudure particulièrement adapté lorsqu'on se trouve en présence de matériaux différents ou non soudables par points (aluminium). L'opération consiste à déformer localement les pièces en exerçant une très forte pression au moyen d'un poinçon et d'une matrice.

Ce poinçon et cette matrice sont montés sur une pince pouvant ressembler à une pince à souder en J.

Les points étant déformés, ce procédé ne peut être utilisé pour des points d'aspect.

On ne doit pas, sous peine de risquer d'abîmer la matrice, clincher une deuxième fois sur un point déjà fait. C'est pourquoi, en manuel, une validation opérateur est nécessaire avant d'effectuer réellement tout point de clinchage.

*Depuis la version 3.68, les reprises de trajectoire de clinchage sont possibles, avec détection des points réellement clinchés précédemment. Dans ce cas, une confirmation de l'opérateur est nécessaire sur chaque point supposé non clinché. Les points réputés déjà clinchés sont passés sans clinchage.*

Une clé AVEC / SANS Clinchage est placée sur la façade de la baie. Cette clé permet d'inhiber la puissance hydraulique. En position SANS, la pince se ferme, mais avec une pression faible. L'amplificateur hydraulique est inhibé. Ce mode permet de vérifier la position du point appris, ou de faire des tests répétés sur la même pièce.

Depuis la version 3.3 du standard, les pinces sont toutes **instrumentées** (système CEP). Il est, de plus, possible de piloter 2 pinces sur le même robot. (l'une après l'autre, et non pas simultanées).

De plus, deux systèmes de mesure cohabitent: le "Mono-Point", qui ne comporte qu'un seul canal de mesure, et le "Double\_Point", qui en comporte deux fonctionnant en parallèle.

Dans le cas de l'utilisation de deux pinces, seul 1 système de mesure existe, qui est donc partagé. Pour cela, une commutation est effectuée sur les canaux de mesure au moyen de la sortie robot "do\_Pince2". Cette commutation est invisible pour l'utilisateur.

Toutefois, il faut savoir que toute commutation de la mesure exige un ré-étalonnage des signaux, opération totalement automatique, mais qui prend forcément un "certain temps". Si l'on doit utiliser 2 pinces, il est donc judicieux, pour optimiser le temps de cycle, de minimiser le nombre de commutations de pinces.

Cet étalonnage est fait automatiquement par le Standard avant tout **premier** clinchage réel si l'on vient de changer de pince, ou si l'on a fait un Special 3 (Reset du programme).

Depuis la version 3.4, deux canaux de mesure peuvent exister physiquement, ce qui supprime la contrainte de l'étalement à chaque changement de pince.

Le standard gère jusqu'à 2 pinces, chacune pouvant être de type MP ou DP. Cette déclaration se fait, comme à l'accoutumée, lors de la première installation du programme. Elle peut toutefois être modifiée après coup manuellement, en agissant sur les booléens suivants:

CONST bool Clin2_Existe	→ Y a t'il une deuxième pince ?
CONST bool Pince1_DP	→ La pince 1 est elle de type double point ?
CONST bool Pince2_DP	→ La pince 2 est elle de type double point ?

En cas de modification manuelle, il est nécessaire de faire une sauvegarde par le menu "Sauvegarder les modifs", puis un P-Start avec la disquette standard présente (ou le réseau Ethernet). Attention toutefois aux déclarations d'entrées sorties, qu'il peut être nécessaire de modifier, puisque le programme d'installation déclare systématiquement les E/S non utilisées en virtuelle 32.

D'un point de vue exploitation, le clinchage est exécuté en appelant la routine "**MoveClinche**" dans une trajectoire. Cette routine est disponible dans la Liste Commune 2 sous le nom abrégé "**Clinche**". Le déroulement est le suivant:

La routine ouvre la pince par sécurité et recule la matrice, puis vérifie les contrôles d'ouverture. Le robot est ensuite amené sur la position spécifiée, en mode articulaire ou linéaire (articulaire par défaut). Le point est exécuté en point d'arrêt fin.

Si l'on est en manuel et AVEC Clinchage, un message avec confirmation apparaît, permettant d'exécuter ou non la séquence de clinchage. Utile pour les mises au point en manuel, en évitant de clincher deux fois au même endroit.

Si l'on est en mode SANS Clinchage, la sortie *do\_inhib\_clin* est montée, pour inhiber la puissance hydraulique. La pince sera donc fermée, mais avec un effort faible.

Si l'on est en mode SANS Pièce, la routine se termine après une temporisation de 3 secondes. (La pince ne se ferme pas).

Si nécessaire, le cycle d'étalonnage des capteurs est effectué, puis le clinchage lui-même. Se reporter au descriptif PSA pour le détail des chronogrammes de clinchage et d'étalonnage.

Dans le cas d'une pince DP, les deux canaux de mesure sont activés simultanément. Dès que l'un des canaux a répondu, (Mesok ou MesNok ou Mes2ok ou Mes2Nok), la pince est ouverte. Une fois ouverte, les deux canaux sont testés, et doivent répondre tous **les deux** "Mesure Ok" pour que le clinchage soit considéré comme bon. Ceci permet éventuellement de gagner un peu de temps si les deux canaux n'ont pas le même temps de réponse.

Les paramètres de la routine **MoveClinche** sont:

- **\L**: paramètre optionnel pour effectuer un mouvement linéaire au lieu d'articulaire.
- **\Pince2** : pour indiquer que l'on utilise la pince n° 2.
- **ToPoint**: point à atteindre.
- **No\_Prog**: numéro de programme à envoyer au pilote de clinchage.
- **Speed**: paramètre de vitesse pour le mouvement.
- **Tool** : nom de l'outil courant.
- **Wobj** : repère objet .

La zone n'a pas à être spécifiée: le point est forcément un point d'arrêt.

#### *Actions manuelles:*

Le clinchage peut être effectué dans le cadre des actions manuelles. Les mêmes tests sont effectués que dans le clinchage "normal": position du commutateur, mode sans pièce, etc...

La seule différence avec la routine MoveClinche est qu'il n'y a pas de mouvement robot.

Il est nécessaire de préciser le numéro de programme et le numéro de pince utilisé (même si il n'y en a qu'une).

## **5 – Soudure SR Pneumatique**

Les principales routines accessibles du process soudage sont « **MoveP** » et « **Soude** ». La première permet de déplacer la pince en même temps que le robot, par exemple de grande à petite ouverture, la deuxième réalise une soudure.

Si la pince est déclarée avec basculement, celui-ci est vérifié avant la fermeture.

Les paramètres de la routine Soude sont :

- \L :	Permet d'effectuer un mouvement linéaire.
- \PO1 :	Indique d'ouvrir la pince 1 en PO après la soudure.
- \GO1 :	Indique d'ouvrir la pince 1 en GO après la soudure.
- \PO2 :	Indique d'ouvrir la pince 2 en PO après la soudure.
- \GO2 :	Indique d'ouvrir la pince 2 en GO après la soudure.
- <b>ToPoint</b> :	Point visé.
- \No_Ident1:	Numéro de prog soudure pince 1
- \No_Ident2:	Numéro de prog soudure pince 2
- <b>Speed</b> :	Vitesse de déplacement du robot.
- \SansDec1 :	Pas de contrôle DEC pince 1 sur ce point.
- \SansDec2 :	Pas de contrôle DEC pince 2 sur ce point.
- \Att_ouv :	Inhibe le contrôle PO pour le cas de la fausse pince.
- \Fausse_Pince	Outil courant.
- <b>Tool</b> :	Repère objet.

Selon le type d'application SR (1 pince pneumatique, 2 pinces pneumatiques, pince électrique), les paramètres optionnels peuvent varier, ou ne pas exister. Toutefois les principes sont identiques pour ces trois types d'appli.

La routine « *MoveP* » permet de déplacer le robot avec un mouvement de pince synchronisé, de la P.O. vers la G.O. et réciproquement. Le mode pas à pas ainsi que la marche arrière sont gérés.

Si le point est un point de passage, le mouvement de la pince est déclenché lorsque le robot passe par la projection de la trajectoire sur le point.

Cette routine n'existe évidemment pas dans le cas de la pince électrique.

#### *Contrôle Dec*

##### *a) Principe:*

Le contrôle DEC (décollement électrodes) est basé sur la détection du passage d'un faible courant entre les électrodes de soudage et la masse, indiquant que la pince est fermée. Si cette info est présente alors que l'on sollicite l'ouverture de la pince, c'est que les électrodes sont restées collées.

L'entrée robot di\_dec1\_Ok est à 0 si la pince est fermée sur la pièce, à 1 dans le cas contraire.

##### *b) Utilisation:*

Le contrôle DEC peut être déclaré physiquement présent ou absent lors de l'installation de l'application, en répondant à la question "CONTROLE DEC INSTALLE ?". (Depuis la version 3.2). Si l'on réponds NON, le contrôle n'est jamais utilisé.

Après installation, il est possible de modifier cet état en modifiant la déclaration de la constante DEC\_INSTAL dans le module CONFIGUR. Il est nécessaire de sauvegarder le module et de faire un P-Start pour que cette modification soit prise en compte.

Si le contrôle a été déclaré installé, il est possible de l'inhiber à tout moment, en modifiant la persistante sr\_Avec\_Dec1 (true ou false), qui est true par défaut. La prise en compte du nouvel état est immédiate, il n'y a pas de P-Start à faire.

Si le contrôle est inhibé, la verrine rouge clignote et le message "Contrôle DEC inhibé" apparaît à l'écran si l'on appuie sur P1 en mode Auto.

Il est également possible de l'inhiber pendant une soudure, par exemple si le point est trop proche de l'outillage et que la pince le touche pendant la soudure, ce qui provoque un défaut. Il suffit d'ajouter le paramètre optionnel Sans\_Dec à l'appel de la routine SOUDE. Dans ce cas, la verrine ne clignote pas.

#### *c) Fonctionnement lors d'une soudure:*

Pendant la soudure, on vérifie que le contrôle DEC est passé à 0. Si ce n'est pas le cas après avoir reçu le fin de cycle soudure, le défaut "DEC NON PASSE PAR ETAT BAS" apparaît, la verrine rouge s'allume et il est nécessaire de valider.

A la fin de la soudure, si le contrôle DEC est resté à 0, un message "Defaut Dec" apparaît, et l'on attend sr\_TPS\_DECOL\_G01 ou sr\_TPS\_DECOL\_P01 (modifiables) avant de tester le contrôle de petite ou de grande ouverture. On est dans le cas du défaut Dec non bloquant.

Si le contrôle d'ouverture est absent, on retente d'ouvrir la pince et on ré\_attends la tempo. Si le contrôle Dec est toujours à 0 et que les contrôles d'ouverture sont absents, on déclenche un défaut bloquant "DEFAULT DEC BLOQUANT" avec allumage de la verrine rouge et validation opérateur obligatoire. Le défaut est consigné dans le fichier de suivi.

Après validation de l'opérateur, un menu apparaît, permettant de choisir l'action à faire:

##### 1 - Retester le contrôle Dec

S'il devient Ok, le défaut est annulé, on peut repartir.

##### 2 – Ouvrir / Fermer:

Cette action tente de refermer, puis de ré\_ouvrir la pince pour la décoller. Le contrôle Dec est à nouveau vérifié.

##### 3 – Ressouder le point

Le cycle de soudage est repris (presque) entièrement, le point est ressoudé et les contrôles sont refaits. Il arrive en effet que le fait de ressouder permette de décoller les électrodes.

##### 4 – Sauter le point

C'est l'action ultime si aucune des autres n'a fonctionné. Il y a évidemment risque d'arrachement des électrodes, il est donc prudent de couper l'eau avant de tenter cette action. Toutefois, le robot devrait être arrêté assez rapidement par le contrôle de collision.

*d) Fonctionnement en mode sans pièce (Déverminage):*

Dans ce cas, le Dec est tout de même testé. Si il passe à 0 pendant la fermeture de la pince, c'est soit que la pince touche l'outillage, soit qu'une pièce est présente alors que l'on est en déverminage (cas où l'on a oublié de repasser en mode Avec Pièce).

Dans ce cas, le robot s'arrête et la question "Passage en mode AVEC PIECE ?" apparaît. Si l'on répond OUI, le mode déverminage est désactivé.

Pour information

*Rodage :*

Le déclenchement du rodage se fait par appel de la routine « *Rode\_Pince* », **qui doit être réécrite** dans les trajectoires de rodage, pour des problèmes de compatibilité entre les différents process soudure. Si la fraise est trop épaisse, le paramètre optionnel \Fraise\_Epaisse permet d'inhiber le contrôle de P.O. pendant le rodage.

Le standard positionne automatiquement le numéro de programme 31 à envoyer à la CPS (ou 127 si le code est sur 8 bits). Si un changeur d'outils est présent, le numéro envoyé dépend de l'outil courant : on a sr\_PROG\_RODAGE1 pour la pince 1, sr\_PROG\_RODAGE2 pour la pince 2, et sr\_PROG\_RODAGE3 pour la pince 3. Par défaut, ces constantes sont à 31, 30 et 29.

ATTENTION: dans le cas où 127 programmes de soudure ont été déclarés, ces constantes sont à 127, 126 et 125.

Si le rodage en temps masqué n'a pas été déclaré lors de la configuration ou si le flag sr\_ROD\_MSQ est à false, le rodage se fait par appel de la trajectoire de rodage.

Sinon, le rodage se fait en « temps masqué » (cas de la pince au sol et rodeuse au sol montée sur un abattant). Par défaut, le standard autorise le rodage masqué dès que le robot est au DPO. Si l'on souhaite anticiper le rodage dès que le robot est dégagé (par ex pendant la traj de dépose pièce), un appel à « *Hors\_Zone\_Rod* » est nécessaire.

Le rodage en temps masqué est totalement géré par le standard, aucune programmation n'est à faire. Toutefois, si la rodeuse est trop épaisse pour pouvoir roder à partir de la petite ouverture, il faut positionner le flag « **Frais1\_trop\_epai** » ou « **Frais2\_trop\_epai** » à TRUE dans le fichier Configur.sys. Dans ce cas, le graphe de rodage est un peu différent : la pince va toujours en grande ouverture avant d'avancer le bras rodeuse et de la lancer. Ensuite, le robot ferme la pince mais sans vérifier le contrôle de petite ouverture. Seule la perte du contrôle de grande ouverture est vérifiée.

Si le robot peut revenir dans la zone de la pince un peu plus loin dans le cycle, il faut s'assurer que l'éventuel rodage masqué est terminé et que la rodeuse est dégagée. La routine « *Dans\_Zone\_Rod* » sera utilisée pour cela.

**ATTENTION :**

Depuis la version 3.73, la sécurité d'interverouillage Robot / Abattant qui existait par défaut en standard a été **supprimée**. Cela permet au robot de repartir dans un nouveau cycle même si le rodage n'est pas terminé, d'où un gain en temps de cycle appréciable.

Mais il devient du ressort de l'intégrateur d'utiliser la routine *Dans\_Zone\_Rod* dans ses trajectoires, avant les points de collision éventuelle entre le robot et l'abattant.

Lors du lancement du robot en production ou suite à un reset du programme en cours de rodage masqué, une procédure de recyclage est lancée : celle-ci ouvre la pince au sol (sauf si elle est déjà ouverte), puis recule le bras rodeuse, et arrête la rodeuse. Si l'abandon du

programme a eu lieu avant la fin normale du rodage, celui-ci est considéré comme **non fait** : le compteur n'est pas remis à 0.

Le nombre de rodages élémentaires et le temps entre chaque rodage élémentaire peuvent être spécifiés dans les constantes num *sr\_ROD\_NBRE\_ELEM* et *sr\_ROD\_TPS\_INTER*.

Ces valeurs sont communes aux deux pinces dans le cas d'une pince embarquée et d'une pince au sol.

L'arrêt de la rodeuse est vérifié avant de commencer le rodage.

La rotation de la rodeuse est vérifiée pendant tout le cycle de rodage.

A la fin du rodage, le standard restitue la position qu'avait la pince avant le rodage : petite ou grande ouverture.

A la fin du rodage, l'arrêt de la rodeuse n'est plus vérifié (depuis la version 3.3) pour gagner en temps de cycle.

NOTA : il est inutile et interdit, comme cela s'est vu, de mettre des instructions de soudure dans une trajectoire de rodage.

Le rodage en temps masqué est possible même avec une pince électrique asservie, depuis la version 3.1 du standard.

#### *Rodage avec contrôle de vitesse (AMDP ou EXROD):*

Ce système consiste à mesurer la vitesse de rotation de la rodeuse avant, pendant et après rodage, à l'aide d'un codeur incrémental et d'un boîtier électronique spécial, ce qui permet de s'assurer que la pince est effectivement sur la rodeuse, et que la vitesse de rodage est correcte.

Cette gestion spéciale est totalement transparente pour l'utilisateur: il doit seulement déclarer le type de rodeuse au moment de l'installation de l'application. Cela peut également être modifié à tout moment en changeant la constante bool *RodX\_Amdp* de la rodeuse considérée (X). Si ce flag est à FALSE, le contrôle de la rotation rodeuse est classique (1= rodeuse en route), sinon, le fonctionnement est sensiblement différent:

- Rodeuse à l'arrêt, l'entrée robot *di\_Rod1\_Rot* doit être à 1.
- Lors du lancement de la rodeuse, cette entrée doit passer à 0. (avant moins de 5 secondes). Sinon, défaut rodeuse "Phase 2".
- Elle doit ensuite repasser à 1 avant 2 secondes, indiquant la fin de la période d'étalonnage du boîtier AMDP (mesure de la vitesse de rotation à vide). Sinon, défaut rodeuse "Etalonnage AMDP Phase 3".
- Elle repasse alors à 0 avant 1 seconde. Sinon, défaut rodeuse "AMDP à 0 Phase 4".
- La pince se ferme. L'entrée *di\_Rod1\_Rot* doit repasser à 1 et y rester tant que la pince est fermée. Sinon, le défaut "Perte rotation rodeuse" apparaît.

### *Changement Electrodes :*

Le changement d'électrodes est déclenché par l'appel de la routine « *Change\_Elec* », déjà écrite (mais en **commentaire** !) dans la trajectoire de changement électrodes.

On doit spécifier si les pinces doivent être en GO ou en PO (paramètre PO1/GO1). S'il s'agit d'une pince au sol, rien de plus n'est à écrire. Si par contre, il existe une pince embarquée, il faut créer les points de mouvement robot dans la trajectoire.

Dans le cas d'une SR 2 pinces dont une au sol, le changement électrodes peut être exécuté sur les deux pinces simultanément.

Pendant la phase de changement électrodes, la remise en eau est inhibée, et le départ distant également, pour éviter à l'opérateur une douche inattendue. La remise en air, eau et le départ du robot ne sont possibles qu'après une validation expresse de l'opérateur, qui est renforcée par l'introduction d'un code secret si celui ci a été précisé dans le menu "Reglages".

Si le flag *sr\_ROD\_AVT\_CHGT* est à true ou si le choix « Rodage avant changement électrodes ? » a été validé lors de la configuration, toute demande de changement électrodes est précédée d'un rodage, que ce soit avec pince au sol ou embarquée.

Dans le cas d'une application 2 pinces, le menu « SERVICES » permet de changer les électrodes sur la pince 1, la pince 2 ou les deux à la fois ( avec éventuellement rodage de chaque pince).

### *Cas de la soudure sur fausse pince :*

#### a- Pince Pneumatique:

Dans ce cas, c'est un numéro de programme CPS propre à la pince 2 qui est utilisé pour souder. Si une demande de changement electrodes survient, le Standard se comporte comme si une demande sur la pince 2 avait eu lieu. En fin de cycle, il positionne le programme *sr\_PROG\_RODAGE2*, ce qui permettra de reseter le bon compteur de points (et non pas celui de la pince embarquée sur le robot).

#### b- Pince Electrique:

Jusqu'à 3 fausses pinces peuvent être gérées. Il suffit de préciser le numéro de la fausse pince concernée dans l'appel de la routine SOUDE.

Le standard gère le changement de compteur de points de la carte Swit et sa restitution. Si une demande de changement électrodes survient, le robot part en trajectoire de changement électrodes, quelque soit la pince concernée. Le changement électrodes doit être validé pour toutes les pinces pour lesquelles il y a eu une demande (vraies ou fausses).

Résumé des routines accessibles au programmeur:

<b>- Soude:</b>	Effectue une soudure sur le point spécifié.
<b>- MoveP:</b>	Effectue un déplacement robot avec mouvement de la pince synchronisé.
<b>- sr_Marche_Air</b>	Met en marche la vanne d'air.
<b>- sr_Coupe_Air</b>	Coupe la vanne d'air.
<b>- sr_Marche_Eau</b>	Met en marche le circuit d'eau.
<b>- sr_Coupe_Eau</b>	Coupe le circuit d'eau.
<b>- Rodage</b>	Effectue un rodage au point courant.
<b>- Lance_Rodeuse1</b>	Démarre la rodeuse et positionne le programme pour la CPS.

Les actions manuelles suivantes peuvent également être appelées:

<b>- Ferme_Pince:</b>	Ferme la pince avec la consigne de pression spécifiée.
<b>- Ouvre_Pince:</b>	Ouvre la pince spécifiée à la plus grande ouverture possible.
<b>- Ferme_Bascu:</b>	Ferme le basculement si cela est possible.
<b>- Ouvre_Bascu:</b>	Ouvre le basculement si cela est possible.
<b>- Soude_Pince:</b>	Effectue une soudure à l'endroit courant quelque soit la position de la pince.
<b>- Rode_Pince:</b>	Lance la séquence de rodage depuis l'endroit courant.
<b>- Programme_CPS1:</b>	Envoie la consigne de pression spécifiée à la CPS.
<b>- Pince1_PO:</b>	Met la pince spécifiée en petite ouverture.
<b>- Pince1_GO:</b>	Met la pince spécifiée en grande ouverture.

#### Constantes modifiables:

Les constantes suivantes peuvent être modifiées par l'utilisateur. Leur prise en compte est immédiate. Penser à faire une sauvegarde par le menu "Sauvegarder les modifs" après tout changement validé.

NOM	TYPE	APPLI	ACTION
Sr_PinceX_DC	Bool	Sr1p, Sr2p	La pince X est à double course.
Sr_PinceX_basc	Bool	Sr1p, Sr2p	La pince X est avec basculement.
Sr_Ctrl_POX	Bool	Sr1p, Sr2p	La pince X est contrôlée en petite ouverture.
Sr_Ctrl_GOX	Bool	Sr1p, Sr2p	La pince X est contrôlée en grande ouverture.
Sr_Prog_AvecSoud	Num	Sr1p, Sr2p	Numéro de programme CPS appelé après un rodage.
Sr_Maintien_Rode	Num	Sr1p, Sr2p, SrEL	Temps de maintien de la rotation fraise en fin de rodage (sans attente robot).
Sr_Prog_RodageX	Num	Sr1p, Sr2p	Numéro de programme envoyé à la CPS pour le rodage de la pince X ou sur la fausse pince si X=2.
Sr_InstAv_Ferme	Bool	Sr1p, Sr2p	En mode pas à pas en marche avant, la

			pince se ferme sur une instruction Soude.
Sr_Instav_Soude	Bool	Sr1p, Sr2p	En mode pas à pas en marche avant, la soudure est exécutée.
Sr_ExecManuFerm	Bool	Sr1p, Sr2p	En mode exécution continue manuelle, la pince se ferme sur une instruction Soude.
Sr_ExecManuSoude	Bool	Sr1p, Sr2p	En mode exécution continue manuelle, la soudure est exécutée.
Sr_Tps_Max_GOX	Num	Sr1p, Sr2p	Timeout pour génération défaut contrôle GO pince X OU temps d'attente si le contrôle est inhibé.
Sr_Tps_Max_POX	Num	Sr1p, Sr2p	Timeout pour génération défaut contrôle PO pince X OU temps d'attente si le contrôle est inhibé.
Sr_Tps_Max_FcyX	Num	Sr1p, Sr2p	Timeout pour génération défaut attente Fcy CPS.
Sr_DecX_Suff_PO	Bool	Sr1p, Sr2p	Timeout pour génération défaut ouverture en PO pince X quand le contrôle DEC est inhibé.
Sr_Tps_Decol_POX	Num	Sr1p, Sr2p	Temps au bout duquel on tente de ré-ouvrir la pince en PO si le contrôle DEC est inhibé.
Sr_DecX_Suff_GO	Bool	Sr1p, Sr2p	Timeout pour génération défaut ouverture en GO pince X quand le contrôle DEC est inhibé.
Sr_Tps_Decol_GOX	Num	Sr1p, Sr2p	Temps au bout duquel on tente de ré-ouvrir la pince en GO si le contrôle DEC est inhibé.
Sr_Accost_GOX	Num	Sr1p, Sr2p	Temps d'accostage venant de la grande ouverture pince X. (Temps entre la fermeture pince et le start).
Sr_PinceX_Rodmsq	Bool	Sr1p, Sr2p, SrEL	La pince X est en rodage masqué (pince au sol).
Sr_T_Detect_Air	Num	Sr1p, Sr2p, SrEL	Temps pendant lequel un défaut air doit persister pour être signalé.
Sr_T_Montee_Air	Num	Sr1p, Sr2p, SrEL	Temps au bout duquel le contrôle d'air est vérifié après une remise en air.
Sr_T_Montee_Eau	Num	Sr1p, Sr2p, SrEL	Temps au bout duquel le contrôle d'eau est vérifié après une remise en eau.
Sr_Surv_AirX	Bool	Sr1p, Sr2p, SrEL	Activation de la surveillance présence air pince X.
Sr_Surv_EauX	Bool	Sr1p, Sr2p, SrEL	Activation de la surveillance présence eau pince X
Sr_Rod_Avt_Chgt	Bool	Sr1p, Sr2p, SrEL	Activation du rodage avant changement d'électrodes.
Sr_Rod_Nbre_Elem	Num	Sr1p, Sr2p, SrEL	Nombre de rodages élémentaires.
Sr_Tps_Inter	Num	Sr1p, Sr2p, SrEL	Temps d'attente entre chaque rodage élémentaire.

## **6 – Soudure SR Electrique**

L'application Soudure Electrique se décompose en trois variantes principales:

- Avec séquenceur de soudage interne (carte swit).
- Avec séquenceur de soudage externe (Aro).
- Sans carte Swit mais avec séquenceur externe (S4C+ AUTOMOTIVE).

Dans les deux premiers cas, l'asservissement de la pince est assuré par la carte swit.

### *6.1: Avec séquenceur interne (swit)*

Cette variante se décompose elle même en sous variantes:

- 1 carte swit sur liaison série.
- 1 carte swit sur réseau CAN.
- 2 cartes swit sur réseau CAN.

Attention: ces sous variantes sont incompatibles entre elles. Les entrées sorties sont différentes, les modules système de la carte swit sont différents, et les modules applicatifs également.

Le choix se fait lors de l'installation initiale de l'application.

La version liaison série utilise SpotWare, alors que la version CAN ne l'utilise plus.

La donnée de n° de programme est de type *SpotData* dans le premier cas, alors qu'elle est simplement de type *Num* dans le second.

Chacune de ces sous variantes accepte en outre les options suivantes:

- Pince(s) au sol ou embarquée(s).
- Rodage masqué ou non.
- Rodeuse classique ou de type AMDP.
- Support du changeur d'outils.
- 1 changeur automatique d'électrodes sur l'une des pinces.

En sus des constantes communes avec la soudure pneumatique (voir ci dessus), on trouve des données propres à ce process:

Sr_Freq_Secteur	Num	Fréquence secteur (en Hertz).
Sr_Rod_Epaiss	Num	Valeur de l'épaisseur de la rodeuse si rodage masqué (en mm).
Sr_Rod_Force	Num	Valeur de la force lors du rodage masqué.
Sr_Rod_Temps	Num	Valeur du temps de rodage masqué.
Sr_Ferme_Force	Num	Valeur par défaut de la force de fermeture en action manuelle.
Sr_Ferme_Epaiss	Num	Valeur par défaut de l'épaisseur lors d'une fermeture

en action manuelle.

Cas particulier:

Si l'action Rodage a été demandée dans le cadre des actions manuelles ET gâchette relâchée, un menu apparaît permettant de modifier les valeurs de temps, force et épaisseur lors du rodage. Si il y a validation, ces nouvelles valeurs remplacent les valeurs par défaut.

Si les paramètres optionnels de temps, force et épaisseur sont envoyés à la routine Rode\_Pince (sous forme de paramètres optionnels), donc dans une trajectoire, les valeurs envoyées sont prises en compte mais n'écrasent pas les valeurs par défaut.

#### 6.2: Avec séquenceur externe (Aro).

Cette application comporte les sous options suivantes:

- 1 pince embarquée ou fixe avec rodage en temps masqué.
- 2 pinces simultanées, fixes ou embarquées (uniquement avec swit CAN).
- Rodage classique ou avec boîtier AMDP.
- Changeur automatique d'électrodes sur l'une des pinces.
- Changeur d'outils.

Cette application reprends à la fois le fonctionnement d'une carte swit, et la gestion d'un séquenceur externe identique à la soudure pneumatique.

2 séquenceurs peuvent être pilotés en même temps, impliquant l'utilisation de 2 cartes swit (donc forcément sur réseau CAN).

Dans ce cas, tous les paramètres de soudure sont programmés dans la séquence ARO, qui devient identique à une pince pneumatique, mais doit être équipé d'une carte d'entrées sorties supplémentaire et du programme de dialogue adéquat.

Les paramètres deviennent alors accessibles aux spécialistes soudure, sans avoir à manipuler le pupitre robot.

De plus, ils peuvent être modifiés à distance grâce au réseau AroNet.

127 codes soudure peuvent être utilisés au lieu de 31. Par conséquent, le programme de rodage passe à 127 (et 126 et 125 pour les autres pinces éventuelles). Ceci est fait automatiquement lors de l'installation du logiciel.

Cette application utilise tout de même la carte Swit, qui assure les asservissements d'épaisseur et d'effort de la pince.

Le principe de fonctionnement du dialogue lors d'une soudure consiste à indiquer à la séquence le n° de programme soudure, puis à lire les valeurs programmées d'effort et d'épaisseur dans la séquence. Ces valeurs sont ensuite transmises à la carte swit.

Pour information

Le chronogramme des échanges simplifié se présente comme suit:

<b>Sequenceur ARO</b>	<b>Baie S4C</b>	<b>Carte Swit</b>
	← N° de programme + Parité ← Envoi du Start soudure	
	Attente 40 ms.	
Di_Don1_Ok=0	→ La sequence a été modifiée, il faut donc la relire. Sinon, fin du graphe.	
	← Envoi "Dmde d'épaisseur" Attente réponse	
Consigne d'épaisseur Bit de parité Sortie "Données prêtes"	→ → → Prise en compte consigne épaisseur ← Raz "Dmde d'épaisseur" → Attente RAZ "Données prêtes". ← Envoi "Dmde Effort" Attente réponse	
Consigne Effort Bit de parité Sortie "Données prêtes"	→ → → Prise en compte consigne effort ← Raz "Dmde Effort" → Envoi Effort et Epaisseur → Attente RAZ "Données prêtes" Attente acquittement Swit	→ Prise en compte ←
	ARRIVEE AU POINT DE SOUDURE Demande de fermeture pince Attente pince fermée et effort atteint ← Demande de soudure do_Autor_Soud1 → Attente fin soudure ou défaut	→ Prise en compte ←
Envoi FCY soudure	→ Raz Start sauf si défaut soudure Demande d'ouverture pince Attente pince ouverte	→ Prise en compte ←
	Depart robot vers point suivant	

Naturellement, toutes les "attentes" sont limitées par un time out permettant de générer un défaut si l'attente est trop longue.

En cas de défaut, la pince est toujours réouverte, puis le message d'erreur s'affiche. En cas de défaut soudure, donc venant de la CPS, un menu apparaît sur le séquenceur avec le choix "Ressouder", "Sauter". Simultanément, le robot affiche "Attente choix CPS".

Le robot reste bloqué tant qu'aucun choix n'a été fait sur la CPS. Contrairement à l'application SR Pneumatique, c'est donc sur la CPS qu'il faut acquitter le choix, et **non pas sur le robot**.

Le robot sait, grâce à l'entrée di\_Ev\_Pince, quel choix a été validé, et monte ses mémoires en conséquence.

Tout le dialogue avec la CPS prends, comme on s'en doute, un certain temps. Les mesures ont donné des valeurs de l'ordre de 300 ms. De plus, le transfert des données à la carte Swit prends également de l'ordre de 400 ms.

En général, le dialogue se fait en temps masqué, pendant le mouvement robot vers le point. Mais si l'on effectue une couture avec des points proches, le temps de mouvement robot peut devenir plus faible que le temps de dialogue, ce qui entraîne une attente visible, ou des points de passage dégénérés en points d'arrêt.

Pour cette raison, le dialogue a été optimisé au maximum:

- Le dialogue avec la CPS ne s'effectue QUE si les paramètres de soudure de CE programme ont été modifiés dans la CPS OU que c'est la première fois que l'on soude avec ce programme après un Spécial 3.
- 
- Le chargement de la carte Swit ne s'effectue QUE si les données de force ou d'épaisseur sont différents des précédents.
- 

Ainsi, comme on utilise généralement le même numéro de programme sur une couture, ce temps devient nul.

On évitera donc, pour optimiser les temps de cycle, et dans la mesure du possible, de changer le numéro de programme soudure à chaque point.

L'instruction de soudage se présente comme pour la pince pneumatique:

Soude \L, Point, No\_Ident, vitesse, outil, objet.

Par défaut, le mouvement est fait en mode articulaire. L'option \L permet d'effectuer un mouvement linéaire.

Le contrôle DEC est **supprimé**.

Contrairement à la soudure électrique "classique", le paramètre N° Ident est de type **num**, c'est simplement le numéro de programme CPS à appeler.

Tout ce qui est lié à Spotware est supprimé.

L'exploitation des fonctions annexes (rodage, changement électrodes, etc...) est strictement identique à la pince pneumatique.

L'application fonctionne également dans le cas d'une pince au sol ou d'un changeur d'outils, et supporte le contrôle de vitesse de la fraise de rodage (AMDP ou EXROD).

Comme pour la SR électrique "classique", la routine *Init\_Pince* ne doit jamais être appelée dans les trajectoires, ceci étant fait automatiquement par le standard.

### 6.3: Cas de la baie S4CPLUS AUTOMOTIVE:

Dans ce cas, il n'y a plus de carte Swit: la pince est gérée comme un axe externe standard, sauf lors des fermetures pince ou soudure, ou un module système particulier (SERVOGUN) gère l'asservissement de position et d'effort.

Quant à la séquence ARO, elle est reliée à la baie par une liaison CAN et protocole DeviceNet. Il n'y a donc plus de dialogue par E/S parallèles.

Pour cette raison, le dialogue avec la séquence ARO est considérablement simplifié: le robot envoie le code programme, attend 40 ms de sécurité, puis lit directement et en une seule fois les consignes d'effort et d'épaisseur programmés dans la séquence, avant de commander la pince avec ces valeurs, lorsque le robot est arrivé au point.

Le dialogue avec la séquence ARO est extrêmement court, et de plus s'effectue pendant le mouvement robot vers le point visé, donc en temps masqué. Il est donc exécuté systématiquement à chaque point, contrairement à l'application précédente.

#### Limitations:

- 1 – Seule 1 pince simultanée peut être gérée.
- 2 – Le fonctionnement de plusieurs pinces sur un changeur d'outils n'a pas été validé.

## **7 – Changements d'électrodes**

### a) Contrôle de brillance

Le contrôle de brillance est une option qui doit être déclarée lors de l'installation de l'application, ou éventuellement plus tard en mettant le booléen *Brill\_instal* à TRUE et en démarrant l'application (spécial 3) avec la disquette standard présente.

Ce booléen permet également d'inhiber le contrôle de brillance en le mettant à FALSE.

Le dispositif se présente sous la forme d'un capteur optique placé près de la rodeuse. Il indique si le rodage s'est effectué correctement en analysant le niveau de brillance des électrodes.

Le déroulement du rodage se fait donc comme suit :

- Approche de la rodeuse.
- Rodage.
- Déplacement robot.
- Init pince.
- Déplacement vers le détecteur de brillance.
- Fermeture pince (à effort faible !!!).
- Test du détecteur.

Si le contrôle est mauvais, le robot peut reprendre l'intégralité de ce cycle, dans la limite de la donnée numérique *Nb\_Rod\_Elem*, qui donne le nombre de répétitions maximum autorisé. Si cette donnée est inférieure à 1, elle est forcée à 1 par l'application.

Par ailleurs, la donnée num *NbRodOkMin* permet de préciser à partir de combien d'exécutions de la trajectoire de rodage un contrôle mauvais est considéré comme normal. Dans ce cas, un changement d'électrodes en **automatique** est effectué, si bien sûr il y a un changeur. Dans le cas inverse, le message bloquant

#### Contrôle Brillance : Rodage Incorrect

apparaît, la verrine rouge s'allume, la sortie *do\_Def\_Outil* monte, et un changement d'électrodes **en manuel** est exécuté.

Par exemple, si *NbRodOkMin* vaut 10, tout rodage incorrect **avant** la 10<sup>ème</sup> exécution de la trajectoire de rodage provoquera un changement d'électrodes manuel. A partir du 11<sup>ème</sup> rodage, c'est le changement d'électrodes automatique qui sera appelé, sans message particulier, ni alarme.

Un rodage est considéré comme mauvais lorsqu'on a épuisé toutes les répétitions autorisées sans succès. Les compteurs accessibles *Nb\_RodOk* et *Nb\_RodNOK* permettent de voir combien de rodages **complets** se sont bien et mal terminé. Ces compteurs ne sont jamais remis à 0. (provisoire).

La trajectoire de rodage devient alors un peu plus complexe :

```
CONST robtarget
PosRODAGE:=[[9E+009,0,0],[1,0,0,0],[0,0,0,0],[9E+009,9E+009,9E+009,9E+009,9E+009,0]];
VAR bool fin;
VAR num Cpt_Rod;

nom_traj:=NomTraj{ID_T_RODAGE};
IF ApprencyCLE RETURN;
IF Apprentrajs Stop\NoRegain;
Anti_Reboucle;
MoveAbsJ PAbsPEO,v_norm,fine,Tool_Defaut;
!
! Approche de la rodeuse
!MoveJ *...
!cas avec controle brillance
if brill_instal then
  If Enprod Incr NbTrajRod;
fin:=false;
```

```

cpt_rod:=0;
while not fin do
    ! Point de rodage
    ! MoveL PosRODAGE,v_norm,fine,Out_Pincl;
    AffPosition "POINT DE RODAGE";
    "Rode_pinc 1 ;e%" 1;

    !degagement pour init pince

    !moveL *...
%"Initince%" 1;

    !mouvement vers controle brillance

    !MoveJ

    !compteur de rodages elementaires

    Incr cpt_rod;
    Ctrl_Brillance cpt_rod,fin ;
endwhile

else !cas sans controle brillance

    ! MoveL PosRODAGE,v_norm,fine,Out_Pincl;

    AffPosition "POINT DE RODA      Rode_Pince 1 ; GE";
    %"Rode_pince%" 1;
endif
! retour au PEO
MoveAbsJ PAbsPEO,v_norm,z_moy,Tool_Defaut;
IF Apprentrajs Stop\NoRegain;
ERROR
IF ERRNO=ERR_CALLPROC OR ERRNO=ERR_REF

```

**ATTENTION :** Compte tenu de la possibilité de répéter les rodages élémentaires, il peut y avoir discordance entre l'usure réelle des électrodes et la valeur des compteurs de rodage, notamment dans le cas d'un séquenceur externe (ARO).

#### *b) Changeur automatique d'électrodes*

3 types différents de changeur automatique peuvent être gérés par l'application : Exrod, Sciaky ou PCI. Ceci doit impérativement être déclaré au moment de l'installation initiale, car des entrées sorties spécifiques doivent être déclarées.

Quel que soit le type de changeur, celui ci peut être inhibé en forçant le flag « *Chg\_Elec\_Actif* » à FALSE.

La trajectoire de changement électrodes devient un peu plus complexe dans ce cas .

L'exemple fourni ressemble à ceci :

```

PROC T_Chgt_Elec()
  VAR bool ok;

  nom_traject:=NomTraj{ID_T_CHGT_ELEC};
  IF ApprencyCLE RETURN;
  IF Apprentrajs Stop\NoRegain;

```

```

Anti_Reboucle;
!
! Points de la trajectoire
MoveAbsJ PAbsPEO,v_norm,fine,Tool_Defaut;
!pour anticiper la coupure d'eau
%"sr_Coupe_Eau"%;
! Detection etat du changeur
If (NbTrajRod =0) OR (NbRodOkMin=0) OR (NbTrajRod> NbRodOkMin)
%"Debut_Extract"% ok;

IF ok THEN
!-----
! CHANGEUR AUTOMATIQUE PRESENT ET ACTIF ET OK
!-----
! OUVRE_EXTRACT ou AVANCE_EXTRACT;
%"CTRL_PRES_ELEC12"%;
! MoveJ... amener le robot en position pour extraction electrode 1
! FERME_EXTRACT ou RECUL_EXTRACT;
! %"OUVRE_EXTRACT"\Ctrl_Ferme;
! MoveJ... engagement de l'electrode 2
! %"FERME_EXTRACT"%;
! %"OUVRE_EXTRACT"\Ctrl_Ferme;
!-----
! CHARGEMENT DES ELECTRODES
!-----
%"CTRL_PRES_ELEC12"%;
! MoveJ... Prise electrodes 1 & 2 + degagement pince fermee
%"Change_Elec"%;
! Fermer la pince une ou deux fois pour mettre en place les
electrodes
!ferme_pince 1;
%"CTRL_PRES_ELEC12"%;
ELSE !CHANGEMENT MANUEL
!MoveJ...
MoveAbsJ PAbsCHGT_ELEC,v_norm,fine,Tool_Defaut;
!forage change elec par le controle brillance
if change_a_faire AffDefProcess SOUDURE,"Rodage incorrect"\Entete:="
    CONTROLE BRILLANCE "\Nostop;
%"Change_Elec"%;
ENDIF
NbTrajRod:=0;
! Retour au PEO
MoveAbsJ PAbsPEO,v_norm,z_moy,Tool_Defaut;
IF Apprentrajs Stop\NoRegain;
ERROR
IF ERRNO=ERR_CALLPROC OR ERRNO=ERR_REFUNKPRC TRYNEXT;
ENDPROC

```

L'appel à la routine « Debut\_Extract » permet de détecter si :

- Un changeur automatique existe.
- Il est déclaré actif.
- Les magasins sont approvisionnés.

Cet appel est de plus conditionné par le résultat du contrôle de brillance, car le changement d'électrodes doit impérativement se faire en manuel si un défaut de brillance est apparu avant que le robot ait exécuté *NbTrajOkMin* trajectoires de rodage.

Si ces conditions sont réunies, le flag *Ok* est mis à TRUE et c'est donc le changement automatique qui sera réalisé.

S'il n'y a pas de changeur, on peut supprimer tout ce qui s'y rapporte, pour simplifier la trajectoire, et ne conserver que ce qui suit le « ELSE » (cas manuel).

Dans l'exemple fourni, il faut remplacer les instructions en commentaire ou encadrées de % » par les instructions réelles correspondantes. Ex :

% »CTRL\_PRES\_ELEC12 »% sera remplacé par CTRL\_PRES\_ELEC12.

Les instructions spécifiques du changeur apparaissent dans la Liste Commune 2.

Si l'un des magasins est vide, le message « CHANGEUR VIDE » s'affiche et la verrine rouge clignote.

Si le message a disparu, on peut savoir pourquoi la verrine clignote en appuyant sur P1, robot en Auto et En Production.

## **8 - Goujonnage**

La routine principale du goujonnage s'appelle « **SoudeTeteL** ». Elle peut gérer deux têtes de soudage non simultanées. Le robot est amené à la position spécifiée, en mode linéaire, puis le goujonnage est exécuté.

Les paramètres à fournir à cette routine sont :

- |                          |                                 |
|--------------------------|---------------------------------|
| - <b>ToPoint</b> :       | point visé.                     |
| - <b>No_Ident</b> :      | Point B.E.                      |
| - <b>Speed</b> :         | vitesse de déplacement du robot |
| - <b>Tete1   Tete2</b> : | numéro de tête de soudage.      |
| - <b>Tool</b> :          | outil courant.                  |
| - <b>Wobj</b> :          | repère objet.                   |

Il n'y a pas de paramètre de zone puisque la soudure s'effectue sur un point d'arrêt. A l'instar des autres routines de pilotage de process, celui-ci n'est effectivement réalisé qu'en mode exécution continue en avant. En pas à pas ou en mode arrière, seul le mouvement est réalisé.

Certaines fonctions sont également appelables soit dans les trajectoires, soit en commande manuelle:

- Avance\_Tete
- Recule\_Tete
- Soude\_Tete
- Programme\_Tuck

Toutes ces routines doivent recevoir en argument le numéro de tête à commander.

Depuis la version 3.68, les trajectoires de goujonnage sont reprenables: le standard détecte quels goujons ont été réellement soudés et propose un menu permettant de souder ou non sur chaque point douteux.

Les paramètres modifiables du process Goujonnage sont:

- tu\_TETE1\_SURV : surveillance présence tête 1.
- tu\_TETE2\_EXISTE : déclaration d'une deuxième tête.
- tu\_TETE2\_SURV : surveillance présence tête 2.

## **9 - Manutention**

Il n'existe pas de routines complètes de gestion des préhenseurs, le nombre de cas possible étant *à priori* trop important. Toutefois, des routines de base existent et se nomment *Ferme\_Main\_X* et *Ouvre\_Main\_X*. Il est à la charge de l'intégrateur d'écrire le contenu de ces routines selon le type de préhenseur. (dans le module ACTUSITE)

Une routine d'ouverture pourra ressembler à ceci:

```
PROC Ouvre_Main1()
```

```
Reset do_Ferme1;
Reset do_Ferme2;
Reset do_Ferme3;
Set do_Ouvre1;
Set do_Ouvre2;
Set do_Ouvre3;
Attends_Entrees 5,di_ctrl_ouv1,1,"Ouverture serrage 1";
Attends_Entrees 5,di_ctrl_ouv2,1,"Ouverture serrage 2";
Attends_Entrees 5,di_ctrl_ouv3,1,"Ouverture serrage 3";
ENDPROC
```

*NOTA:* La routine *Attends\_Entrees* peut recevoir, sous forme d'argument optionnel, une deuxième entrée à tester simultanément. Toutefois, si le nombre d'entrées est supérieur à 2, il est préférable d'écrire autant d'appels à *Attends\_Entrees* que d'entrées à tester pour une meilleure lisibilité et un dépannage plus facile en cas de problème. La différence de temps de traitement est négligeable.

### ***NOUVEAU DEPUIS V 3.4***

Si le champ commentaire est laissé vide (""), le standard affiche le nom en clair de l'entrée attendue, soit ici "ATTENTE di\_ctrl\_ouv1". On peut donc soit faire afficher le nom de l'entrée physique attendue, soit un texte quelconque à sa convenance.

La routine « **Attends\_PP** » peut être utilisée pour tester de 1 à 5 présences pièces simultanément. Cette routine gère le mode sans pièce. Il est donc inutile d'en faire le test dans les routines utilisateur.

Si un présence pièce est attendu à 1, il est ensuite surveillé en permanence : la retombée du signal provoque un stop du robot avec message à acquittement obligatoire :

« Perte Presence Piece X »

X est alors le **nom réel** de l'entrée physique attendue, par exemple *di\_PP1*. (depuis version 3.4 ET si les fonctions Developpeur sont résidentes).

Il est alors possible de relancer le robot, même si le présence pièce est toujours à 0. Par contre, s'il passe à 1 **puis** retombe à 0, un nouvel arrêt est généré (surveillance par interruption sur front descendant). La surveillance cesse après un appel à **Attends\_PP** avec état attendu à 0.

#### ATTENTION:

Il n'est pas expressément interdit d'utiliser la routine **Attends\_PP** à 1 sur un point de passage, (cas de la prise de pièce) toutefois ce fonctionnement n'est pas logique et risque d'entraîner des défauts de point de "passage dégénéré en point d'arrêt", si les contrôles sont absents au moment du test, ce qui sera probablement le cas, voire des défauts système plus graves. On veillera donc à ce que le point qui précède l'appel à **Attends\_PP** soit bien déclaré en point d'arrêt, s'il s'agit d'une prise de pièce.

Avant de lâcher une pièce, il est souvent nécessaire d'inhiber la surveillance des présences pièces, sous peine de tomber en défaut. Cette inhibition se fait par un appel à la routine **Attends\_PP** avec la valeur (-1):

Attends\_PP di\_SQ1,-1;                         inhibe la surveillance de *di\_SQ1*.

Une fois la pièce lâchée, on peut contrôler que le détecteur est bien à 0:

Attends\_PP di\_SQ1,0; (*Qui peut se faire sur point de passage*).

Un lâcher de pièce classique pourra donc s'écrire sous la forme:

Attends_PP di_SQ1,-1;	Inhibe le contrôle présence pièce
Ouvre_Main 1;	Ouvre le préhenseur et teste les contrôles d'ouverture
MoveL pRecul...	Dégage le robot du montage
Attends_PP di_SQ1,0;	Vérifie que le présence pièce est bien à 0.

NOTA : Dans cet exemple, le point pRecul peut être exécuté en point de passage : le robot ne s'arrêtera que si le contrôle est incorrect (ou pendant le temps de retombée du présence pièce).

Les autres présence pièces à déclarer sont accessibles par les paramètres optionnels de la routine.

Cas du test de présences pièces successifs:

Si l'on doit tester et surveiller un présence pièce, puis un deuxième un peu plus loin dans la traj, tout en continuant à surveiller le premier, il est nécessaire de rappeler le nom du premier, sinon sa surveillance est annulée. Il faut écrire:

```
Attends_PP di_PP1,1;  
Move...  
Move...  
Attends_PP di_PP1,1\Nompres2:=di_PP2\Etat2:=1;
```

Si l'on écrit

```
Attends_PP di_PP1,1;  
Attends_PP di_PP2,1;
```

Alors seul le di\_PP2 sera surveillé dynamiquement.

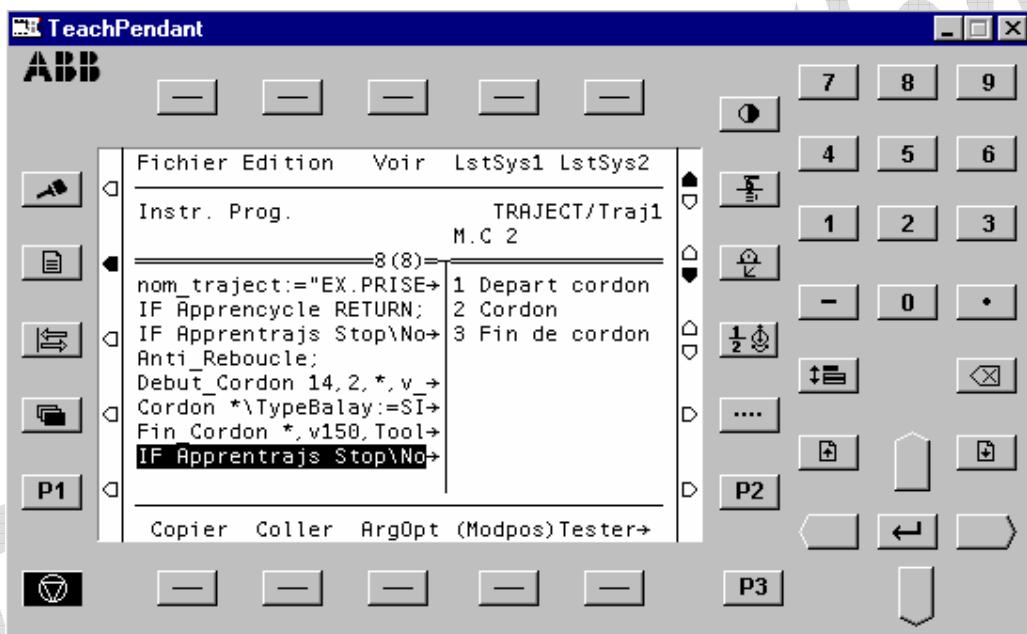
Par contre, si tous les présences pièce sont testés dans le même appel à Attends\_PP, ils sont tous surveillés. (Parallélisme en cas d'appel groupé et écrasement en cas d'appel successif).

## **10 – Soudure ARC :**

Le standard permet de piloter un poste COMMERCY de type 385 MPA ou 386 MPRA. Les routines à appeler sur trajectoire sont :

- Debut\_Cordon.
- Cordon
- Fin\_Cordon.
- Hors\_Zone\_Decras

Leur appel est direct à partir de la liste MC 2 :



Ces routines incluent un mouvement robot et ne peuvent donc pas être appelées à partir d'un cycle. (Ceci est vérifié par le Standard et entraîne un défaut en cas d'infraction).

**NOTA :** A l'inverse des autres process continus (encollage), il est possible de souder aussi bien en Manuel qu'en Auto, y compris sur reprise, car le process Arc utilise des vitesses faibles qui peuvent être identiques dans les deux modes. Il est donc toujours autorisé de souder en Manuel.

Le **Présence Torche** est contrôlé exactement de la même manière que pour les autres process, et peut être inhibé par le menu « Reglages ».

La routine **Debut\_Cordon** amène le robot au point spécifié, puis déclenche l'amorçage de l'arc, avec une légère anticipation du code programme et du gaz par rapport à la demande d'amorçage.

Si le robot est en mode Manuel ou en Apprentissage et que la clé est en mode Avec Soudure,, une confirmation est nécessaire :

« Voulez vous souder ce cordon ? »

L'acceptation du choix déclenchera la soudure réelle, qui est donc possible en Manuel.

Le code programme de soudage **doit** être spécifié dans le paramètre NoProg\_Soud (de type *num*).

La version 3.1 /21 du logiciel de base n'acceptant pas la fonction **ArgName()**, permettant de récupérer le nom d'un paramètre, le nom du cordon ne peut être précisé que sous forme d'un paramètre optionnel (string *Nom\_Cordon*).

S'il est précisé dans l'appel de la routine **Début\_Cordon**, il apparaîtra lors de l'amorçage, en cours de soudure et en fin de trajectoire en cas de défaut.

L'arrivée sur le point se fait en mode articulaire par défaut, mais peut se faire en mode linéaire si le paramètre optionnel \L est spécifié.

Un départ de cordon se fait forcément sur point d'arrêt, il n'y a donc pas à préciser de zone.

La séquence d'amorçage teste si les conditions initiales sont réunies (présence gaz, présence fil, défaut bloquant...), puis active la commande de fil et la sortie **do\_Arc\_Start**.

Le contrôle fil n'est plus géré par le robot, mais directement par le poste.

Si une condition initiale manque, il est possible de la sauter à condition d'être en mode manuel. (fonctionnement classique de « Attends\_Entrées »). Par contre, en mode auto, toutes les conditions doivent être présentes.

Après l'envoi du Start, le programme attend jusqu'à 5 secondes l'arrivée de l'info **di\_Pres\_Ri\_Gen**, signifiant que l'arc est amorcé.

Dès que l'info arrive, le robot exécute le Move qui suit. Notez bien que si aucune instruction de mouvement suivant n'a été programmée, le robot soude **sur place**.

Si l'entrée n'arrive pas au bout de ce temps, les sorties Gaz, Fil et Start sont remises à 0. Si c'est la première fois qu'un défaut apparaît sur le cordon en cours, le robot attend 0.5 seconde, puis retente un amorçage. Si celui-ci est toujours incorrect après 5 secondes, un message bloquant « **Defaut Amorçage** » apparaît, avec consignation dans le fichier de suivi et arrêt des asservissements.

Lors de la relance du programme, un menu apparaît, proposant les choix :

F1 - RETENTER L'AMORCAGE  
F2 - ABANDONNER LE CORDON.

Le choix 1 permet de relancer la séquence d'amorçage, et de re-boucler éventuellement indéfiniment si le défaut persiste.

Le choix 2 permet de réaliser le cordon **sans souder** : la trajectoire se poursuit, mais la soudure est inhibée. Le défaut est mémorisé et le message « Continuer, Refaire » apparaîtra en fin de trajectoire.

La soudure est néanmoins ré-activée dès le début du cordon suivant, sauf si l'on a mis le commutateur sur « SANS SOUDURE ».

Une fois l'arc amorcé, le programme vérifie toutes les 0.5 secondes les entrées suivantes :

- Présence Gaz
- Défaut Gaz Non Bloquant.
- Mode Avec / Sans soudure.
- Défaut Gaz Bloquant.
- Générateur Ok.
- Défaut Générateur.

La perte, même momentanée, de l'une de ces infos entraîne l'apparition du message « Continuer », « Refaire » à la fin de la **trajectoire** en cours, sauf pour les 3 derniers, où le défaut est signalé dès la fin du **cordon** (en plus).

Aucun de ces défauts n'entraîne l'arrêt immédiat du robot. (Seul le Présence Torche le fait).

En cas de passage en mode Sans Soudure en cours de cordon, la soudure est immédiatement arrêtée, mais le robot continue sa trajectoire. La soudure ne sera ré-activée qu'au prochain cordon et si la clé est remise sur Avec Soudure.

Notez que la perte de l'une de ces informations -sans retour- se traduira de toutes façons par un arrêt bloquant dès le début du cordon suivant.

L'instruction **Fin\_Cordon** permet de continuer la soudure en cours, puis **d'arrêter** l'arc. Ceci se fait forcément sur point d'arrêt. Le mouvement est **linéaire** par défaut. On peut toutefois spécifier un mouvement en arc de cercle avec le paramètre optionnel \PtPassage.

Le fonctionnement est alors le même que pour l'instruction MoveC, excepté le fait qu'en mode pas à pas en marche avant, le robot **s'arrête** sur le point intermédiaire de l'arc de cercle, ce qui permet de le modifier aisément. C'est pour cette raison que ce point est obligatoirement un point nommé.

Le paramètre optionnel « \Anticratere » permet de changer le numéro de programme de soudure à la fin du cordon, permettant de réaliser l'anti-crâtere avec d'autres paramètres que ceux du cordon lui-même. (vitesse fil, tension...). Par défaut, les paramètres du cordon en cours sont utilisés.

Le gaz est coupé **après** la séquence d'anticrâtere, et **avant** de tester le fil collé.

Si du balayage était en cours, il continue : l'instruction Fin\_Cordon tient compte des paramètres de balayage précédents. Si l'on souhaite terminer un cordon sans balayage alors que du balayage est actif, un appel à la routine **Cordon** sans balayage est nécessaire.

Voir la paragraphe *Balayage* un peu plus loin dans ce chapitre.

## IMPORTANT

L'instruction **Fin\_Cordon** provoque également le dégagement du robot dans l'axe Z- de l'outil, de 50 mm, après avoir testé l'information « Fil Collé ». Cette fonctionnalité peut être inhibée en déclarant le paramètre optionnel **\Sans\_Recul** dans l'appel de **Fin\_Cordon**.

Entre les instructions **Début\_Cordon** et **Fin\_Cordon**, on peut insérer autant d'instructions de mouvement « classiques » que souhaité : MoveJ, MoveL, MoveC. La soudure qui a été activée le reste, et les surveillances continuent. Un arrêt robot au cours de ces mouvements provoque l'arrêt de l'arc immédiatement.

Toutefois, si l'on souhaite soit changer de programme de soudure en cours de cordon, soit effectuer du balayage, il est nécessaire de faire appel à la routine « **Cordon** ».

Cette routine permet :

- D'aller au point visé en ligne droite ou en arc de cercle.
- De changer le numéro de programme de soudure dès le **début** du tronçon.
- D'effectuer le parcours en superposant du **balayage**.
- De spécifier les **caractéristiques** du balayage souhaité. (Type, Angle, Pas, Amplitude)

Il est important de noter que toutes les caractéristiques du cordon sont prises en compte pour **rejoindre** le point déclaré, et non pas à partir de ce point.

*Balayage* :

Les instructions **Cordon** et **Fin\_Cordon** permettent de spécifier du balayage, qui viendra se superposer au mouvement normal.

La direction du balayage dépend :

- De la direction du cordon.
- Des angles initiaux et finaux de la torche.
- Du paramètre *AngleBalay* déclaré dans la routine.

Les paramètres optionnels des instructions **Cordon** et **Fin\_Cordon** concernant le balayage sont :

- |             |                                  |
|-------------|----------------------------------|
| - TypeBalay | (Sinus, Triangle, Sans_Balayage) |
| - Pas       | (en millimètres)                 |
| - Amplitude | (en millimètres)                 |
| - Angle     | (en degrés).                     |
| - Inverse   |                                  |

Le paramètre TypeBalay permet de choisir entre deux motifs de balayage : pseudo-sinusoidal ou triangulaire.

Il est également possible d'indiquer « Sans\_Balayage », pour l'inhiber localement.

Remarquez que lors de l'apprentissage, seuls ces trois choix sont proposés...

Le paramètre Pas permet d'indiquer de combien de millimètres la torche va avancer au cours d'une oscillation. Ce paramètre ne doit pas être négatif, sinon l'effet obtenu risque de surprendre...

Le paramètre Amplitude spécifie de combien de millimètres la torche va se déplacer autour de l'axe du cordon. Ce paramètre peut être négatif : dans ce cas, le mouvement est inversé.

Le paramètre Inverse change le sens de démarrage du balayage : au lieu de démarrer vers la droite du cordon, par exemple, la torche partira d'abord vers la gauche. Ce paramètre peut être utile pour affiner les raccordements entre tronçons d'un même cordon.

Le paramètre Angle précise l'angle du balayage par rapport à l'axe du cordon. Si aucune valeur n'est spécifiée ou si Angle est nul, le balayage s'effectue d'avant en arrière par rapport à l'axe du cordon.

Si il vaut  $90^\circ$ , le balayage s'effectue perpendiculairement à l'axe du cordon. Toutes les valeurs sont admises pour Angle, y compris négatives.

Cet angle modifie donc la forme du balayage. Par exemple, avec un motif Triangle et un angle de  $45^\circ$ , on obtient une dent de scie.

NOTA : Si l'on souhaite changer un paramètre de balayage en cours de cordon, il n'est pas nécessaire de ré-écrire tous les paramètres, mais seulement celui ou ceux que l'on souhaite changer.

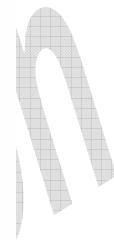
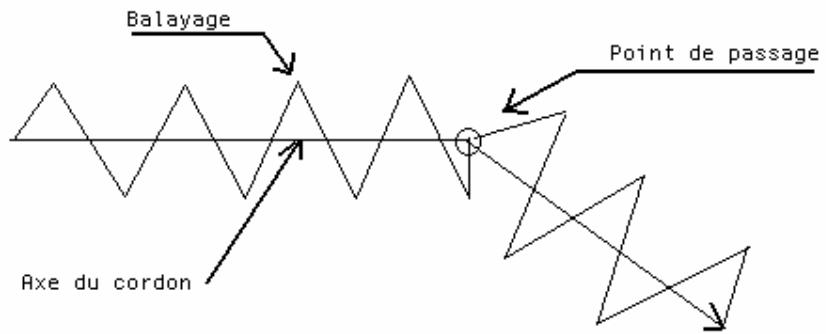
Le paramètre de zone n'a aucune influence en cas de balayage. Il ne sert que pour définir le rayon du point de passage si l'on est sans balayage.

#### ATTENTION

Le balayage sur des arcs de cercle n'est pas opérationnel en version 3.6. Il est inhibé lors de l'exécution. Il est toutefois possible d'approximer un arc de cercle en créant des points rapprochés avec une zone importante.

#### Raccordement entre tronçons d'un même cordon avec balayage :

Lorsqu'on fait du balayage, le problème se pose si plusieurs tronçons se suivent. En effet, le motif de balayage ne se superpose généralement pas parfaitement avec le point appris, ce qui peut entraîner des problèmes de raccordement :

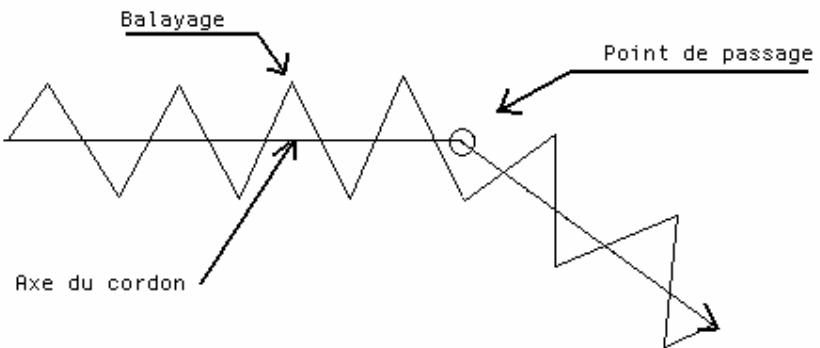


Dans cet exemple, issu de T1, on voit que le raccordement au point de passage n'est pas très satisfaisant...

Le standard PCA T5 essaie de résoudre ce problème en ne tentant pas de passer par le point appris, dans le cas où plusieurs instructions **Cordon** se suivent. Le robot ne rejoint la position apprise **que sur une instruction Fin\_Cordon**.

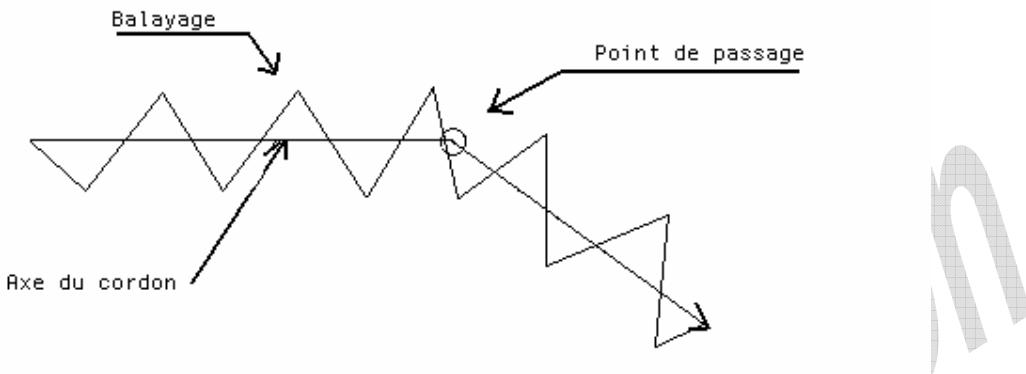
De plus, le standard s'arrange pour que la première crête d'un tronçon soit du côté opposé à la position courante, c'est à dire que **le balayage est continu** sur les points de passage.

Le motif peut alors ressembler à ceci :



Dans cet exemple, on voit qu'il y a déjà un mieux, mais que le balayage « tombe mal », car on a toutes les chances d'avoir un manque matière au niveau du point de passage.

Pour qu'il tombe bien, il aurait fallu inverser le sens du balayage. Le paramètre \Inverse sert justement à cela !



Dans cet exemple, on a utilisé le paramètre \Inverse sur le premier tronçon. On voit (à la qualité du dessin près), que le résultat est nettement meilleur, le remplissage du cordon étant bien plus régulier au niveau du point de passage.

On aurait également pu inverser le sens sur le deuxième tronçon.

Pour affiner la qualité, on peut ajouter des points intermédiaires, et jouer sur le paramètre *Pas*.

Dans le cas de nombreux tronçons enchaînés, il est clair que la mise au point des raccordements peut être délicate.

ATTENTION : le motif de balayage dépend de l'angle Z de la torche par rapport au cordon. Si, par exemple, l'orientation de la torche varie d'un point au suivant, le motif du balayage sera également modifié progressivement.

Ceci permet d'effectuer du balayage selon une hélice, (le plan du balayage reste perpendiculaire à celui de la torche) mais par contre modifie le motif si la torche tourne autour de son axe Z durant le mouvement.

Une trajectoire de soudure Arc pourra ressembler à ceci :

```
PROC Traj1()
Nomtraj := « Soudure renfort AR Gauche »;
```

...

...

```

Debut_Cordon *\Nom_Cordon := »Debut virole »,24,v_soud, torche, Wobj0 ;
Aproche et Debut cordon avec
programme n° 24
Cordon *,3,v_soud...
Change le n° de programme, puis soude.
Cordon *\TypeBalay :=SINUS\AngleBalay :=90,Pas :=20, Amplitude :=50...
Soude en balayant avec un motif
sinusoïdal, un pas de 20 mm et une
amplitude de 50 mm de part et d'autre de
l'axe du cordon, le balayage étant incliné
de 90° par rapport à cet axe.

```

```

Fin_Cordon *\Anticratere :=5,v_soud,torche,Wobj0 ;
Soude vers le point visé avec le numéro de
programme courant, puis active le numéro
de programme 5 pour l'anti-cratère, coupe
la soudure, puis dégage le robot de 50 mm
dans l'axe Z- de l'outil courant.

```

MoveJ...

MoveJ...

Naturellement et comme tous les autres process, le numéro de cordon et le numéro de programme peuvent être déclarés explicitement en constantes globales, par exemple :

```

CONST string Renfort_1 := « Renfort 1 »;
CONST num Prg_Renfort_1 :=24 ;

```

Dans ce cas, on pourra écrire :

```
Debut_Cordon \Nom_Cordon :=Renfort_1, Prg_Renfort_1...
```

*Gestion des reprises après arrêt :*

Lors d'un arrêt robot (perte autorisation de zone, stop, AU...) et si un cordon était en cours, l'arc est immédiatement stoppé. Lors du redémarrage, le choix classique apparaît :

#### REPARTIR AVEC SOUDURE ?

Il est alors possible de terminer le cordon avec ou sans soudure, et ceci en manuel aussi bien qu'en auto. Si, au cours de l'arrêt, le commutateur a été placé sur SANS SOUDURE, ce choix n'est pas proposé et le cordon se termine sans souder.

La séquence de redémarrage de l'arc est strictement la même que pour une instruction de Depart\_Cordon. (Attente présence fil, Gaz, etc...). En cas d'échec du ré-amorçage de l'arc, tout se passe exactement comme pour le cas d'un démarrage normal. (choix Abandonner le Cordon si défaut d'amorçage).

En cas de défaut d'amorçage répété ayant entraîné le choix « Abandon Cordon », le cordon est terminé sans souder et le menu « Continuer, Refaire » s'affiche en fin de trajectoire.

### *Décrassage buse :*

Le décrassage consiste à souffler de l'air mélangé à de l'huile dans la buse pour enlever les résidus de soudure.

Il n'y a pas de fraise de nettoyage.

La trajectoire de décrassage buse est modifiable par le menu Apprentissage, Trajectoires de Service. Elle doit faire appel à la routine **Decrasse** avec les paramètres suivants :

- |           |                                    |
|-----------|------------------------------------|
| - NbCoups | Nombre de coups de soufflage       |
| - Duree   | Durée de chaque coup de soufflage. |

En production, cette trajectoire est appelée automatiquement à la fin d'un cycle selon la fréquence indiquée dans la constante num **NbCyc\_Av\_Decr**. La valeur 0 inhibe la fonction (pas de décrassage cyclique).

De plus, une alarme est signalée 5 cycles avant l'appel réel du décrassage.

Si la torche est fixe et qu'il existe un bras de décrassage, il faut déclarer le Décrassage en Temps Masqué soit au moment de l'installation, soit en changeant le flag **Arc\_Decrass\_Masq**. Attention : si vous modifiez ce flag, pensez à vérifier la déclaration des entrées sorties. En effet, le programme Install déclare ces signaux en Virtuel si le décrassage masqué est inactivé, et en réel sinon.

Dans ce cas, le décrassage masqué est autorisé par défaut dès que le robot rejoint le DPO. Le départ du robot en cycle ne peut se faire que si l'entrée Bras Reculé (di\_Ab\_Dec\_Re) est à 1. Si l'on souhaite pouvoir anticiper le décrassage masqué en cours de cycle, il suffit d'appeler la routine **Hors\_Zone\_Decras** dans une trajectoire, dès que le robot est hors zone du bras.

Si le robot peut revenir dans la zone du bras un peu plus loin dans le cycle, la sécurité impose de tester le recul du bras par un appel à la routine **Dans\_Zone\_Decras**. Cette routine attend la fin du décrassage masqué si celui-ci est en cours, puis vérifie le recul du bras. Celle-ci ne devrait pas être d'une utilisation fréquente, elle n'apparaît pas dans la liste des instructions et il est nécessaire de l'appeler par **ProcCall**.

### *Trajectoire de contrôle :*

Comme pour la soudure par points, la trajectoire de contrôle est appelée cycliquement selon la valeur donnée à **NbCycles\_Av\_Cont**. La valeur 0 inhibe la fonction. Cette trajectoire peut servir à contrôler le bon alignement de la buse sur une pointe de référence, ou à lancer la trajectoire de calcul automatique du centre outil à l'aide du Bulls Eye, si celui-ci est installé.

### *Actions manuelles :*

Les actions manuelles disponibles sont :

- Décrassage buse.

- Avance du fil (pendant une certaine durée)
- Activation / Arrêt du gaz (sans contrôle).
- Avance masse.
- Recul masse

## **11 – Rivetage électrique :**

La seule routine à utiliser dans les trajectoires s'appelle **SertitRivet**. Calquée sur le modèle du clinchage, elle amène le robot au point visé, puis effectue le rivetage si les conditions sont réunies.

Elle est disponible dans la liste M.C.2, comme les autres routines Process.

Elle gère les modes sans pièce, sans rivet et manuel avec confirmation du rivetage effectif.

Le numéro de programme de soudure est spécifié soit sous forme littérale, soit en déclarant une variable de type num dont le nom sera le numéro du point, et le contenu le numéro de programme. Exemple :

SertitRivet \*,1,v\_norm,Tool,Wobj ; sertit avec le programme 1.

*Ou bien :*

CONST num r1234 :=1 ;

SertitRivet \*,r1234,v\_norm,Tool,Wobj.

Il n'y a pas à spécifier de zone puisque le rivetage se fait forcément sur point d'arrêt.

Depuis la version 3.68, les trajectoires de rivetage sont reprenables, à l'instar des process goujon et clinchage.

## **12 – Depalettisation:**

Le process dépalettisation comporte deux options radicalement différentes:

- 1 – Capteur de pile vide sur le container.
- 2 – Capteur de pile vide embarqué sur le robot.

### 1- Capteur fixe sur container.

Dans ce cas, l'automate est maître du container et de la pile. Seuls 2 containers sont possibles. L'automate envoie un code cycle différent pour chaque **pile** et chaque **container**. On aura ainsi, par exemple:

Container 1 / Pile 1 → Code cycle 1  
Container 1 / Pile 2 → Code cycle 2  
Container 2 / Pile 1 → Code cycle 3  
Container 2 / Pile 2 → Code cycle 4

C'est le module Depbase.sys qui est utilisé.

Il est donc du ressort de l'automate d'envoyer un code "valide", c'est à dire correspondant à un container non vide.

### 2 – Capteur embarqué sur le robot:

Dans ce cas, 10 containers de 10 piles chacun sont possibles. L'automate envoie seulement un code cycle **par container**. Toute la gestion du container est assurée par le robot (détection des piles vides, détection de la fin container).

C'est le module Dep2base.sys qui est utilisé.

L'automate envoie également une autorisation de zone, qui peut être la même pour toutes les piles, ou spécifique à chaque pile. Ceci est traité de façon habituelle dans la routine *AppelleTraj*.

### 1- Crédit des programmes:

La première chose à faire est de créer les points de début et de fin de chaque pile de chaque container. Pour cela, valider le menu "Apprentissage", "Points Spéciaux", "Points Palettes".

Un second menu vous permet de choisir la palette concernée (1 ou 2), puis vous pouvez choisir les points de début et de fin de chaque pile.

Le principe est le même que pour les points spéciaux (PEO, Repli...). Pour chaque point, vous pouvez accepter de rejoindre le point existant s'il a déjà été créé, puis le programme s'arrête pour vous permettre de positionner le robot, puis, en appuyant sur "Exec", le programme vous demande de valider ou non la nouvelle position.

Cet apprentissage ne peut se faire qu'en mode Manuel.

Le point de début de pile doit être créé un peu au dessus de la pièce la plus haute, mais pas trop pour le temps de cycle (quelques centimètres).

Le point de fin de pile doit être créé sur la position de prise de la dernière pièce, ou quelques centimètres plus loin.

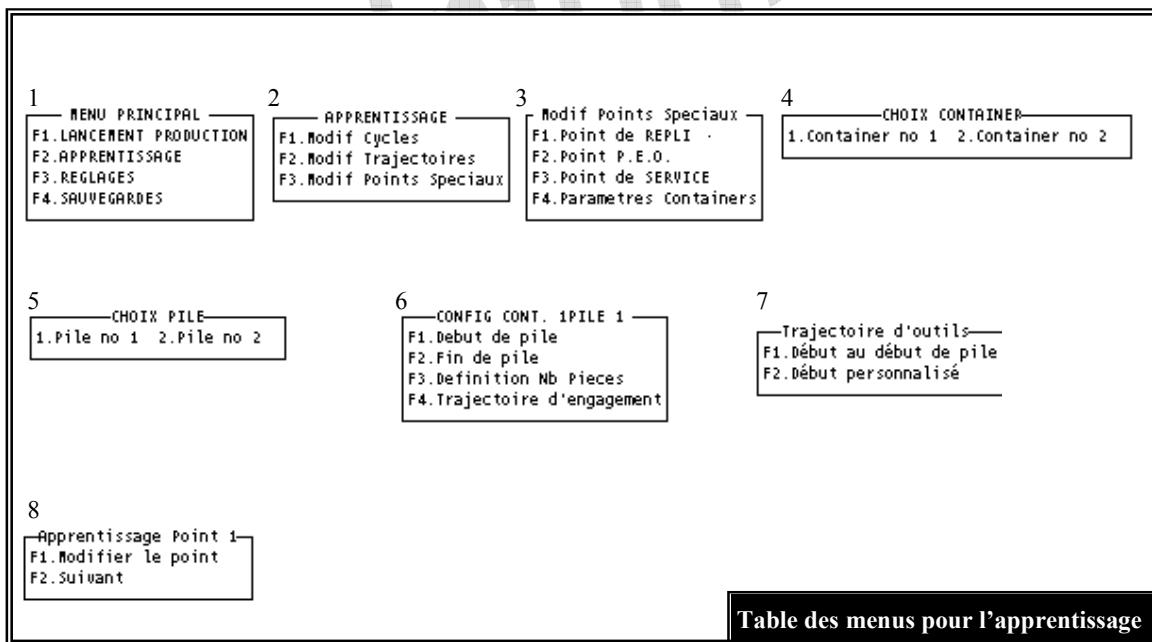
S'il s'agit d'une pile "glissante", il est préférable d'apprendre d'abord le point de début de pile (pile pleine), puis de vider la pile et d'apprendre le point de fin de pile sur la dernière pièce.

Attention: les mouvements de l'un vers l'autre se font toujours en mode **linéaire**. Mais pendant ces mouvements, les contrôles de configuration poignet sont **inhibés**.

Il peut donc arriver que le mouvement de retour vers le premier point de la pile se fasse avec une configuration du poignet différente (axe 4 inversé), si l'on passe par un point singulier sur le parcours.

Ceci n'est pas trop gênant en apprentissage, mais peut le devenir en exécution. On obtient en effet une rotation des axes 4 et 6 de 180 °. On s'efforcera donc au moment de la conception du site de ne pas faire passer le poignet par des points singuliers entre le début et la fin de pile.

A part le fait que l'on peut être amené à vider la pile pour créer le point de fin de pile, l'apprentissage des 8 points ne prends pas plus de quelques minutes.



Il faut ensuite "créer" les trajectoires. Procédez comme d'habitude, en choisissant le n° de trajectoire à créer, par ex 10.

Dans cette trajectoire, insérez l'instruction "*Depalettise*", disponible dans la liste commune 2.

Il faut alors préciser:

- Le n° de container concerné (1 à 10).
- Le n° de pile dans ce container; (si l'option « capteur sur container » a été choisie)
- Le n° de main à piloter pour la fermeture du préhenseur (*Ferme\_Main X*).
- Le nom de l'entrée utilisée pour la détection de pièce (OPTIONNEL). Par défaut, c'est l'entrée *di\_Capteur\_Inductif* qui est utilisée.
- La vitesse d'approche au premier point de la pile. (Généralement Vmax)
- La vitesse de recherche de la pièce suivante de la pile. (V100).

Le paramètre optionnel \J permet de rejoindre le point mémorisé précédent, et le point de début de pile en fin de recherche, en mode **articulaire** au lieu de linéaire, ce qui peut améliorer le problème des points singuliers.

#### **ATTENTION:**

Comme dans ce cas, les axes sont synchronisés, le mouvement du robot peut surprendre. Ce paramètre est donc à manier avec précautions. Notamment, bien faire une vérification en Apprentissage de trajectoire que tout se passe bien quelque soit la pièce trouvée (milieu et fin de pile).

Comme l'approche sur le point d'entrée de pile se fait en mode linéaire, il peut arriver, selon la config du robot au point précédent, que celui ci ne soit pas atteignable. Pour éviter cela, insérer l'instruction:

MoveJ Pt\_Debut\_Cont {X,Y},vitesse,...

Où X est le numéro de container, et Y le numéro de pile, ou bien créez un point d'approche.

La programmation est terminée, sauf à écrire les cycles avec les trajets et zones voulues.

#### 2 – Exploitation

Le principe est d'effectuer une recherche systématique de la pièce suivante, jusqu'à ce que le capteur *di\_Capteur\_Inductif* soit à 1, ou que l'on arrive en fin de pile.

Si le capteur de pile vide est embarqué sur le robot, celui ci détecte la prise de la dernière pièce de la pile courante. A l'exécution suivante, il ira donc directement au dessus de la pile suivante. Si elle est vide, le robot va au dessus de la pile suivante, et ainsi de suite jusqu'à ce qu'il trouve au moins 1 pile non vide. Si toutes les piles sont vides, le robot affiche un défaut bloquant ("Container vide") et le programme est ré-initialisé. Il est en effet anormal que l'automate ait demandé d'aller chercher une pièce dans un container vide.

Lorsque le robot prend la dernière pièce de la dernière pile, il monte la sortie *do\_Reset\_Pal* pour informer l'automate que le container est vide.

Après un Special 3, un retour au Repli, ou un changement de container courant (c'est à dire que c'est la première fois que l'on vient sur le container 2 alors que l'on travaillait sur le 1, et

réciproquement), la première pièce de la pile est recherchée systématiquement, car on ignore où elle se trouve. La pile peut être à moitié vide, par exemple.

*Reset automatique:*

**- Par AZ double :**

Le reset des containers par autorisation de zone est utile quand le nombre de container est au maximum de 2.

La recherche de pièce peut être ré-initialisée automatiquement en déclarant des valeurs non nulles dans le tableau **No\_Raz\_Cont{2}**

Ces valeurs représentent le **numéro de zone** spécifique à chaque container que l'automate doit monter pour provoquer le reset des mémoires. Ce reset pourra être lancée à chaque franchissement des barrières, par exemple, prouvant ainsi qu'un opérateur a modifié le contenu du container.

Si cette autorisation de zone "spéciale" est présente, le robot rase ses mémoires, descend le Hors Zone correspondant, et attends que l'autorisation de zone automatique soit retombée, puis remonte son Hors Zone. A la fin du dialogue, les E/S sont donc dans le même état qu'au début. Cette autorisation de zone spéciale n'est pas surveillée dynamiquement comme dans le cas classique.

Les valeurs par défaut sont zone 10 pour le container 1 et zone 9 pour le container 2. La valeur 0 est autorisée (pas de reset automatique).

Il n'est pas possible de reseter les deux piles en même temps par les AZ9 et AZ10.

**- Par AZ unique :**

Le reset des containers par AZ unique est utilisé quand le nombre de containers est supérieur à deux. Le fonctionnement général de ce type est identique au mode avec deux AZ. Seul le dialogue avec l'automate change.

Pour reseter un container, l'automate doit envoyer l'AZ 10 au robot qui, le détectant, resettera la pile dans laquelle il va travailler. La recherche reprendra donc au début. Pour répondre à la montée de l'autorisation de zone 10, le robot prend l'accès zone 10 et la rend cent millisecondes plus tard. Le robot attend que l'automate fasse retomber son AZ10. Si aucun retour d'information n'est reçu, le robot affiche un message d'erreur.

Attention : Lors de la création des cycles, une ligne est à ajouter uniquement dans les cycles de dépalettisation (en gras) :

```
PROC CYCLE160
  nom_travail:="CYCLE VIDE";
  IF Apprentrajs RETURN;
  Teste_raz_Depal 1;
  AppelleTraj 5,2 ;
  ...
  IF Apprencycle Stop\NoRegain;
ENDPROC
```

Cette ligne en gras permet le reset du container. Le numéro qui suit (ici 1 par exemple) doit correspondre au container dans lequel le robot va travailler. Si cette ligne manque, le reset du container ne se fera pas.

La raz de la pile est possible jusqu'au moment où le robot commence son mouvement vers le point de début de pile. Après, c'est trop tard...

Le robot rejoint systématiquement le point de début de pile (en point de passage z10) à la vitesse *V\_d\_Approche*.

Avant la recherche, le standard vérifie que le capteur *di\_Capteur\_Inductif* monté sur le robot est bien à 0. On tombe dans une boucle d'erreur sinon.

La recherche est ensuite lancée. Elle est effectuée en ligne droite entre le point de début de pile et le point de fin de pile, à la vitesse spécifiée *V\_de\_Recherche*.

Les piles peuvent donc être quelconques: verticales, horizontales, ou même "glissantes" (cas des pièces rangées verticalement les unes contre les autres, et qui ont tendance à glisser). Dans ce cas, les points de début et de fin de pile ont dû être appris avec l'angle correspondant.

Dès que la pièce est trouvée, la main de préhension spécifiée est fermée, contrôlée, puis le robot rejoint en ligne droite et à la vitesse spécifiée *V\_d\_Approche* le point de début de pile, qui est en point de passage.

La routine se termine alors et la main est rendue à la trajectoire.

Lors de l'exécution suivante de la trajectoire (ou de l'instruction *Depalettise* avec les mêmes paramètres), le robot passe par le point de début de pile en point de passage à la vitesse *V\_d\_Approche*, puis rejoint directement à la même vitesse le point où la pièce précédente (de la même pile) a été trouvée.

On peut demander au robot d'anticiper ce point avec un décalage configurable par la constante de type num **ParamApproch**. Le robot s'arrêtera **ParamApproch** centimètres **avant** le point de prise précédent.

Une nouvelle recherche est alors lancée jusqu'à la pièce suivante.

En AZ Unique, le standard gère les mémoires de position des piles du même container: il est donc tout à fait possible de travailler alternativement sur chaque pile, le robot sait toujours où il a trouvé la dernière pièce de la pile demandée. Si on utilise plus de deux containers, il est alors possible d'avoir une diversité de container, c'est à dire que le robot mémorise la position des dernières pièces de chaque pile de chaque container.

La recherche systématique de la première pièce de chaque pile n'est effectuée que sur Spécial 3, lancement en production après retour au Repli, changement de container ou demande de RESET par l'automate (voir ci-dessus) pour les cas d'AZ Double et seulement par demande de RESET par l'automate pour les AZ Simples.

Dans le cas du capteur de détection de pile sur le préhenseur, lors de la prise de la pièce, le robot vérifie si cette pièce est la dernière en comparant sa position spatiale au point de fin de pile, dans la limite de la tolérance **TolFinPil** (modifiable). Dès que le robot est dans cette tolérance, il vérifie qu'il n'y a plus de pièce dans la pile avant d'aller à la suivante. Si le robot laisse malgré tout des pièces, c'est qu'il faut réapprendre le point de fin de pile plus profond.

Si aucune pièce n'est trouvée avant d'atteindre le point de fin de pile, le robot rejoint le point de début de pile à vitesse réduite et affiche un message d'erreur:

"PILE VIDE !!! (Vérifiez Capteur Palette)".

Il faut acquitter le défaut, puis le programme est totalement ré-initialisé. Il s'agit en effet d'un cas de panne du détecteur de pile, trompant ainsi l'automate.

La gestion des reprises de trajectoire est supportée. (Reprise avant, après...). La dépalettisation est en effet considérée comme une trajectoire de prise pièce ordinaire.

**Nota :** Pour la dépalettisation avec capteur embarqué, le robot gère lui-même les piles, c'est-à-dire que l'utilisateur n'indique que le n° du container dans lequel il va travailler. Le robot détecte alors quelle pile est pleine avec son capteur longue-portée. Il utilise ce capteur lorsque l'on change de pile ou quand on s'approche de la fin d'une pile. Il vérifie ainsi qu'aucune pièce ne reste dans une pile de pièces (qu'elles soient collées les unes aux autres ou non). Ainsi, le robot se présentera devant la pile qu'il vient de vider pour vérifier qu'aucune pièce ne reste, et si cette pile est effectivement vide, il passera à la pile suivante. Pour ce système de vérification, il faut que les points de la trajectoire ne soient pas appris à plus de 3 mm plus bas que le point de prise de la dernière pièce.

Cette méthode apporte les avantages suivants:

- Grande souplesse puisqu'une programmation très simple permet d'obtenir de nombreux modes de fonctionnement différents: prendre X pièces successivement de la pile 1, puis Y de la pile 2, puis revenir sur la 1, etc...Le trajectoriste n'a rien à faire: tout se fait "en temps réel" en fonction du besoin du moment, selon le code automate ou, pourquoi pas, un bouton de demande opérateur.
- Les pièces de chaque pile peuvent être de type différent et l'on peut même utiliser une main de préhension différente pour chaque type.
- Les piles peuvent contenir un nombre quelconque de pièces. Elles n'ont pas besoin d'être pleines au départ.
- On peut enlever des pièces d'une pile à tout moment sans inconvénient.
- On peut rajouter des pièces dans une pile sous réserve de ramener le robot au Repli ou d'utiliser la raz automatique.
- Il peut y avoir des "trous" entre les pièces.
- La distance entre chaque pièce n'a aucune importance et peut même varier à l'intérieur d'une pile.

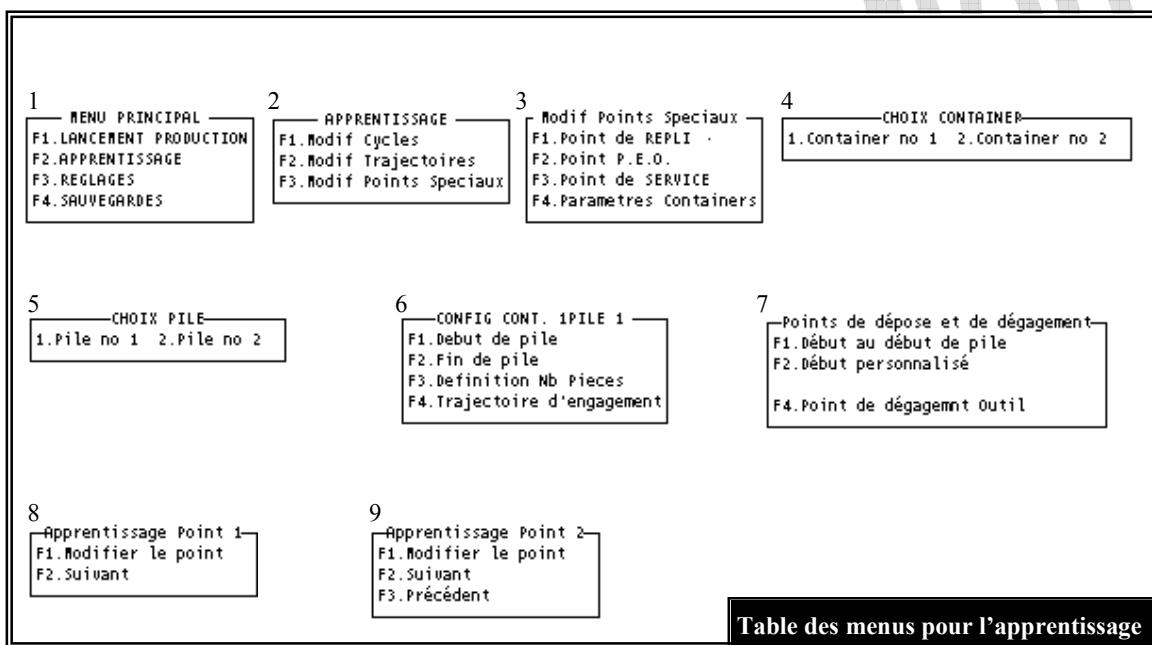
## **13 -Palettisation:**

Le module de Palettisation peut gérer jusqu'à 10 containers de 10 piles chacun. Ceci est défini lors de l'installation de l'application.

### a) Création des trajectoires

Pour apprendre une trajectoire sur un container, il faut renseigner les points de début et de fin de pile de chaque pile, de chaque container. Le point de début et de fin de pile sont appris exactement à leur emplacement, contrairement à la dépalettisation.

Pour la création des points, il faut suivre les menus « Apprentissage », « Points spéciaux », « Paramètres containers », « Choix du container », « Choix de la pile ».



Ainsi pour chaque pile, il sera demandé de renseigner :

- 1) le point de début de pile,
- 2) le point de fin de pile,
- 3) le nombre de pièces sur cette pile
- 4) les trajectoires de pose (engagement de la pièce et dégagement de l'outil)

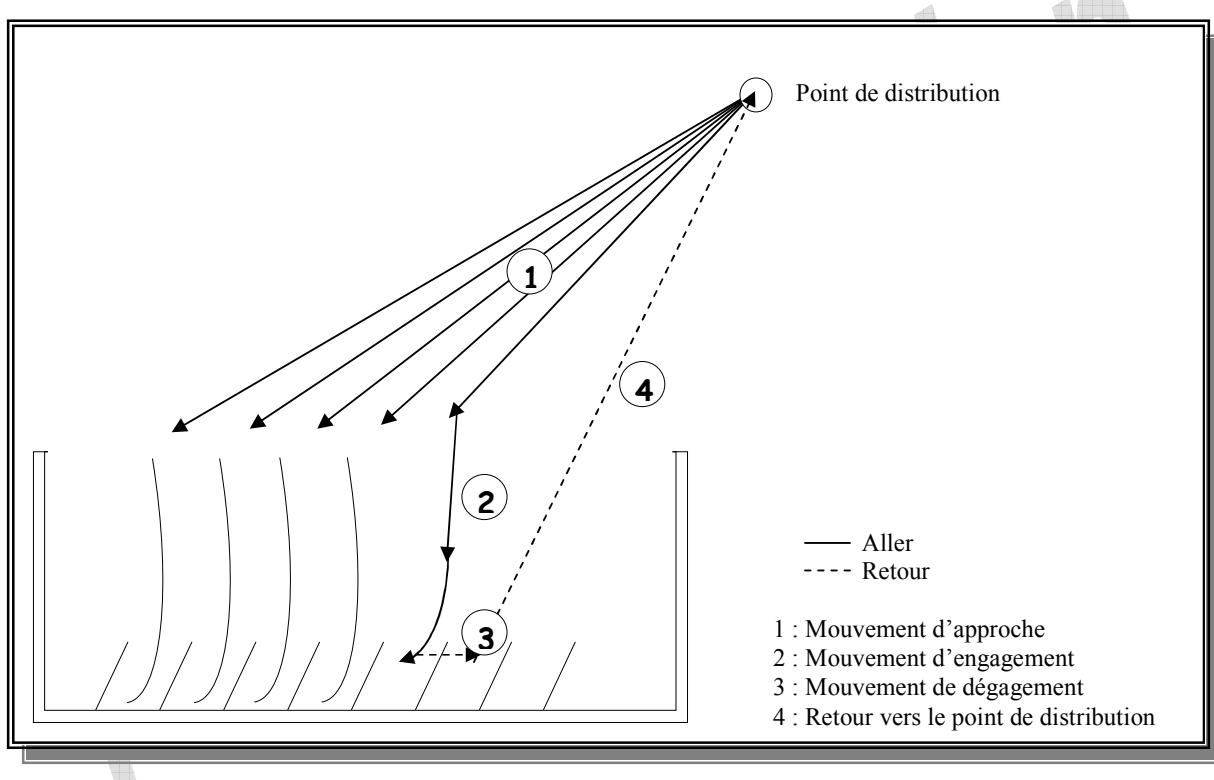
Les points de dégagement de la pièce doivent s'apprendre par rapport au point de début de pile.

Une fois le robot au point de pose, le menu 8 s'affiche avec, en titre, le numéro du point sur lequel on travaille. Il est alors possible d'apprendre ce point en appuyant sur « **F1. Modifier le point** ». L'application vous propose de bouger le robot à l'emplacement désiré. Si vous avez déjà appris le point, vous pouvez passer directement au suivant en faisant « **F2.Suivant** ». Si une erreur est commise, il est possible de faire « **F3.précédent** » et le robot revient au point précédent. Pour chaque mouvement, si des points ont été préalablement appris, l'application demande une confirmation.

Pour information

## NOTAS :

- Par l'apprentissage, l'application calcule des vecteurs. C'est pour cela qu'il est extrêmement important que le robot soit au point précédent avant de faire « Modifier le point ».
- Lors de l'apprentissage des points, il est impératif d'utiliser un repère outil centré par rapport à la pièce ou au préhenseur. Si cette condition n'est pas respectée, le robot risque de mal déposer les pièces
- Il est important d'apprendre le point de distribution de pièce avec une très petite zone (risque de choc). Ce point doit être accessible de tous les points de début de pile et des points de dégagement. (voir schéma suivant)



Dans la trajectoire de palettisation, insérez l'instruction "Palettise", disponible dans la liste commune 2.

Il faut alors préciser:

- Le n° de container concerné (1 à 10).
- Le n° de pile dans ce container; (si l'option « capteur sur container » a été choisie)
- Le n° de main à piloter pour la fermeture du préhenseur (*Ferme\_Main X*).
- Le nom des PP à surveiller (jusqu'à 4)
- La vitesse d'approche au premier point de la pile. (Généralement Vmax)
- La vitesse de pose de la pièce. (V100)
- Le repère outil de la trajectoire de pal
- Le repère objet de la trajectoire de pal

## b) Communication avec l'automate

L'automate choisit le container et la pile dans lequel le robot doit poser la pièce au moyen d'un code cycle par pile.

- Voici l'adressage des codes cycles à respecter :

- de 001 à 145 : LIBRES
- de 146 à 245 : Demande de travail sur pile sans RESET
- de 246 à 255 : Demande de RESET + travail sur pile N°1 du container razé

Exemples :

146 sera toujours la pile 1 du container 1 (existe forcément)

147 sera toujours la pile 2 du container 1 (si elle existe)

148 sera toujours la pile 3 du container 1 (si elle existe)

149 sera toujours la pile 4 du container 1 (si elle existe)

150 sera toujours la pile 5 du container 1 (si elle existe)

151 sera toujours la pile 6 du container 1 (si elle existe)

152 sera toujours la pile 7 du container 1 (si elle existe)

153 sera toujours la pile 8 du container 1 (si elle existe)

154 sera toujours la pile 9 du container 1 (si elle existe)

155 sera toujours la pile 10 du container 1 (si elle existe)

156 sera toujours la pile 1 du container 2 (si elle existe)

157 sera toujours la pile 2 du container 2 (si elle existe)

158 sera toujours la pile 3 du container 2 (si elle existe)

159 sera toujours la pile 4 du container 2 (si elle existe)

160 sera toujours la pile 5 du container 2 (si elle existe)

161 sera toujours la pile 6 du container 2 (si elle existe)

162 sera toujours la pile 7 du container 2 (si elle existe)

163 sera toujours la pile 8 du container 2 (si elle existe)

164 sera toujours la pile 9 du container 2 (si elle existe)

165 sera toujours la pile 10 du container 2 (si elle existe)

236 sera toujours la pile 1 du container 10 (si elle existe)

245 sera toujours la pile 10 du container 10 (si elle existe)

246 sera toujours la demande de RESET compteur du container 1 avec demande de travail  
sur la pile 1 du container 1

247 Idem Container 2

248 Idem Container 3

249 Idem Container 4

250 Idem Container 5

251 Idem Container 6

252 Idem Container 7

253 Idem Container 8

254 Idem Container 9

255 sera toujours la demande de RESET compteur du container 10 avec demande de travail  
sur la pile 1 du container 10

C'est à l'automate de gérer les BI ainsi que les demande de RESET (demandées en début de cycle également).

**NOTA:**

En cas de P-Start ou de rechargeement manuel du fichier PALBASE.sys, le Standard considère que les containers sont pleins, par sécurité. En effet, si l'on fait un P-Start alors qu'un container était en cours, le robot ne peut plus savoir quel est le véritable état du container (vide, plein ou entre les deux...).

Pour éviter d'écraser la pile de pièces éventuellement présente, le Standard force une demande de RAZ container dans ce cas précis.

Pour information

## X - HISTORIQUE DES DEFAUTS

### AVERTISSEMENT:

Cette fonction n'est qu'une simple consignation de certains événements, horodatés. Elle est destinée à donner une indication générale sur le nombre de défauts survenant en cours d'exploitation. Elle ne **permet pas** de calculs de disponibilité, de rendement ou autre.

ABB peut toutefois fournir, sur demande, une application dénommée "Journal de Bord", qui permet de calculer (sur le robot) tous les paramètres souhaités, notamment Disponibilité Propre, Disponibilité Opérationnelle, Temps d'Arrêt Propres, Temps d'Arrêt Induits, Temps de cycle moyen par type de pièce, etc... Les fonctions proposées ici ne peuvent en aucun cas rivaliser avec celles de cette application spécifique, qui doit être paramétrée finement selon la demande.

Tous les événements "graves" affectant la vie du robot sont consignés dans un fichier de suivi. Ce fichier porte le nom du site qui a été donné lors de l'installation du standard, suivi de l'extension ".xls". Ex: "R\_023154.xls". Cette extension indique simplement que le fichier est compatible avec le format EXCEL. Si ce tableau est utilisé, les données apparaîtront proprement rangées à l'écran, par colonne. Mais n'importe quel éditeur de texte sur PC permet de lire ce fichier, en perdant éventuellement la mise en page.

Les événements "graves" sont:

- 1 - Tous les défauts de l'application considérés comme bloquants avec validation.  
Ex: "Défaut Présence Préhenseur".
- 2 - Certains événements qui, bien que non bloquants, permettent de se repérer et de suivre un peu l'exploitation du site. (Ex: "Lancement Production", "Arrêt Production" ...).

Les défauts « système » ne peuvent être interprétés par l'application. Ils sont tout de même détectés et provoquent l'allumage de la verrine rouge et la montée de la sortie *do\_Err\_Sys* si le programme était en cours d'exécution. De plus, ils sont archivés par catégorie, et visibles via le menu Maintenance.

Pour pouvoir lire ce fichier, il est nécessaire de le copier sur une disquette. Pour cela, validez, Hors Production, le menu "**Sauvegardes**", "**Sauver le fichier de Suivi**". Insérez la disquette du robot dans le lecteur, ou toute autre disquette. Si un réseau Ethernet est installé, la sauvegarde peut se faire directement sur le PC distant. Validez. Le fichier est copié sur la disquette, il porte l'extension ".xls".

Le fichier est géré comme une pile FIFO. Les événements les plus récents sont en début de fichier, les plus anciens à la fin. Lorsque le fichier atteint la taille maximum, les événements les plus anciens sont automatiquement effacés.

La taille maximum du fichier peut être modifiée en changeant la valeur de NBLIGSUIVI (100 par défaut).

Il est vivement conseillé de ne pas exagérément dépasser cette valeur pour deux raisons: le fait de gérer le fichier comme une pile oblige à le recopier entièrement dans un tampon, pour pouvoir ré-écrire la première ligne, puis recopier le tampon (moins 1 ligne) à la suite. Cette opération prends donc un certain temps, qui se voit parfaitement lors d'un défaut.

De plus, la mémoire disponible en Ramdisk doit être d'au moins 2 fois la taille déclarée.

Pour information

## XI - FONCTIONNEMENT SOUS DESKWARE

Le programme est prévu pour un fonctionnement transparent sur PC à l'aide du logiciel de simulation de baie DESKWARE. L'ensemble des écrans de cette documentation ont d'ailleurs été créés ainsi.

Pour installer le programme, procédez comme suit:

- 1 - Créez un type de robot comprenant les options indispensables (Multitaches, Fonctions avancées...).
- 2 - Ouvrez ce type de robot.
- 3 - Insérez la disquette "STANDARD PSA " dans le lecteur **a**.
- 4 - Une fois le robot virtuel initialisé, validez "Fichier", "Ouvrir", "Installe" à partir du lecteur de disquettes.
- 5 - Validez la touche "EXEC" pour lancer l'exécution du programme.

Le programme d'installation détecte automatiquement qu'il s'agit d'un fonctionnement sur PC et non sur robot réel. Il génère donc des fichiers de configuration spécifiques. Toutefois, ceci est invisible pour l'utilisateur, qui voit exactement le même fonctionnement que sur un robot réel.

Tous les fichiers sont générés dans le répertoire du **robot courant**, ce qui permet d'en avoir éventuellement plusieurs avec des configs différentes (Ex: c:\program files\ABBDesware\Controls\V31\Robots\IRB6400).

A l'invite du message "Chargez les paramètres système...", chargez les fichiers A1, A2 et A3 présents non pas en Ramdisk, mais sur C: Puis faites le P-START, exactement comme sur une baie réelle.

Le programme charge alors les fichiers propres au robot à partir de C:, puis les modules standard à partir de la disquette. En fait, lors d'une utilisation sur PC, le répertoire RAMDISK est simplement remplacé par ".\", et le support "fpl1:" est remplacé par "a:".

Lors de redémarrages de DeskWare suivants, la présence de la disquette du standard n'est plus utile. En effet, Desware mémorise les fichiers dans un répertoire tampon, et sait les recharger à partir de là lors des démarrages ultérieurs.

A l'instar du robot réel, la disquette standard n'est nécessaire que lors de la première installation, d'un changement de configuration, ou d'un P-START.

NOTA: Le fonctionnement de Desware n'est pas forcément strictement identique à une baie S4 C réelle. Notamment, la gestion de la longueur des chaines de caractères est différente (mode UNIX / mode DOS-WINDOWS). Ne vous étonnez donc pas si certains affichages ne sont pas strictement conformes à ceux observés sur la baie, ou si un trait vertical inattendu apparaît à l'écran.

L'horloge du système, le temps d'exécution et l'environnement étant également très différents entre un PC et une baie S4, certains phénomènes inattendus peuvent apparaître sur Deskware. Sachez qu'ils n'existent pas sur une baie réelle, et qu'il ne s'agit pas de bogues.

De plus, certaines fonctions sont implémentées sur le robot réel, mais pas sous Deskware, qui subit toujours un certain retard de développement par rapport à RobotWare.

Le Standard PSA tente de pallier à ce défaut en installant lui-même les routines pouvant manquer dans Deskware, et en simulant plus ou moins le fonctionnement réel.

Vous pouvez parfaitement créer, simuler ou modifier des trajectoires à partir de Deskware, et les recharger ensuite sur le robot réel. ("Fichier", "Ouvrir", "Traject").

Toutefois, **ne pas utiliser** les fichiers de paramètres système, car ils sont différents entre robot réel et robot virtuel.

Il est également possible de simuler le fonctionnement en réseau Ethernet, en créant un répertoire "c:\S4boot\AppliXX". Voir le chapitre "Sauvegardes" pour plus de détails.

NFS n'a dans ce cas, pas besoin d'être installé sur le PC.

## XII - LISTE DES ROUTINES ACCESSIBLES AU PROGRAMMEUR

Les routines disponibles du standard de base sont regroupées dans la liste commune MC 1, accessible par la touche en haut à droite lorsqu'on est en apprentissage.

Les listes MC 2 et MC 3 comprennent les actions propres à chaque process.

Si ces listes sont vides, c'est que la baie n'a jamais été coupée depuis l'installation du standard. En effet, ces listes ne sont remplies par le système que si les fonctions déclarées sont présentes en mémoire, ce qui n'est pas le cas lors du P-START initial. Faites donc un redémarrage, ou coupez la baie: les listes apparaissent alors.

ROUTINE	ACTION	PARAMETRES OPTIONNELS
<i>AffAlerte</i>	Ecrit la chaîne spécifiée dans le champ « Alerte » de l'écran de production. Affiche le message dans un cadre centré à l'écran si l'on est Hors Production.	Aucun
<i>AffDefProcess</i>	Allume la verrine rouge, prends la main sur l'écran de production si besoin, affiche le message spécifié dans un cadre en trait double centré à l'écran. Arrête le robot. Ecrit le message dans le fichier de suivi avec la date et l'heure. Interdit le départ distant. Il est nécessaire d'acquitter le message, puis le programme s'arrête et les asservissements retombent. Il est nécessaire de faire DEPART pour le relancer. Les interruptions sont ré-activées, la main est éventuellement rendue à l'écran de production.	\Entete (string) Le titre de la fenêtre.  \NoStop Provoque l'arrêt du robot, mais sans arrêt du programme, qui continue dès la validation de l'opérateur. Le message n'est pas écrit dans le fichier de suivi.
<i>AffDefTab</i>	Même chose que ci-dessus, mais pour un tableau de chaînes de caractères.	Idem
<i>AffMessage</i>	Ecrit la chaîne spécifiée dans le champ « EtatRobot » de l'écran de production si on est En Production, affiche le texte dans un cadre centré sinon.	Aucun
<i>AffPosition</i>	Ecrit la chaîne spécifiée dans le champ « Position du Robot » de l'écran de production.	Aucun
<i>AffMesStop</i>	Arrête le robot, provoque un arrêt du programme et une coupure des asservissements. Affiche le message spécifié, qui reste visible tant que le robot	Aucun

	est en auto. Il n'y a pas à acquitter le message pour repartir. Attention : bien que les moteurs soient coupés, la sortie <b>RobotOk reste à 1</b> , pour que l'automate n'interprète pas cet arrêt comme un défaut robot. Cette routine est utilisée notamment lors du changement électrodes.	
<i>FUNC Affmenu</i>	Prends éventuellement la main sur l'écran de production, puis affiche le tableau de chaînes spécifié sous forme d'un menu comportant autant de choix possibles que de lignes non vides dans le tableau (limité à 4). Formate les chaînes selon la plus longue, ajoute le numéro de choix. (Fx). Ajoute le choix « QUITTER ». Attends un choix de l'opérateur et renvoie le numéro choisi, sauf si on fait « Quitter ». Dans ce cas la fonction renvoie false. L'affichage est régénéré en cas de lâcher gâchette et réappui, ou passages auto/manu. Pour programmeurs avertis.	\Noquit : Interdit de quitter un menu, un choix est obligatoire. \Entete := » ? » Titre du menu.
<i>FUNC ChoisiDansListe</i>	Prends éventuellement la main sur l'écran de production, puis affiche le tableau de chaines spécifié sous forme d'une liste numérotée, dans un cadre centré à l'écran. Le nombre de lignes et de colonnes s'adapte en fonction de la chaine la plus longue. Gère une file déroulante si le nombre de données ne tient pas sur une seule page écran. Attends un choix de l'opérateur, renvoie le numéro choisi sauf si on fait « Quitter ». Dans ce cas la fonction renvoie false. Pour programmeurs avertis.	\A_Choisir Affiche la liste, mais sans choix possible (lecture seule).
<i>AppelleTraj</i>	Lance l'exécution de la trajectoire spécifiée et gère la zone donnée si elle est non nulle. Gère les bifurcations de trajectoires ainsi que les reprises et les autorisations de zones. Attention: la libération de la zone est imprécise par rapport au dernier point de la traj (anticipation).	\ou_traj \si_zone permettent de gérer les bifurcations de trajectoires.
<i>Attends_Entrees</i>	Teste si une entrée est dans l'état spécifié. Si oui, la fonction rends la main immédiatement. Sinon, un message apparaît : « ATTENTE... ». Au bout du temps spécifié, un défaut avec acquittement est généré si le robot est en mode automatique. Sinon, un message avec choix permet de reprendre l'attente ou de sauter l'événement attendu.	\di_sig2 \etat_att2 Permettent d'attendre simultanément une deuxième entrée. \Abandon booléen renvoyé par la routine indiquant que l'attente a été abandonnée en manuel.

<i>Attends_PP</i>	Si on est en mode sans pièce, vérifie que le présence pièce spécifié est à 0. Si on est avec pièce, vérifie que celui ci est dans l'état spécifié. A appeler dans une trajectoire. Le(s) presence(s) piece(s) est ensuite surveillé en permanence jusqu'à un appel à Attends_PP avec présence pièce à 0 ou à -1.	<b>Presence_piece2,</b> <b>Presence_piece3</b>
<i>Att_Acces_Zone</i>	Attends de l'automate le numéro d'autorisation de zone spécifié, puis active la surveillance de cette zone.	<b>Aucun</b>
<i>Entree_Interver</i>	Gère l'entrée du robot dans une zone d'interverrouillage, éventuellement sur point de passage. Si l'entrée interverrouillage est absente, le point est dégénéré en point d'arrêt. 16 zones sont possibles.	\Maitre Doit être positionné sur le robot maître s'il est possible que les deux robots se présentent exactement en même temps dans la zone.
<i>Fin_de_Cycle</i>	Permet de monter la fin de travail sur point de passage et d'anticiper la lecture code.	<b>Aucun</b>
<i>Fin_de_Traj</i>	Permet de monter le hors zone de la traj sur point de passage sans anticipation.	<b>Aucun</b>
<i>Fin_de_Zone</i>	Permet de monter le dernier hors zone consommé de façon précise sur point de passage, dans le cas de zones imbriquées.	<b>Aucun</b>
<i>MoveJSync,</i> <i>MoveLSync,</i> <i>MoveCSync</i>	Permettent d'exécuter une <b>routine</b> sur un point de passage. La routine spécifiée est exécutée sur la projection de la trajectoire au centre de la sphère définie sur le point. La routine spécifiée peut par exemple monter des sorties. <b>Elle ne doit jamais attendre d'événements.</b> Attention : le traitement du mode pas à pas doit être écrit par le programmeur. Pour programmeurs avertis.	
<i>Hors_Zone</i>	Monte le hors zone spécifié en envoyant éventuellement un front à l'automate. <b>(Ne devrait plus être utilisée).</b>	<b>Aucun</b>
<i>Sortie_Interver</i>	Libère la zone d'interverrouillage spécifiée. Peut se faire sur point de passage.	\L Effectue un mouvement linéaire au lieu d'articulaire.
<i>Subsitution</i>	Exécute la trajectoire de substitution correspondant au type défini en mode Réglage et gère la zone donnée si elle est non nulle.	<b>Aucun</b>
<i>Trace</i>	Ecrit le message spécifié dans le fichier de suivi, en ajoutant la date et l'heure.	<b>Aucun</b>
<i>Verif_Came</i>	Permet de vérifier, dans un cycle ou dans une trajectoire, le changement d'état des cames de sécurité. Les valeurs Hors_Came, Sur_Came, Hors_Came_Sim et	<b>Aucun</b>

	Sur_Came_Sim sont définies.	
<i>FUNC bool Rep_Av_Prise()</i>	<p>Teste si les présence pièce spécifiés sont à 1. S'ils sont tous à 0, la fonction renvoie VRAI si l'opérateur a validé. S'ils sont tous à 1, la fonction renvoie FALSE. Si l'un des présence pièces est à 1, propose le choix « Reprendre après prise pièce ? ». Si l'on réponds OUI, <b>tous</b> les présences pièces doivent être à 1. Si ce n'est pas le cas, la routine reste bloquée jusqu'à ce que ce soit le cas, puis renvoie FALSE.</p> <p>Si l'on réponds NON, la routine attend que tous les présences pièces soient à 0, et renvoie VRAI.</p> <p>Cette fonction doit être utilisée pour gérer les reprises des trajectoires de prise pièce.</p>	<b>Prespiece1, Prespiece2, Prespiece3.</b>
<i>FUNC bool Rep_Av_Depose()</i>	Même principe que ci-dessus, mais pour une dépose pièce. Le fonctionnement est parfaitement symétrique.	<b>Prespiece1\Prespiece2\Prespiece3.</b>
<i>FUNC bool Rep_Av_Process</i>	Même principe que ci-dessus, sauf que l'exécution de la reprise ne peut être faite que si tous les présence pièce spécifiés sont à 1 (pas de gestion automatique du mode Reprise_Apres...).	<b>Idem.</b>

### XIII - LISTE DES MESSAGES D'ERREUR

Tous ces messages sont bloquants avec acquittement obligatoire et consignation dans le fichier de suivi.

<b>MESSAGE</b>	<b>RAISON / ACTION</b>
Paramètre manquant pour Appelletraj	Vous avez écrit une bifurcation de trajectoire sans avoir spécifié la zone. Corrigez la programmation.
2 zones programmées identiques	Vous avez programmé une bifurcation de trajectoire avec deux zones identiques. Corrigez la programmation.
Trajectoire XX Inexistante !	Vous avez fait appel à une trajectoire qui n'existe pas. Corrigez le numéro dans AppelleTraj, ou créez une nouvelle trajectoire par "Dupliquer".
Trajectoire T_Subst Inexistante !	Vous avez fait appel à une trajectoire de substitution qui n'existe pas. Corrigez la programmation ou créez la trajectoire.
Routine T_Prise_Out non trouvée.	Une des trajectoires du changeur d'outil a été supprimée. Le changeur a besoin des routines suivantes: T_Prise_Out1, T_Prise_Out2, T_Prise_Out3, T_Depose_Out1, T_Depose_Out2 et T_Depose_Out3. Elles peuvent être vides, mais doivent exister.
Traj de service inconnue : XX	L'arrière plan a demandé à l'avant plan d'exécuter une trajectoire de service inconnue. Relevez la valeur de <i>sup_move_along</i> et contactez ABB.
Cycle Inconnu : XX	L'automate a envoyé un numéro de code cycle pour lequel aucun cycle n'existe. Créez le s'il n'existe pas, ou corrigez l'automate.
Cycle XX vide !	Le cycle XX ne contient aucune trajectoire, ou ne contient qu'un traitement dont la durée d'exécution est inférieure à 0.5 seconde.
Interruption inconnue	Bugg dans le programme: contactez ABB.
N° de zone invalide	Idem.
N° de zone inconnu	Idem.
Erreur Automate: 2 zones simultanées!	L'automate a envoyé simultanément les deux autorisations de zones pour une bifurcation de trajectoire, ce qui est interdit. Corrigez le programme automate.
Sortie_intervenir interdit dans un cycle !	La routine <i>sortie_intervenir</i> ne peut être appelée que depuis une trajectoire car elle contient un mouvement. Corrigez la programmation.

N° d'interverrouillage incorrect	Vous avez fait appel à une routine d'interverrouillage avec une valeur < 1 ou >2. Corrigez la programmation.
Defaut capteur came X attendu à Y.	L'état de la came ne corresponds pas à l'état attendu dans le programme. Réglez les cames ou corrigez la programmation.
Fin_de_Cycle interdit dans un cycle !	Cette routine ne peut être appelée que depuis une trajectoire. Corrigez la programmation.
Fin_de_Traj interdit dans un cycle !	Idem.
Erreur disquette !	La disquette est protégée en écriture ou n'est pas bien mise dans le lecteur. Vérifiez et validez le choix "Refaire".
Defaut Parite Code API	Le calcul de la parité ne corresponds pas à l'état du bit reçu de l'automate. Vérifiez le programme automate, le câblage, et relancez.
Incohérence verif cames sur traj !	Le changement d'état des cames de sécurité ne s'est pas fait au cours du cycle. Vérifiez les cames, le câblage, la programmation.
Etape inconnue	Erreur interne. Contactez ABB.
Fichier xxx non trouvé !	L'un des modules nécessaires a été effacé ou non chargé. Mettez la disquette STANDARD PSA T5 dans le lecteur et validez.
Outil XX inconnu ! (Changeur)	Le nom de l'outil à prendre ne corresponds pas au nom de l'un des trois outils possibles. Vérifiez l'écriture de l'appel à Chge_Outil. Les majuscules et minuscules doivent être respectées.
Pas d'outil détecté ! (Changeur)	Les capteurs de codage de numéro d'outil ne fonctionnent pas. (di_cod_out1 et di_cod_out2).
Mauvais outil détecté ! (Changeur)	L'état des entrées de codage outil ne corresponds pas au numéro d'outil que l'on vient de prendre. Vérifiez les entrées ou modifiez la programmation.
Outil X inactif ! (Changeur)	Vous avez voulu aller chercher un outil qui est déclaré en panne. Modifiez le flag <i>outil_actif{X}</i> .
Type d'outil inconnu ! (Changeur)	Vous n'avez pas déclaré le type de cet outil. Faites le par le menu "Changeur", "Noms des Outils", du menu Apprentissage.
MoveClinche interdit dans un cycle !	Cette routine doit être appelée à partir d'une trajectoire. Corrigez la programmation.
Pince non ouverte (Clinchage)	Le contrôle d'ouverture pince à clincher est à 0.
Defaut Fin de cycle (Clinchage)	Après avoir attendu 2 secondes, le fin de cycle est toujours à 1 avant le clinchage.
Defaut Recul Det (Clinchage)	Après avoir attendu 2 secondes, le contrôle de recul poinçon n'est toujours pas à 1 avant de fermer la pince.

Defaut Fin de Cycle (Clinchage)	5 secondes après le début du clinchage, le fin de cycle est toujours à 0.
Defaut recul Det (Clinchage)	5 secondes après la demande d'ouverture du poinçon, le contrôle est toujours à 0.
Defaut Presence Clincheur (Clinchage sur changeur)	L'entrée di_Pres_Prehen est à 0.
Entrée Comptage à 0 (Debit)	L'entrée Comptage en cours est passée à 0 pendant l'encollage.
Entrée Comptage restée à 1 (Debit)	L'entrée Comptage en cours est restée à 1 300 ms après avoir envoyé l'arrêt comptage.
Defaut dose trop faible (Debit)	Le volucompteur a mesuré une dose trop faible.
Defaut dose trop forte (Debit)	Le volucompteur a mesuré une dose trop forte.
Defaut volucompteur (Debit)	Le volucompteur ne marche pas. L'entrée di_DefDebit est passée à 0 pendant l'encollage.
Defaut dose min à 0 (Debit)	L'entrée di_DefDoseMin est à 0 avant le début du premier cordon.
Defaut dose max à 0 (Debit)	Idem pour l'entrée di_DefDoseMax.
Defaut débitmètre à 0 (Debit)	Idem pour l'entrée di_DefDebit.
Defaut comptage déjà à 1 (Debit)	L'entrée comptage est déjà à 1 avant de commencer le premier cordon.
Alarme groupe Debit	L'entrée di_AlarmeEPO est à 0.
Defaut bloquant groupe Debit	L'entrée di_BloqEpo est à 0.
DebitL interdit dans un cycle ou DebitC interdit dans un cycle	Cette routine doit être appelée depuis une trajectoire. Corrigez la programmation.
Arg DernierCordon interdit sur P1On.	Corrigez la programmation de DebitL.
Arg PremierCordon interdit sur P1Off	Idem.
Defaut Presence Pistolet Debit	L'entrée di_pres_pistoEPO est à 0. Corrigez ou inhibez la surveillance par le menu Réglages.
Attention: mode sans colle !	Le commutateur est sur la position sans colle alors qu'on est avec pièce lors du lancement en production.
n° de pisto inconnu (Mastic)	Vous avez fait un appel à <i>Purge_Pisto</i> avec un numéro de pistolet invalide. Corrigez la programmation.
Def pression min Cordon X (Mastic)	Un défaut pression min est survenu pendant l'encollage du cordon X.
Def pression Max Cordon X (Mastic)	Idem pour pression max.
Alarme groupe Encollage	L'entrée di_Grp_Alarme est à 0.
Defaut bloquant groupe Encollage	L'entrée di_Grp_Bloq est à 0.
Manque Option P1 ou P2 pour...	Erreur de programmation dans DefautsPisto ou dans CtlAvantPisto. Corrigez votre instruction en validant le paramètre optionnel.

PistoL interdit dans un cycle	Cette routine doit être appelée depuis une trajectoire. Corrigez la programmation.
Defaut Presence Pistolet (Mastic)	Vérifiez les entrées di_pres_pisto. Inhibez la surveillance par le menu Réglages.
Attention : mode sans colle (Mastic)	Le commutateur est sur la position sans colle alors qu'on est avec pièce lors du lancement en production.
Gene Bloque (Goujons)	L'entrée di_gene_ok est à 0 avant soudure goujon.
Def avant soude (Goujons)	L'entrée di_tete_ok est à 0 avant soudure.
FCY avant soude (Goujons)	L'entrée di_Tete_FCY est à 1 avant la soudure.
Tete 2 non déclarée (Goujons)	Vous avez voulu souder avec la tête 2 alors qu'elle n'est pas déclarée. Corrigez la programmation ou modifiez le flag tu_tete2_existe.
SoudeTeteL interdit dans un cycle	Cette routine doit être appelée depuis une trajectoire. Corrigez la programmation.
Manque param tete1 ou tete 2	Vous avez fait appel à SoudeTeteL sans préciser le numéro de tête. Corrigez la programmation.
Numéro de tête invalide (Goujons)	Vous avez spécifié un numéro de tête différent de 1 ou 2 pour l'appel à SoudeTeteL. Corrigez la programmation.
Defaut Presence Tete X (Goujons)	L'entrée di_TeteX_Ok est à 0. Corrigez ou inhibez la surveillance par le menu Réglages.
Attention: mode sans soudage (Goujons)	Le commutateur est en position sans soudure alors qu'on est avec pièce lors du lancement en production.

## XIV - LISTE DES MODULES

### **1 - Liste des modules utilisés uniquement par l'avant-plan:**

BASE.sys (module du système, non présent sur la disquette)  
USER.sys (Idem)

PSA.prg  
SUPTRA.sys  
MESSTRA.sys  
STD AUTO.sys  
TRAJECT.sys  
AVPFRAN (ou AVPANG ou AVPESP) (Textes du noyau en Français, Anglais, Espagnol).

On trouve ensuite les modules qui ne sont installés que si le process correspondant est déclaré:

PROCESS	NOM DU MODULE
Encollage Debit	ECEPBASE.sys ECFRAN.sys (ou ECANG ou ECESP)
Encollage Mastic	ECREBASE.sys
Encollage Doseur ou Pulvérisation	ECDOBASE.SYS
Encollage double Doseur type A8	ECD2BASE.sys
Clinchage	CLINCH.sys CLINFRAN.sys
Soudure Goujons	TUCKBASE.sys TUCKFRAN.sys
Changeur d'outils	CHANGEUR.sys CHGFRAN.sys
Soudure 1 pince pneumatique	SR1PBASE.sys SAVPFRAN.sys
Soudure 2 pinces pneumatiques	SR2PBASE.sys SR2PMOVE.sys SAVPFRAN.sys
Soudure Arc	ARCBASE.sys ARCFRAN.sys
Rivetage Electrique	RIVBASE.sys RIVFRAN.sys
Changeur d'Electrodes	CHGELEC.sys
Soudure Electrique 1 pince (swit)	SRELBASE.sys SAVPFRAN.sys
Soudure Electrique avec séquenceur externe	SAROBASE.sys SAVPFRAN.sys
Soudure double carte Swit sur CAN	SRE2BASE.sys SAVPFRAN.sys

Soudure double swit + double seq.externe	SAR2BASE.sys SAVPFRAN.sys
Depalettisation version 1 (capteur fixe)	DEPBASE.sys PALFRAN.sys
Depalettisation version 2 (capteur embarqué)	DEP2BASE.sys PALFRAN.sys
Palettisation	PALBASE.sys PALFRAN.sys

Ces modules contiennent les actions que l'on peut appeler sur trajectoire. (Ex: SoudeTeteL).

*Pour information*

## **2 - Liste des modules partagés entre l'avant et l'arrière plan:**

ACTUSITE.sys  
CONFIGUR.sys  
SUPGLOB.sys

Puis les modules dédiés process:

PROCESS	NOM DU MODULE
Encollage Debit	ECEPACTU.sys
Encollage Mastic	ECREACTU.sys
Encollage Doseur ou Pulvérisation	ECDOACTU.sys
Encollage Double Doseur type A8	ECD2ACTU.sys
Clinchage	CLINACTU.sys
Soudure Goujons	TUCKACTU.sys
Soudure 1 pince	SR1PACTU.sys
Soudure 2 pinces	SR2PACTU.sys
Soudure ARC	ARCACTU.sys
Rivetage Electrique	RIVACTU.sys
Soudure Electrique	SRELACTU.sys SCOMFRAN.sys
Soudure Electrique et séquenceur externe	SAROACTU.sys SCOMFRAN.sys
Soudure Double carte Swit	SRE2ACTU.sys SCOMFRAN.sys
Soudure Double Swit + Double seq.externe	SAR2ACTU.sys SCOMFRAN.sys

Ces modules contiennent les actions process que l'on peut appeler soit dans les actions manuelles (donc par l'arrière plan), soit dans une trajectoire. (Ex: Man\_Clinche).

### **3 - Liste des modules utilisés uniquement par l'arrière-plan:**

SUPROB.prg  
ACTMBASE.sys  
ACTMSITE.sys  
ROBAUTO.sys  
SUPGLOFR.sys (ou SUPGLOAN ou SUPGLOES): Textes du noyau ARP.

Puis les modules dédiés aux différents process:

PROCESS	NOM DU MODULE
Encollage Debit	ECEPSURV.sys
Encollage Mastic	ECRESURV.sys
Encollage Doseur ou Pulvérisation	ECDOSURV.sys
Soudure Goujons	TUCKSURV.sys
Soudure 1 pince pneumatique	SR1PSURV.sys SRGESURV.sys SARPFRAN.sys
Soudure 2 pinces pneumatiques	SR2PSURV.sys SRGESURV.sys SRG2SURV.sys SARPFRAN.sys
Soudure avec pince électrique 1 swit	SRELSURV.sys SRGESURV.sys SARPFRAN.sys
Soudure ARC	ARCSURV.sys
Rivetage Electrique	RIVSURV.sys
Soudure électrique avec séquenceur ARO	SRELSURV.sys SRGESURV.sys SARPFRAN.sys
Soudure double carte Swit sur CAN	SRE2SURV.sys SRGESURV.sys SRG2SURV.sys SARPFRAN.sys
Soudure double pince électrique avec 2 swits CAN et 2 séquenceurs ARO	SRE2SURV.sys SRGESURV.sys SRG2SURV.sys SARPFRAN.sys

Ces modules contiennent les surveillances d'outil ou process. (Présence outil, air, eau...), ainsi que la prise en compte des demandes asynchrones (Rodage, Changement électrodes, Purge...).

Les textes sont dans le répertoire a:\textes\langue, avec *langue* = Français, Espagnol ou Anglais.

Les textes du noyau d'avant plan sont dans le fichier AVPFRAN.sys (resp AVPESP et AVPANG).

Les textes de chaque process sont ensuite dans les fichiers de type:

*Process + langue.sys*

Ex: le fichier ARCESP.sys contient tous les textes du process Soudure Arc en langue Espagnole.

Pour information

## XV – FONCTIONNEMENT DES SORTIES VERS L'AUTOMATE

Les sorties robot vers automate (autres que celles du dialogue code cycle et autorisations de zones) sont au nombre de 9. Leur fonctionnement est décrit ci-après:

### 1 – Do\_Appel\_TravA

Cette sortie sert à la gestion des travaux centralisés, lorsque l'on souhaite que tous les robots aillent en traj de service en même temps.

Elle est activée dès que le robot reçoit une demande de service de type A, soit Rodage dans le cas de la soudure SR. La demande peut provenir de la CPS, de la carte Swit, d'une demande de service, ou d'une demande centralisée par l'automate.

Elle informe l'automate qu'une demande de rodage est présente, ce qui lui permet de forcer une demande de rodage sur les autres robots, par l'entrée *di\_Dmd\_TravA*.

Elle reste à 1 jusqu'à ce que le travail demandé soit terminé sans erreur (fin de rodage).

En fin de rodage, le robot attend que l'entrée *di\_Dmd\_TravA* retombe à 0. (anti-répétition).

### 2 – Do\_Appel\_TravB

Idem que ci dessus, mais pour le changement d'électrodes centralisé.

NB: La demande de changement électrodes est prioritaire sur la demande de rodage, quelle que soit sa provenance.

### 3 – Do\_Error\_Sys

Cette sortie indique qu'une erreur d'exécution du programme s'est produite. Le message d'erreur correspondant reste affiché jusqu'à validation par l'opérateur.

A ce moment, la sortie retombe. Toutes les erreurs "système" sont gérées, par exemple:

- Erreur disquette.
- Erreur de programmation trajectoire.
- Erreur de syntaxe.
- Point non atteignable.

- Etc.

Attention: les erreurs Application **ne font pas** monter cette sortie. Il s'agit uniquement des erreurs "système".

Il s'agit d'erreurs majeures qui ne devraient se produire qu'en phase de mise au point.

#### 4 - Do\_Def\_Outil

Cette sortie indique qu'un défaut bloquant s'est produit sur **l'outil** (et non sur le robot).

Par exemple:

- Défaut bloquant groupe Encollage.
- Défaut Ouverture pince.

Elle monte dès que le défaut apparaît, et retombe dès que l'on acquitte le défaut. Elle remonte à nouveau si le défaut est toujours présent au redémarrage. Donc, si l'on acquitte le défaut et que l'on passe 1 heure à réparer l'outil avant de relancer, la sortie restera à 0 pendant 1 heure.

#### 5 - Do\_Def\_Choc

Cette sortie indique 2 choses, soit:

- La supervision de mouvement s'est déclenchée (collision).
- Un défaut DEC est apparu (si déclaré).

La sortie reste à 1 jusqu'à acquittement du défaut. Elle remonte si le défaut est toujours présent au redémarrage.

#### 6 - Do\_Robot\_Ok

Cette sortie monte si le robot est:

- En Auto.
- Sous Asservissements (motor on).
- Programme En Exécution.
- Aucun défaut bloquant en cours, quelque soit sa source.

Elle retombe dès que l'on perds une des conditions ci-dessus, **sauf** en Changement d'Electrodes et sur Présentation Colle (les asservissements sont coupés par sécurité, mais la sortie reste à 1 pour éviter que l'automate ne l'interprète comme une panne).

**Attention: cette sortie n'indique rien sur la position physique du robot.**

**7 – Do\_StrajAuto**

Cette sortie indique que le robot est:

- En Auto.
- En Execution de programme.
- Sur trajectoire apprise.

Elle **n'indique pas** qu'un défaut bloquant est présent. En effet, si un défaut Groupe Bloquant apparaît, par exemple, cette sortie reste à 1.

Elle retombe dès que l'on perds une des conditions ci-dessus. En cas de déplacement du robot au Joystick, puis relance en mode Auto, cette sortie remonte lorsque le robot **est recalé sur la trajectoire**.

**8 - Do\_Alarme\_Outil**

Cette sortie est le reflet des alarmes non bloquantes provenant de la CPS (Alarme fin de vie Electrodes), de la Swit (Idem), ou de l'antenne d'encollage (Defaut non bloquant groupe encollage).

Elle fait également clignoter la verrine rouge, sauf si celle ci est déjà allumée en fixe.

Elle retombe lorsque l'alarme retombe.

**9 – Do\_Sa\_Robot**

Cette sortie indique que le robot est **physiquement** soit au Repli, soit au PEO. Elle retombe dès que le robot n'est plus dans l'une de ces positions.

Elle permet d'effectuer des mouvements du moyen en mode manuel Automate. Elle n'est pas prévue pour être utilisée dans la gestion des automatismes en mode normal.

Notamment, il ne faut pas l'utiliser pour déclencher le départ de la pièce, ou tout autre mouvement du moyen.

## XVI – FICHIER LISEZ-MOI DE LA DISQUETTE

Ce fichier est au format Texte Simple et peut donc être lu avec n'importe quel éditeur de texte, y compris sous DOS.

VERSION 3.76 du 23 JANVIER 2006  
(Version de Test)

--> Correction du problème de relance suite à un redémarrage de rôdeuse sur abattant électrique.

VERSION 3.76 du 02 OCTOBRE 2005 (Version Validée)

- SRE2BASE : Allumage de la verrine de défaut à la question : "DEFAUT SOUDURE" "F1:ressouder le point" "F2:Sauter le point"  
Routine "Soude"
- DEP2BASE : Nouvelle variable MemoPile permettent de mémoriser la pile travaillée dans chaque container
- Ajout d'un message de prévention à la fermeture pince suite au changement d'électrodes
- Ajout de l'allègement du système de mise à jour pour éviter le dépassement de la mémoire
- Ajout des fichiers pour l'installation des modifications sur les BACK\_ACTIONS. Le fichier écran Back\_Act.cfg doit être chargé dans le robot après toute première installation par la procédure "Charger" dans le screen viewer.  
Il permet d'activer le nouvel écran du screen viewer pour la Pseudo\_Init. Une fois chargé, cet écran fera partie intégrante des paramètres sauvé.
- NOTE : Pour charger un écran, l'utilisateur à une minute après chaque redémarrage pour aller dans le screen viewer et pour appuyer sur charger afin de choisir le fichier CFG qui correspond à l'écran désiré.
- Initialisation des valeurs de poolrate des carte PROFIBUS à 30.
- Gain de temps de cycle sur les changeurs d'outils avec pince électriques.

VERSION 3.75 du 07 JUILLET 2005

- Correction de problèmes sur reprises de cycle

ATTENTION : la pseudo Init ne peut être activée par les action manuelle. L'init\_Pince/pseudoInit à donc été supprimée dans les actions manuelles quand le robot est de type changeur d'outils.

- Modification de la fonction de dépalettisation
- modification pour tester\_arret seulement en AUTO

VERSION 3.74 du 07 MARS 2005

07 Mars 2005 - Correction des problèmes "STOP PENDANT RODAGE" rencontré lors de rodage sur les pince électriques.

Correction de la déclaration du code cycle en 255

21 Decembr 2004 - Rétablissement de la surveillance continue du recul abattant en mode automatique.

19 Novembr 2004 - Allumage de la verrine sur demande de visualisation colle.

22 Octobre 2004 - Modification de la position du "NON" à la question "RAZER LES COMPTEURS ?"  
pour toutes les pinces à souder.

06 Septemb 2004 - Ajout de fonctions pour pouvoir utiliser la nouvel repprom corrigeant  
les pb de pince non fermée sur changeur d'outil.

Correction des problèmes de rodage pince pneumatique avec contrôle de rotation sur application SSPF (problème réglé depuis un certain temps en Angleterre.  
Ecris dans LISEZMOI pour rappel.)

03 Septemb 2004 - Correction de l'affichage du message retenter sur 1 pince quand l'autre rode.

31 Aout 2004 - Ajout d'une sortie supplémentaire Do\_Inhib\_CtlPos pour la pseudo souder d'un init lors d'un changement d'électrode sur une pince à robot changeur d'outils.

18 Aout 2004 - Correction du pb de pièce mal soudée sur la pince 1 pendant le rodage de la pince 2

18 Aout 2004 - Ajout de la possibilité de faire une demande d'arret en fin de chaque process.

18 Aout 2004 - Correction du pb d'allumage de la verrine et de la retombée du robot\_Ok sur un arrêt lors du rodage avec une ou deux pinces pneumatiques.

13 Aout 2004 - Correction du blocage de la tache de fond survenant sur des robots de soudure elec.

13 Aout 2004 - Ajout d'un message pour prevenir le recul de l'abattant suite à un spécial 3

10 Août 2004 - Ajout de la sortie weld error en sim\_board lors de la MAJ auto  
pour les métiers soudure elec.

28 Juin 2004 - Correction du pb re reprises à MU.

01 Juin 2004 - Ajout de la fonctionnalité OKPF pour autorisation de mouvement par automate

L'entrée OKPF sera déclarée sur l'API\_BOARD en 3 et inversée. Ceux qui voudront utiliser cette fonctionnalité devrons enlever l'argument inverse.

#### VERSION 3.73 BETA du 12 Juin 2004

12 Juin 2004 - Correction du pb sur le rodage masqué de la pince au sol lors du travail de la pince embarquée.

01 Juin 2004 - Activation de la sortie alarme outil sur alarme tube contact.

18 Mai 2004 - Nouvelle modification pour le problème de zones sur reprises (Cas de SX)

18 Mai 2004 - Correction de l'erreur de position pince sur double pince avec changeur  
APPLIED lors du changement d'électrodes.

12 Mai 2004 - Modification du fichiers SWUSRC.SYS pour permettre la reprise de point efficacement.

10 Mai 2004 - Fermeture en PO pour rodage avec effort de soudure pour éviter TimeOut PO avec les pinces au sol.

29 Avril 2004 - Correction de l'erreur de référence à l'entrée Weld\_Error lors d'une première installation sur les doubles princess.

14 Avril 2004 - Ajout d'un choix pour razer ou non les compteurs tubes lors du changement sur robots ARC : "CHANGEMENT DE TUBE EFFECTUE ?" OUI/NON

08 Avril 2004 - Correction du problèmes d'incohérence cames en fin de cycle.

05 Avril 2004 - Passage en persistant du booléen arc\_Decras\_Masq pour pouvoir activer la fonction sans P-START.

30 Mars 2004 - Vérification préalable de l'état repos des capteurs de position avance/recul décrassage pour les torches arc au sol.

19 Mars 2004 - Non affichage du message "Confirmez le changement electrodes ?" sur les doubles pinces pneumatique et électrique avec séquences ARO.

ATTENTION : Le principe d'utilisé un changeur d'électrodes automatique avec un robot double pince dont une embarquée n'a pas été demandé par PSA à ABB.  
Ce type de fonctionnement DOIT rester execptionnel.

18 Mars 2004 - Ajout de la commande du robot arrete si demande d'arret et position repli.

18 Mars 2004 - Plus de message "Soufflage" bloquant sur les double pinces pneumatique

18 Mars 2004 - Correction de l'erreur liée au ChangeurOk (cas n'arrivant que sur double pince électrique avec changeur d'électrodes automatique).

15 Mars 2004 - Ajout de la gestion de l'arret cycle sur défaut...

15 Mars 2004 - Correction de l'affichage du message de soufflage avec les doubles pinces électriques.

12 Mars 2004 - Correction du problème de production de pièce entre le rodage et le changement d'élecrodes.

10 Mars 2004 - Correction des problèmes lors du rodage sur les pinces masqué des robot SELSELPF et SEL2PF avec séquence ARO.

17 Février 2004 - Correction du non redémarrage de la rodeuse sur défaut petite ouverture avec rodeuse sans contrôle de vitesse.

17 Février 2004 - Correction des problèmes lors de la mise à jour automatique.

11 Février 2004 - Installation du palliatif pour le non affichage de message lors du pilotage en position recul d'un abattant électrique suite à un redémarrage îlot.

11 Février 2004 - Ajout de la possibilité de tester une traj d'encollage en mode Manu (en passant par l'apprentissage).

06 Février 2004 - Modification pour pouvoir passer de la GO à la fermeture complète sur demande dans un cycle. Sur action manuelle, la pince ira en PO puis se fermara à l'action ferme\_pince suivante.

05 Février 2004 - Correction de l'erreur de référence à la données SAVPTXT en ARP sur simple pince PNEU.

05 Février 2004 - Modification de la déclaration des E/S pour les encolleurs doseurs lors de la première install (déclarés sur la bonne carte E/S).

05 Février 2004 - Possibilité d'utiliser FraiseEpaisse pour le capteur de brillance.

26 Janvier 2004 - Fermeture en PO pour rodage avec effort de soudure pour éviter TimeOut PO.

26 Janvier 2004 - Raz du do\_tete1\_start par sécurité action.

26 Janvier 2004 - Clignotement de la verrine sur défaut SWIT. (ajout de la sortie weld\_error en 32 sur VIRT\_BOARD).

26 Janvier 2004 - Inhibition du message "CHANGEZ LES ELECTRODES PINCE ..." quand on utilise un changeur d'électrodes automatique.

23 Janvier 2004 - Utilisation d'un booléen pour activer le contrôle d'alignement.

23 Janvier 2004 - Ajout d'une sécurité pour ne pas souder les fraises.

23 Janvier 2004 - Modification pour la gestion des reprises.

23 Janvier 2004 - Correction sur la gestion des pinces à basculement.

23 Janvier 2004 - Modification pour la supervision : la supervision est toujours désactivée sur changement d'outils. Le trajectoiriste DOIT mettre un motionsup avec un une tunevalue s'il considère le mouvement risqué. L'appli reforcera la tunevalue à 100 en fin de mouvement.

05 Janvier 2004 - Suppression de la sécurité de verrouillage Robot / Rodeuse permettant de repartir en cycle alors que le rodage masque n'est pas terminé. Il devient de la responsabilité de l'intégrateur de mettre "Dans\_Zone\_Rod" dans ses trajs (Doc. Page 97).

05 Janvier 2004 - Correction defaut de non-réaffichage automatique de l'écran de production après 60 secondes si il a été inhibé par appui sur P2.

05 Janvier 2004 - Suppression du clignotement de la verrine sur valeur de supervision trop élevée.

05 Janvier 2004 - Raz du boolean fraise\_epaisse suite au rodage.

05 Janvier 2004 - Ajout du palier pour les problèmes de lecture de code cycles.

#### VERSION 3.72 du 20 Octobre 2003

1 - Validation du Contrôle Alignement Electrodes.

2 - Correction erreur ligne 167 de Robauto pour SELSEL.

#### VERSION 3.72 du 13 Octobre 2003

1 - Correction defaut en SR pneumatique apparu suite aux dernieres demandes d'évolution: comme on charge un programme soudure pour ouvrir la pince, on ne peut plus razer le compteur de rodages. Il faut donc remettre le prog rodage apres ouverture.

2 - Correction de l'erreur de type "num" pour les métiers ARC.

VERSION 3.71 du 29 Septembre 2003

1 - Correction de l'erreur qu chargement d'une application DEPAL

2 - Modification de la gestion des abattants rodeuse sur Restart : le recul est repiloté systématiquement et non plus simplement si on a perdu le contrôle.

3 - Correction erreur de reference sur Swit\_Lock en cas de modif de parametres systeme ou de Reboot de la baie.

4 - Correction du pb de detection de rotation rodeuse apparu à SN.

VERSION 3.7 du 5 Septembre 2003

1 - Modification du comportement sur perte presence pièce (demande de D.Lehrer).

Le message affiché devient bloquant et le départ distant redevient autorisé.

VERSION 3.7 du 10 Aout 2003

1 - Correction des erreurs lors du rodage masqué.

2 - Autorisation du départ distant en attends\_PP (demande Dreschler).

VERSION 3.7 du 22 Juillet 2003

1 - Validation du système de rodage avec abbattants électriques.

2 - Utilisation d'un paramètre pour le timeout d'amorçage de l'arc.

Il faut donc ajouter Sr\_Tps\_Amorce en Constante, variable globale (non locale)

dans le module configur (fichier configur.sys)

Utilité : Même s'il s'agit d'un timeout, laisser débobiner le fils pendant

trop longtemps peut engendrer certains problèmes côté dérouleur.

3 - Mise en place de l'instruction "Lance\_Rodeuse1" pour la pince 1 et "Lance\_Rodeuse2"

pour la pince 2 (si un jour le cas se produit...) à ajouter pour l'anticipation au rodage des pinces embarquées.

4 - Pour un métier Em + (Ed ou Ev) + S, il faut faire attention au signaux di\_Grp\_Bloq2 et di\_Grp\_Alarm2 qui risquent d'être en conflit. Etant donné qu'un

seul robot est dans cette situation, un modification manuelle sera demandée.

#### VERSION 3.7 du 08 Juillet 2003

1 - Mise au point du contrôle Brillance avec pince Elec et pince Pneu (testé à SN).

Attention, cette fonction n'est bien sûr utilisé que pour les pinces embarquée.

Il n'est donc pas possible d'utiliser la brillance pour une pince 2.

2 - Mise au point de l'affichage pour les actions manuelles.

3 - Ajout d'un TimeOut pour les rodeuses des pinces au sol afin d'éviter leur surchauffe en cas de disfonctionnement de l'abattant rodeuse.

4 - Evolution du process tucker pour l'aiguillage avec deux tête sur changeur d'outils

#### VERSION 3.7 du 25.06

1 - Gestion de l'arret cycle.

2 - Suppression de SwTimer2 (remplace par SwTimer).

3 - Ajout gestion de la sortie DesactGun au redemarrage.

4 - Remis l'ancienne methode pour l'init pince (init normale au lieu de pseudo-init) et generalisation de la double tentative en cas d'echec de la premiere.

La pseudo-init reste utilisee uniquement dans le cas du changeur.

5 - Ajout fonction Controle de Brillance (Attention aux modifs dans la trajectoire de rodage, voir Notice).

6 - Ajout appel automatique des traj de presentation colle (voir Notice).

7 - Sur demande de validation de l'operateur, allumage de la verrine rouge en fixe et montee de la sortie do\_def\_outil (meme si il n'y a pas de defaut outil). Demande de A.Roth du 20.02.03

8 - L'alarme FDVE electrodes n'est plus signalee si on a un changeur auto d'electrodes en etat de marche. (demande de A.Roth du 20.02.03).

9 - Lorsque les chargeurs sont vides, la sortie do\_def\_outil ne monte que sur apparition de l'alerme FDVE. (demande A.Roth du 20.02.03).

10 - Arc: L'arc n'est plus coupe sur perte controle torche si la surveillance

est inhibee (demande de A.Roth du 20.02).

11 - Ajout securite sur Special 4: la sortie do\_StrajAuto ne remonte plus.

12 - Remise systematique de l'eau avant soudure si l'on est avec soudure.

13 - Declenchement d'un defaut si on est sans eau a la fin d'une soudure, sauf si le controle est inhibe.

- 14 - Amelioration methode de depart distant.
- 15 - Passe le TimeOut sur defauts fermeture pince elec de 5 à 2 secondes.
- 16 - Passe le TimeOut sur defauts d'amorçage en S.ARC de 0.8 à 2 secondes.
- 17 - Remise systématique de l'eau avant chaque soudure si mode soudure
- 18 - Reprise intégrale de la gestion rodeuse pour pouvoir anticiper le démarrage.  
Routine "Lance\_Rodeuse".
- 19 - Gestion des abattants électriques.
- 20 - Ajout gestion Defaut Coffret Rodeuse (entrée di\_DefRod).

VERSION 3.69\_7 du 06.06

- 1 - Correction de l'application dépalettisation pour saut de pile indésirable.
- 2 - Correction du rodage de la pince 2 pour les métiers à 2 pinces pneumatique embarqué (cf Mulhouse)
- 3 - Correction du problème d'initialisation du mode soudure pour les métiers doubles pinces pneumatiques.
- 4 - Corrections du pb de rodage non effectué pour les robots SELSELPF sans détection de ralentissement rodeuse.
- 5 - Correction du problème de disjonction des armoires de soudure.
- 6 - Mise au point de l'appli pour les changeurs Exrod.

VERSION 3.69\_7 du 02.05

- 1 - Correction défaut SWIT 24 pour les pinces au sol avec séquence ARO
- 2 - Correction problème de reprises de cycle avec coupure d'eau et sélecteur d'eau sur Arrêt (Problème remonté par M. Ketta, Aulnay)

VERSION 3.69\_7 du 29.04

- 1 - Corrections de l'erreur SWIT 24 arrivant sur les robots avec séquence externe.

VERSION 3.69\_7 du 20.03

- 1 - Correction des problèmes de rodages des pinces pneumatiques (SR1PACTU.SYS)

VERSION 3.69\_7 du 7.3 POUR TEST CHANGEUR SCIAKY

- 1 - Suppression de SwTimer2 (remplace par SwTimer).
- 2 - Ajout gestion de la sortie DesactGun au redemarrage.
- 3 - Remis l'ancienne methode pour l'init pince (init normale au lieu de pseudo-init) et generalisation de la double tentative en cas d'echec de la premiere.  
La pseudo-init reste utilisee uniquement dans le cas du changeur.
- 4 - Sur demande de validation de l'operateur, allumage de la verrine rouge en fixe et montee de la sortie do\_def\_outil (meme si il n'y a pas de defaut outil). Demande de A.Roth du 20.02.03
- 5 - L'alarme FDVE electrodes n'est plus signalee si on a un changeur auto d'electrodes en etat de marche. (demande de A.Roth du 20.02.03).
- 6 - Lorsque les chargeurs sont vides, la sortie do\_def\_outil ne monte que sur apparition de l'alarme FDVE.(demande A.Roth du 20.02.03).
- 7 - ARC: Si la surveillance est inhibee, l'arc n'est plus coupe en cas de perte  
du controle presence torche, et le robot ne s'arrete plus.  
(demande A.Roth du 20.02)

#### CORRECTIONS OU PRECISIONS SUR 3.69\_6

- 1 - La donnee Swit\_Cur\_Gun dans le module SwTimer2 doit ^tre declaree en persistante simple et non pas en tableau.
- 2 - Pour le process simple pince electrique: verifier dans les parametres systeme (modules taches) que c'est bien SWTIMER qui est declare et non pas SWTIMER2.
- 3 - Correction affichage du message "Pistolet inconnu" en process Doseur+ Volucompteur.
- 4 - Ajout instruction Ferme\_pince pour le changement electrodes en pince pneumatique en venant de la GO.
- 5 - Correction sortie do\_Puis\_Soud non declaree lors d'une montee de version  
automatique en sr swit aro sans changeur.
- 6 - La parite etant maintenant PAIRE en process Doseur, si l'antenne d'encollage  
n'est pas a jour, il faut inverser la sortie do\_Dos\_Parite.
- 7 - Resolution probable du pb aleatoire de double changement electrodes sur demande de service.

#### VERSION 3.69\_6 du 12.12

- 1 - Changement de nom de 2 sorties en Rivetage: (Demande de JP Brietz)  
do\_Broche\_Equil --> do\_Recule\_Mat  
do\_Broche\_Invers --> do\_Avance\_Mat
- 2 - Modification gestion equilibrage en Rivetage.

3 - Remise securite sur pinces a basculement.

4 - Resolution du pb OverRun consigne pince.

Validations faites avec RW 3.2\_48P :

S (ARO)	Ok : MA Azevedoneto	le 02/12/02 Robot n° ?
SEL (ARO)	Ok : MU Lehrer	le 10/11/02 Robot n° 35518
14353		
SPF (ARO)	Ok : MA Azevedoneto	le 02/12/02 Robot n° ?
SELM applied (SWIT)	Ok : SN Nizon Brietz	le 27/11/02 Robot n° R58223
SELSELPF (SWIT)	Ok : SN Nizon Brietz	le 27/11/02 Robot n° R432
SELSELPF (ARO)	Ok : MA Azevedoneto	le 02/12/02 Robot n° ?
S2ELPF (ARO)	Ok : MA Azevedoneto	le 02/12/02 Robot n° ?
Arc	Ok : MU Lehrer	le 09/10/02 Robot n° 35271 10680
Ed	Ok : MU Lehrer	le 09/10/02 Robot n° 36302 17896
DEPAL 1	Ok : AN Baud Dodemont	le 16/11/02 Robot n° ?
DEPAL 2	Ok : AN Baud Dodemont	le 16/11/02 Robot n° 58824
16147		
Goujon	Ok : AN Baud	le 10/11/02 Robot n° ?
EePFEmPFSPF-TI	Ok : MU Lehrer	le 14/10/02 Robot n° 40032
12333		
C	Ok : SX Rumper	le 08/11/02 Robot n° 37443
14145		
M	Ok : SX Rumper	le 08/11/02 Robot n° 37434
13138		
R	Ok : SN Brietz	le 10/12/02 Robot n° R286

VERSION 3.69\_6 du 25.11

1 - Resolution Pb "Reio.c" en mettant 1 tempo de 0.01 entre Set Start\_Soud et Reset Squeeze. Pb de saturation de la pile de requetes au systeme d'E/S.

2 - Suppression controle basculement ferme sur PO et GO (pinces pneu).

VERSION 3.69\_6 du 15.11

1 - Gestion de l'arrêt cycle tel que décrit plus bas.

--> Re-supprime provisoirement car premature. (mais fonction validée).

2 - Parité PAIRE sur Encollage.

3 - Correction pb Attente Sans Soudure sur soudure ARO apres abandon du rodage.

4 - Integration modifs Changeur d'Electrodes. (Demande A.Roth du 5.11).

5 - Correction defaut Swit 26 sur Ferme\_Pince+Init\_Pince en SREL. (tempo 0.1)

6 - Correction bugg sur evolution graphe de change elec pour la pince 2 en applis double pince.

- 7 - Forage effort de fermeture a 150 Dan sur changement elec sur toutes applis pince electrique.(Demande A.Roth + JP Brietz du 12.11)
- 8 - Nouveau Mapping d'E/S pour tous changeurs d'electrodes. Voir le fichier Mapping\_ES.XLS de la disquette d'installation pour mise a jour manuelle. (Pris en compte lors d'une premiere Instal). (Demande A.Roth du 13.11)
- 9 - Augmente Tempo de pre-gaz en S.ARC (0.5 --> 0.8). Demande A.Roth du 13.11.

VERSION 3.69\_6 du 25.10

- 1 - SUPPRIME PROVISOIEMENT  
Gestion de l'arrêt Cycle: celui ci est pris en compte:  
  - A chaque appel de AppelTraj.
  - A chaque appel de Att\_Acces\_Zone.
  - Au PEO, en attente de code.
L'arrêt Cycle est prioritaire sur:
  - Les demandes de service.
  - Les demandes de rodage ou chgt electrodes.
  - Les demandes de changement d'outil.
  - La demande de Repli.
- 2 - SUPPRIME PROVISOIEMENT  
Mis parite PAIRE au lieu de IMPAIRE pour double doseur.
- 3 - Cas du changeur d'outil: si on redemarre sans outil apres l'avoir depose manuellement, la Swit affiche "Numero de pince errone" (message non bloquant)
- 4 - Correction Pb Interrupt Queue Full si la verrine clignote et que la sauvegarde sur disquette est trop longue (accumulation des interruptions cycliques non exécutées pendant la sauvegarde).

VERSION 3.69\_5 du 16.10

- 1 - Ajout nouvelle donnee string "INFO" dans le Configur. Cette string donne un commentaire sur la dernière version de l'appli ayant tourné, par ex "Beta 4 du 10.10".
- 2 - Gestion des changeurs d'electrodes SCIAKY, EXROD et PCI de façon indépendante. Pour l'instant, le changeur PCI est copié sur Exrod, dans l'attente de spécifiques, mais il existe.  
Ajout de la possibilité d'abandonner le changement automatique en cas de défaut changeur: dans ce cas, c'est le changement manuel qui est exécuté.

La verrine clignote si le magasin est vide et le message "CHANGEUR VIDE" apparaît.

ATTENTION: voir le fichier TRAJECT d'exemple pour adapter la programmation de la traj T\_Chge\_Elec. Le flag chg\_elec\_instal est remplacé par chg\_exrod\_instal ou

chq\_sciak\_instal ou  
chq\_pci\_instal  
Attention: les 3 flags doivent etre declares.

VERSION 3.69\_5 du 04.10

- 1 - Validation des reprises de traj sur changement d'outil a SX.
- 2 - ARC: creation nouveau compteur pour lancer la traj de changement de tube contact en automatique.  
Creation d'une alarme 5 cycles avant le changement de tube.
- 3 - ARC : ajout parametre optionnel POSTGAZ dans la routine Fin\_Cordon.
- 4 - ARC : Passage du time out de fcy soudage de 1 a 5 secondes.
- 5 - Encollage VOLUCOMPTEUR: il faut renommer dans le fichier Trajs:  
EpoxyL --> ColleDebitL  
EpoxyC --> ColleDebitC  
Presente\_Epoxy --> Presente\_Debit  
T\_Pres\_Epoxy --> T\_Pres\_Debit  
Purge\_Epoxy --> Purge\_Debit  
  
Penser aussi a remplacer dans ACTMSITE Purge\_Epoxy par Purge\_Debit  
Demande de W.Jeanvoine car on ne parle plus de produit, mais de procede.
- 6 - Creation gestion double equilibrage pour pinces en C ou GP.  
Nouvelle sortie do\_Equil en Action Board 10.(Demande W.Jeanvoine)
- 7 - ARC: Integration modifs A.Roth du 3.10.
- 8 - ARC : Prise en compte uniquement des cycles comportant effectivement du soudage pour l'incrementation des compteurs (demande W.Jeanvoine).  
Il faut que au moins 1 traj du cycle ait demande au moins 1 cordon, et qu'il se soit effectivement realise (mode sans piece ou sans soudure)  
pour que les compteurs de decrassage et de changement de tube s'incrementent.
- 9 - Correction du defaut de Restriction Entrees Sorties provoquée par l'arrêt  
de la tache de fond suite à une tentative de lecture / écriture dans le fichier de suivi en Ramdisk, refusée deux fois de suite par le système.  
(1 refus, ca va, 2 refus, bonjour les dégâts). Re-écriture de la gestion d'erreurs.
- 10 - Par consequent, remise en service de la fonction SUIVI.

VERSION 3.6.9\_4 du 13.09

- 1 - Modification de la gestion du defaut Tucker Arrete.(Pb Aulnay)
- 2 - Correction pb flag Enprod qui passait à False en fin de trajectoire Tucker (Pb Madrid).

3 - Modif pour defaut ouverture pince en rodage masque. La pince est consideree ouverte si elle est au moins a 1.1 fois sr\_Rod\_Epaisseur.

VERSION 3.69\_3 du 1.09.02

- 1 - Mise en place nouvelle methode d'init pince pour changeur.
- 2 - Palliatif au defaut systeme qui n'execute pas les interruptions dans certains cas de STOP, ce qui empêchait de positionner le flag de surveillance de zone. Resultat: la zone était bien libérée, mais toujours attendue !
- 3 - Correction Pb du do\_StrajAuto qui ne remontait pas suite à action STOP, marche avant en pas à pas, Depart.
- 4 - Modification principe de lecture de la position pince pour les pinces électriques en rodage masque (lecture de Gun\_Pos au lieu de CJointT)
- 5 - Correction erreur de référence sur Swit\_Lock\_Gun en cas de modif de paramètres système ou de Reboot de la baie, entraînant l'arrêt de la tâche de fond et la restriction d'entrées sorties.
- 6 - Suppression provisoire de la gestion du fichier de suivi pour voir si ce n'est pas ça qui plante la tâche de fond du fait que le fichier est v,rol, par le système.

VERSION 3.69\_2 DIFFUSION LE : 27 08 02

Correction des erreurs suivantes :

- 1 - Erreur de référence suite à la modification des paramètres systèmes sur me étier pince électrique avec séquence ARO.
- 2 - Problème de blocage de CPS suite à un redémarrage de baie.
- 3 - Problème des purge trop longues.

--> ATTENTION : Pour le pb no 1, des num sont à déclarer pour les montées manuelles. Ce cas se présente lors de la montée depuis une 368. La montée automatique gère ce cas dorénavant. Les données à ajouter sont les suivantes à déclarer dans le configfile :

```
PERS num sr_FREQ_SECTEUR:=50 ;
PERS num sr_ROD_FORCE:=120;
PERS num sr_ROD_TEMPS:=2;
```

VERSION 3.69\_1 DIFFUSION LE : 02 08 02

Correction des erreurs suivantes :

- 1 - Erreur de déclaration sur des groupes d'E/S en double pince électrique
- 2 - Erreur de déclaration sur de signaux analogiques en SELSEL

VERSION 3.69 DIFFUSION LE : 25 07 02

--> ATTENTION : La commande de rodage pour les pinces pneumatiques est Rode\_Pince avec en paramètre le numéro de la pince à roder.

Rappel :

La fonction "rodage" existe depuis toujours sur les différentes versions de l'applicatif PSA

En version 3.65 la fonction "rodage" existait seule

En version 3.67 la fonction "rodage" et les fonctions "rode\_pince 1"  
ou "rode\_pince 2 "

co-existaient (ceci pour faciliter les compatibilités et les travaux des intégrateurs, en

vue de la suppression de la fonction "rodage" (refonte de l'applicatif pour un gain de place

mémoire Cf Cartes DSQC 323-DSQC324 en Spneu))

Nota : sur le site de Madrid par exemple (ou l'information a bien circulée) la fonction "rodage"

n'est plus utilisée sur les robots en 3.67

En version 3.68 la fonction "rodage" a été supprimée

Cette fonction est à remplacer dans les trajects des sauvegardes des versions 3.67 et antérieures.

Pour information