

Post-processeur PL71

Pour AUTOMGEN V6 et automates SCHNEIDER



CHAPITRE N°1.	3
INTRODUCTION	3
CHAPITRE N°2.	4
INSTALLATION	4
PARAMÉTRAGE	4
CONNEXION ENTRE LE PC ET LES AUTOMATES TELEMECANIQUE	6
CHAPITRE N°3.	7
PRISE EN MAIN	7
CHAPITRE N°4.	9
GÉNÉRALITÉS	9
PREMIÈRE PHASE DE TRADUCTION DU GÉNÉRATEUR DE CODE	9
DEUXIÈME PHASE DE TRADUCTION DU GÉNÉRATEUR DE CODE	9
MODULE DE DIALOGUE	9
CHAPITRE N°5.	10
LES QUATRE FICHIERS TEXTES ASSOCIÉS À UNE APPLICATION	10
LA DÉCLARATION UNITAIRE	10
LA DÉCLARATION DE TABLE LINÉAIRE	11
LA DÉCLARATION DE TABLE POUR AFFECTATION AUTOMATIQUE	11
LES TYPES DE VARIABLES AUTOMGEN	11
LE FICHIER MODÈLE « PL71.VUS »	12
LES FICHIERS « .SRT » ET « .END »	12
CHAPITRE N°6.	14
PARAMÉTRAGE DE L'AUTOMATE	14
DIRECTIVE DE SÉLECTION DU TYPE D'AUTOMATE	14
DIRECTIVE DE SÉLECTION DE LA TAILLE MÉMOIRE	14
DIRECTIVE DE PARAMÉTRAGE DES MODULES D'EXTENSION POUR LE TSX 17-20	14
DIRECTIVES DE PARAMÉTRAGE DES TEMPORISATIONS	14
DIRECTIVES DE PARAMÉTRAGE DES COMPTEURS	15
DIRECTIVES SPÉCIFIQUES AU TSX 17-20	15
CHAPITRE N°7.	16
LE MODULE DE DIALOGUE	16
PARAMÉTRAGE	16
CHAPITRE N°8.	17

TECHNIQUES AVANCÉES	17
INSERTION DE CODE PL71 DANS LES APPLICATIONS	17
UTILISATION DE LA TÂCHE RAPIDE	17
<u>CHAPITRE N°9.</u>	<u>18</u>
LIMITES DE COMPATIBILITÉ	18
<u>CHAPITRE N°10.</u>	<u>19</u>
UTILISATION DES COMPTEURS	19
<u>CHAPITRE N°11.</u>	<u>20</u>
UTILISATION DU COMPTEUR RAPIDE SUR UN AUTOMATE TSX 17-20	20
<u>CHAPITRE N°12.</u>	<u>22</u>
EXEMPLE COMPLET	22

CHAPITRE N°1.

Introduction

Le post-processeur PL71 est un module logiciel composé de plusieurs fichiers exécutables.

Il traduit les applications développées avec l'atelier logiciel AUTOMGEN en fichiers binaires PL71 et les télécharge dans les automates TSX 17-10 et TSX 17-20.

PL71 est une marque déposée de TELEMECANIQUE.

Le post-processeur permet d'effectuer la mise au point des applications depuis l'atelier logiciel AUTOMGEN en donnant accès aux fonctions de visualisation dynamique, de changement d'état des variables et de modification du mode de marche.

CHAPITRE N°2.

Installation

Pour installer le post-processeur PL71 sur votre disque dur, procédez de la façon suivante :

1. Placez la disquette repérée « Post-processeur PL71 » dans le lecteur de disquette A ou B,
2. Lancez l'exécution de « IRAINST. EXE »
3. Suivez ensuite les instructions du programme d'installation.

Paramétrage

Une fois l'installation terminée, lancez l'environnement AUTOMGEN suivez les instructions suivantes :

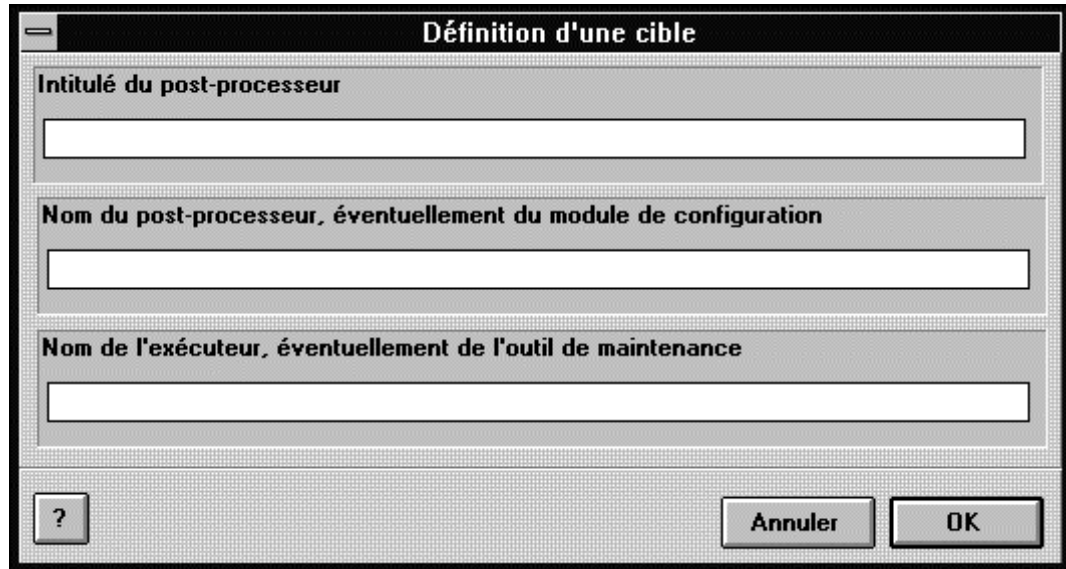


1. Sélectionnez la commande « Cibles ... » du menu « Option »,
2. Choisissez la ligne « PL71 » puis cliquez sur « OK »,
3. Choisissez le menu « Options », « Paramètres par défaut », le bouton poussoir « WPL7132.EXE »,
4. La boîte de dialogue de paramétrage du module de dialogue s'ouvre .



1. Sélectionnez la commande « Cibles ... » du menu « Option »,
2. Cliquez sur le bouton « Insérer »,

La fenêtre suivante est alors affichée sous WINDOWS :



Définition d'une cible

Intitulé du post-processeur

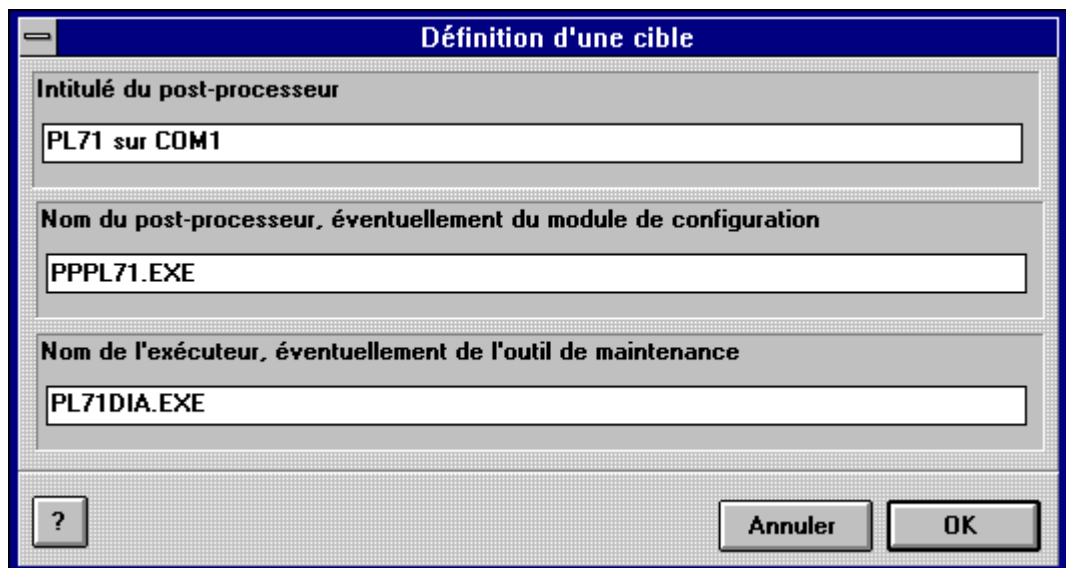
Nom du post-processeur, éventuellement du module de configuration

Nom de l'exécuteur, éventuellement de l'outil de maintenance

? Annuler OK

3. Entrez sur la première ligne l'intitulé du post-processeur. Cette ligne n'est pas fonctionnelle, elle permettra simplement de sélectionner le post-processeur dans l'environnement. Il est pratique d'entrer un nom générique (« **PL71** » par exemple) suivi du port de communication utilisé (« **COM2** » par exemple). Entrez « **PL71 sur COM1** » ou « **PL71 sur COM2** »,
4. Entrez sur la deuxième ligne le nom du générateur de code PL71. Entrez « **PPPL71.EXE** »,
5. Entrez sur la troisième ligne le nom du module de communication. Entrez « **PL71DIA.EXE** » pour utiliser le port COM1 ou entrez « **PL71DIA.EXE -C2** » pour utiliser le port COM2. Il faut un et un seul espace entre PL71DIA.EXE et -C2.

Exemple d'installation du post-processeur PL71 utilisé sur COM1.



Définition d'une cible

Intitulé du post-processeur

PL71 sur COM1

Nom du post-processeur, éventuellement du module de configuration

PPPL71.EXE

Nom de l'exécuteur, éventuellement de l'outil de maintenance

PL71DIA.EXE

? Annuler OK

Exemple d'installation sur COM2.:

Définition d'une cible

Intitulé du post-processeur
PL71 sur COM2

Nom du post-processeur, éventuellement du module de configuration
PPPL71.EXE

Nom de l'exécuteur, éventuellement de l'outil de maintenance
PL71DIA.EXE -C2

? Annuler OK

6. L'installation est terminée, cliquez sur le post-processeur que vous souhaitez utiliser puis sur le bouton « OK ».

Si vous voulez vérifier ou modifier l'installation d'un post-processeur, cliquez sur la ligne correspondante, puis sur le bouton « Modifier ».

Connexion entre le PC et les automates TELEMECANIQUE

AUTOMGEN utilise le même boîtier de communication (le câble 17ACC8) que le logiciel PL71.

CHAPITRE N°3.

Prise en main

Ce chapitre donne le déroulement précis permettant de compiler le fichier d'exemple et de l'exploiter sur un automate TELEMECANIQUE.

Si vous n'avez pas installé le programme d'exemple, relancez la procédure d'installation pour le faire.

1. Lancez l'environnement AUTOMGEN,
2. Cliquez sur l'option « Ouvrir Folio » du menu « Fichier »,
3. A l'aide du sélecteur de fichiers, ouvrez le folio « PL71.GR7 » qui se trouve dans le sous-répertoire « EXPL71 » du répertoire où est installé AUTOMGEN,
4. Cliquez sur l'option « Cibles ... » du menu « Option », sélectionnez la ligne « PL71 sur COM1 » ou « PL71 sur COM2 »¹ puis le bouton « OK »,
5. Si votre automate est un TSX 17-20 passez directement au point numéro 10,
6. Si votre automate est un TSX 17-10 choisissez la commande « Editeur de texte » du menu « Boîte à outils »,
7. Ouvrez le fichier « DEFPL71.VSY » qui se trouve dans le répertoire où est installé AUTOMGEN. Dans ce fichier, modifiez la ligne « \$TSX=1720 » en « \$TSX=1710 ».
8. Sauvez le fichier « DEFPL71.VSY »,
9. Cliquez sur l'option « Compiler » du menu « Compiler »,

La compilation est lancée : elle se termine par l'appel du post-processeur PL71.

Si le post-processeur n'est pas lancé, c'est probablement que l'installation n'a pas été réalisée correctement, reprenez point par point le chapitre « Installation ».

Si tout s'est bien passé, un fichier « PL71.BIN » a été généré dans le même répertoire que le fichier « PL71.GR7 ». C'est ce fichier qui sera transféré dans l'automate.

10. Cliquez sur la commande « Charger » du menu « Exécuter »,
11. Une fenêtre « Module de dialogue TE » s'ouvre et la connexion est réalisée²,
12. Une boîte de dialogue propose de télécharger le programme, cliquez sur « Oui », le téléchargement est alors effectué³,
13. Une boîte de dialogue propose de vérifier le programme, cliquez sur « Oui », la vérification est alors effectuée⁴,

¹si aucune de ces lignes n'apparaît, l'installation n'a pas été réalisée correctement, retournez au chapitre « Installation ».

²si ce n'est pas le cas, vérifiez que l'installation est en accord avec le port de communication utilisé (voir le chapitre « Installation »).

³ si un message d'erreur « le fichier ne correspond pas au type d'Api » apparaît, c'est que l'automate ne correspond pas au type inscrit dans le fichier « DEFPL71.VSY ». Reprenez au point numéro 7.

14. Cliquez sur la commande « Exécuter » du menu « Exécuter », l'automate passe en RUN,

15. Cliquez sur la commande « Visualiser » du menu « Debug », la visualisation dynamique est active sur le folio « PL71 ».

Ceci termine le chapitre « Prise en Main ».

⁴ la vérification permet de contrôler la cohérence entre le programme présent dans l'automate et le programme présent sur le PC.

CHAPITRE N°4.

Généralités

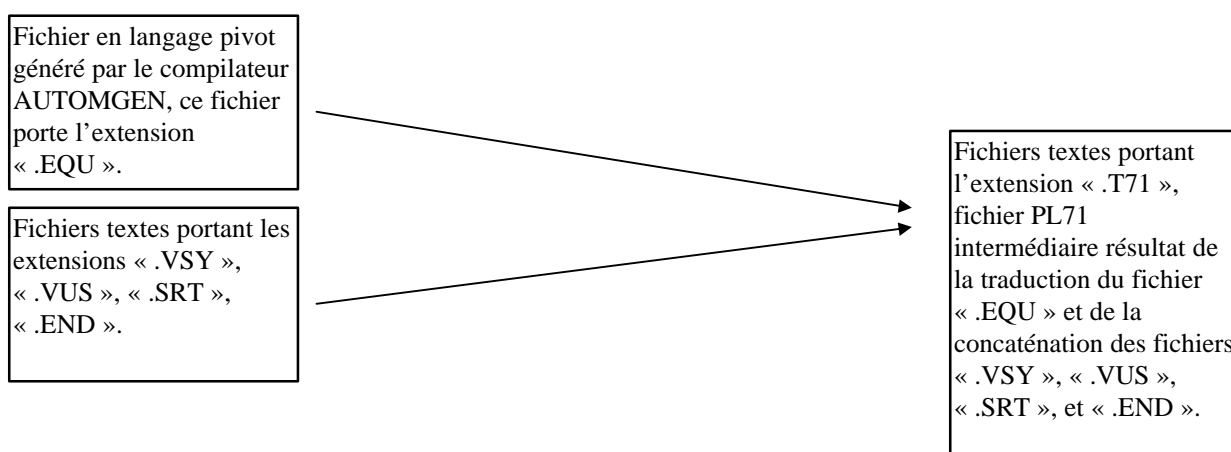
Le post-processeur se compose de différents modules détaillés ci-dessous. Pour rendre les explications plus compréhensibles, seuls les fichiers principaux et accessibles au programmeur seront mentionnés.

Première phase de traduction du générateur de code

« PPPL71.EXE » est l'exécutable qui réalise la première passe de traduction. Il traduit le langage pivot d'AUTOMGEN représenté par un fichier portant l'extension « .EQU » en fichier PL71 intermédiaire qui porte l'extension « .T71 ». En outre, ce générateur de code cherche des fichiers textes qui portent les extensions « .VSY », « .VUS », « .SRT », « .END » et les recopie dans le fichier de sortie s'ils existent.

La création de ces fichiers est à votre charge et sera détaillée dans le chapitre suivant.

Le schéma ci-dessous donne une description de la tâche accomplie par « PPPL71.EXE » :



Deuxième phase de traduction du générateur de code

« PL71COMP.EXE » (16 bits) ou « PL71CO32.EXE » (32 bits) sont les exécutables qui réalisent la deuxième phase de traduction. Ils traduisent le fichier « .T71 » en un fichier « .BIN » compatible avec le logiciel PL71. Ces traducteurs effectuent deux tâches : remplacer le nom des variables AUTOMGEN par le nom des variables PL71 et résoudre les sauts dans le programme. Un fichier « .LAP » est également généré, il contient la liste des instructions en langage PL71.

Module de dialogue

« WPL71DIA.EXE » (16 bits) ou « WPL7132.EXE » (32 bits) sont les modules de dialogue appelés par l'environnement AUTOMGEN. Ils sont utilisés pour les fonctions de téléchargement et de comparaison de programme, de visualisation dynamique, de changement d'état des variables et de forçage des modes de marche de l'automate.

CHAPITRE N°5.

Les quatre fichiers textes associés à une application

Comme il a été dit au chapitre précédent, le générateur de code récupère quatre fichiers textes lors de la première phase de traduction.

Si vous êtes débutant dans l'utilisation d'AUTOMGEN ou du post-processeur PL71, nous vous conseillons d'utiliser ces fichiers sans les modifier. Dans un deuxième temps, et si vous désirez réaliser des applications particulières (optimisation de la taille du programme, ajout de morceaux de programmes en langage PL71), vous pourrez étudier ce chapitre ainsi que le chapitre « Techniques avancées ».

Le générateur de code cherche les fichiers portant l'extension « .VSY », « .VUS », « .SRT » et « .END ».

Pour chacun de ces quatre fichiers, le générateur de code recherche :

- d'abord un fichier portant le même nom que le premier folio de l'application,
- si ce fichier n'existe pas, c'est le fichier portant le nom « DEFPL71 » et qui se trouve dans le répertoire où a été installé AUTOMGEN qui est sélectionné,
- si ni l'un ni l'autre de ces fichiers existent, alors le générateur de code considère cette partie comme vide et ne génère aucun message d'erreur.

Ces fichiers peuvent recevoir des commentaires placés à droite d'un caractère « ; » (point virgule).

Les fichiers « .VSY » et « .VUS » sont des fichiers de déclaration de variables. Ils donnent la correspondance entre les variables AUTOMGEN et les variables PL71.

Le fichier « .VSY » donne la correspondance pour les variables dites « Système ». Ce fichier contient également des directives de compilation (voir chapitre « Paramétrage » et « Techniques avancées »). Le fichier « DEFPL71.VSY » est fourni en modèle dans le répertoire où est installé AUTOMGEN.

Le fichier « .VUS » donne la correspondance pour les variables utilisées dans l'application. Le fichier « DEFPL71.VUS » est fourni en modèle dans le répertoire où est installé AUTOMGEN.

Il existe plusieurs syntaxes permettant de déclarer ces affectations.

La déclaration unitaire

Elle donne la correspondance entre une variable AUTOMGEN et une variable automate.

La syntaxe est :

variable AUTOMGEN = variable automate

Exemple :

#i10=i0,2 ; l'entrée 10 d'AUTOMGEN est attribuée à l'entrée
; PL71 0,2

La déclaration de table linéaire

Elle donne la correspondance entre une série de variables AUTOMGEN et une série de variables AUTOMATE.

La syntaxe est :

longueur de la table : première variable AUTOMGEN = première variable automate

Exemple :

##16:o0=o0,0 ; les sorties AUTOMGEN 0 à 15 sont attribuées aux sorties PL71
; 0,0 à 0,15

La déclaration de table pour affectation automatique

Elle laisse le soin au générateur de code d'affecter un ou plusieurs types de variables dans une table de variables automate. Le générateur de code affecte automatiquement les variables suivant ses besoins et si la table n'est pas saturée.

La syntaxe est :

type(s) de variable AUTOMGEN = première variable PL71: numéro de la dernière variable PL71

Un type est représenté par un nom de variable sans numéro. Si plusieurs types sont à préciser, alors ils doivent être séparés par le caractère « & ».

Exemples :

###x&bx=b0:199 ; affecte automatiquement les bits d'étapes d'AUTOMGEN dans les
; bits PL71 0 à 199

Ce type de déclaration est moins prioritaire que les types « # » et « ## ».

Plusieurs déclarations de ce type peuvent être utilisées pour un même type de variable AUTOMGEN. Dans ce cas, le générateur de code remplira dans l'ordre de déclaration et suivant ses besoins, une ou plusieurs tables.

Les types de variables AUTOMGEN

Les types de variables AUTOMGEN sont un sur-ensemble de ceux décrits dans le manuel d'utilisation du module principal dans la partie « Compileur ».

En voici la liste exhaustive :

x	étape, état actuel
bx	étape, état futur
b	bit, état actuel
bb	bit, état futur
u	bit utilisateur, état actuel
Bu	bit utilisateur, état futur

I	entrée, état actuel
bi	entrée, état futur
O	sortie, état actuel
bo	sortie, état futur
T	temporisation, état de fin
bt	temporisation, état de lancement
Tempo	numéro de temporisation

Pour les variables booléennes d'AUTOMGEN, deux variables booléennes PL71 sont utilisées afin de gérer les états dans le temps.

Le fichier modèle « PL71.VUS »

Détail du fichier « PL71.VUS » présent dans le répertoire où est installé AUTOMGEN.

; Fichier .VUS pour PL71 - AUTOMGEN V6

; E/S pour les TSX 17-20 et 17-10 (module de base)

##32:bi0=i0,0

##16:o0=o0,0

; 32 temporisations

##32:tempo=0

; bits : b2 ... b255

###i&bo&x&bx&bb&bu&b&u&t&bt&ux&dx&ui&di&uo&do&du&uu&ub&db&tt&ut&dt=b2:255

; bits : x1 ... x62

###i&bo&x&bx&bb&bu&b&u&t&bt&ux&dx&ui&di&uo&do&du&uu&ub&db&tt&ut&dt=x1:62

; 15 compteurs

##15:c0=c0

Définition des E/S

Définitions des temporisations

Définitions des autres variables booléennes

Définitions des compteurs

Les fichiers « .SRT » et « .END »

Ces fichiers permettent d'insérer du code PL71 au sein d'une application AUTOMGEN.

Le code contenu dans le fichier « .SRT » est exécuté à chaque cycle, avant le code généré par la compilation de l'application issue d'AUTOMGEN.

Le code contenu dans le fichier « .END » est exécuté à chaque cycle, après le code généré par la compilation de l'application issue d'AUTOMGEN.

Le fichier « DEFPL71.SRT » est fourni en modèle dans le répertoire où est installé AUTOMGEN. L'état du bit 0 d'AUTOMGEN est géré dans ce fichier, il doit obligatoirement être à l'état 1 au premier cycle de scrutation de l'application.

La syntaxe utilisée dans ces deux fichiers est presque en tout points semblable à celle du logiciel PL71. Les compléments de syntaxe suivants sont à prendre en compte :

- il est possible de faire référence aux variables AUTOMGEN en utilisant la syntaxe suivante : « *_variable AUTOMGEN_* ». Par exemple :

```
L      _i0_
=      _o10_
...
```

- pour les sauts, il faut utiliser la syntaxe suivante : « @LAB nom de label » marque la destination,
« JMP *_nom de label_* » marque le saut, Exemple :

```
L      i0,0
JMP   _fin_
...
@LAB fin
```

- Dans les opérations où interviennent des constantes telles que « **L C03 K00123** » la lettre '**K**' doit toujours apparaître avant la constante, ce qui n'est pas le cas lors de la saisie avec le logiciel PL71.

L'examen du fichier portant l'extension « .T71 » permet de comprendre l'utilisation de ces différentes syntaxes.

CHAPITRE N°6.

Paramétrage de l'automate

Pour paramétrer l'automate, il faut modifier le fichier texte « .VSY » de l'application. Le fichier « DEFPL71.VSY » est utilisé s'il n'y a pas de fichier « .VSY » associé à l'application. Des directives de compilation conditionnelle permettent de sélectionner des sections de programme qui seront compilées en fonction du type d'automate. La syntaxe du début de condition est : « \$IF TSX=type d'automate » et se termine par « \$ENDIF ».

Directive de sélection du type d'automate

\$TSX=

Valeurs possibles : 1710 ou 1720.

Directive de sélection de la taille mémoire

\$RAM=

Valeurs possibles : 8 pour le TSX 17-10, 8 ou 24 pour le TSX 17-20.

Directive de paramétrage des modules d'extension pour le TSX 17-20

\$EXTn=

n représente le numéro de l'extension de 1 à 2.

Valeurs possibles : 0 s'il n'y a pas de module, 1 si c'est un module local et 2 si c'est un module réseau.

Directives de paramétrage des temporisations

Les directives de paramétrage de la base de temps et de la prédisposition des temporisations sont générées automatiquement par le post-processeur.

\$TPREDn=

Valeur de présélection.

n représente le numéro de la temporisation (0 à 31)

Valeurs possibles : 0 à 9999

\$TTBn=

Base de temps.

n représente le numéro de la temporisation (0 à 31)

Valeurs possibles : 10MS, 100MS, 1S, 1MIN.

Directives de paramétrage des compteurs

\$CPREDn=

Valeur de présélection.

n représente le numéro du compteur (0 à 15)

Valeurs possibles : 0 à 9999

Directives spécifiques au TSX 17-20

\$IO,0=

Type de l'entrée i0,0.

Valeurs possibles : NORMAL (entrée normale) ou RUN/STOP (entrée RUN/STOP).

\$OO,0=

Type de la sortie o0,0.

Valeurs possibles : NORMAL (sortie normale) ou SECURITY (sortie de sécurité).

\$IO,24=

Type de l'entrée i0,24.

Valeurs possibles : NORMAL (entrée normale) ou EVENT (entrée événementielle).

\$IO,25=

Type de l'entrée i0,25.

Valeurs possibles : NORMAL (entrée normale) ou EVENT (entrée événementielle).

\$FC=

Sélection du mode pour le compteur rapide.

Valeurs possibles : C (compteur rapide), T (temporisateur rapide).

\$IFT=

Sélection du groupe d'entrées affectées à la tâche rapide.

Valeurs possibles : NO (pas d'entrées) IO,0-IO,7, IO,8-IO,15, IO,16-IO,23

\$OFT=

Sélection du groupe de sorties affectées à la tâche rapide.

Valeurs possibles : NO (pas de sorties) OO,0-OO,7, OO,8-OO,15

CHAPITRE N°7.

Le module de dialogue

« WPL71DIA.EXE » est le module de dialogue appelé par l'environnement 16 bits pour le téléchargement et la comparaison des applications, la visualisation dynamique, le changement d'état des variables et le changement d'état des modes de marche.

Le mode pas à pas n'est pas supporté par le module de dialogue.

Les types de variables PL71 suivants sont visualisables : bits, bits Système, bits X, consignes et compteurs de temporisation. Les entrées et les sorties sont visualisables et modifiables.

Le forçages des variables booléennes I et O est utilisable à partir de la version 5.80 de l'environnement AUTOMGEN.

Le module de dialogue utilise un fichier généré par le générateur de code et qui porte l'extension « .DBG ». Ce fichier donne la correspondance entre les variables AUTOMGEN de l'application et les variables automate.

Paramétrage

Par défaut le module de dialogue utilise le port COM1 en mode console.

Des paramètres peuvent être ajoutés à la suite du nom du module de communication sur la troisième ligne de définition de la cible dans l'environnement AUTOMGEN. Les paramètres doivent être écrits à la suite d'un espace. Aucun espace ne doit ensuite apparaître, les paramètres sont écrits à la suite les uns des autres.

Modification du port de communication

Le paramètre « -C2 » permet d'utiliser le port COM2 du PC.

CHAPITRE N°8.

Techniques avancées

Ce chapitre décrit des possibilités du post-processeur qui permettent de créer des applications plus performantes.

Insertion de code PL71 dans les applications

Comme nous l'avons vu au chapitre 6, les fichiers « .SRT » et « .END » associés à une application peuvent recevoir des lignes de code en langage PL71.

Une autre technique peut être utilisée pour insérer du code PL71 dans une boîte de code AUTOMGEN.

Les directives « #BEGIN_MACHINE_CODE » et « #END_MACHINE_CODE » permettent de définir une section de code constructeur.

Cette technique permet d'utiliser des opérations du langage PL71, telles que XO, CU, CD, etc..

Exemple (activation d'une sortie):

100	#begin_machine_code
	L I0,4
	XO I2,3
	= O0,7
	#end_machine_code

Utilisation de la tâche rapide

La directive « #I1 » placée en commentaire sur un folio permet d'associer le code littéral ou le code PL71 écrit sur le folio à la tâche rapide. Le folio ne peut contenir que du code littéral bas niveau ou du code PL71 écrit dans un rectangle d'organigramme.

CHAPITRE N°9.

Limites de compatibilité

Ce chapitre détaille les limites d'utilisation du post-processeur PL71 par rapport aux possibilités d'AUTOMGEN et des automates TSX.

- l'adressage indirect n'est pas utilisable,
- les sous-programmes ne sont pas utilisables,
- les traitements numériques ne sont pas utilisables,
- les incrémentations et décrémentations de compteurs ne sont utilisables qu'à travers du code en langage constructeur PL71.

Si le générateur de code PL71 rencontre une forme de programme intraduisible, alors un message d'erreur de type « code/adressage non supporté dans le fichier C:\AUTOMV6\TEST.EQU à l'adresse XXXX » est généré. XXXX représente une adresse dans le fichier « .EQU » Si vous souhaitez connaître avec précision l'instruction qui a généré cette erreur, utilisez l'utilitaire « CODELIST.EXE⁵ » pour obtenir un listing du fichier « .EQU ».

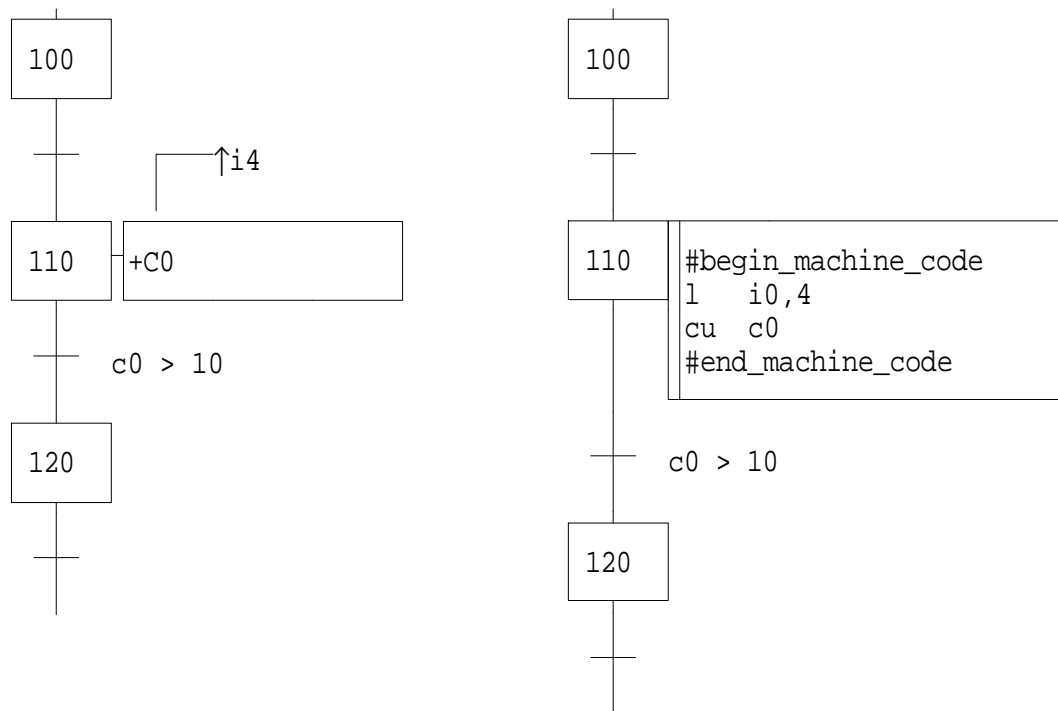
⁵le chapitre « Techniques avancées » du manuel de l'utilisateur associé au module principal d'AUTOMGEN décrit l'utilisation de l'utilitaire « CODELIST.EXE »

CHAPITRE N°10.

Utilisation des compteurs

Les incrémentations et décréments de compteurs en PL71 étant limitées (sur front montant uniquement) par rapport aux possibilités d'AUTOMGEN et des automates TSX il est nécessaire d'utiliser du code en langage constructeur si l'on veut les utiliser.

Exemple :



Ces deux graficets sont équivalents mais le premier n'est pas permis en PL71. Il faut utiliser du code constructeur.

Le second exemple se trouve dans le sous-répertoire « EXPL71 » du répertoire où est installé AUTOMGEN. Il porte le nom « EXCPT ».

CHAPITRE N°11.

Utilisation du compteur rapide sur un automate TSX 17-20

Exemple : le but est de compter 200 impulsions sur le compteur rapide. La sortie O5 sera activée par la tâche rapide en fin de comptage.

1- Paramétrage du compteur rapide dans le fichier « .VSX » de l'application (le fichier DEFPL71.VSX si l'application ne possède pas son propre fichier). Dans la section consacrée à l'automate TSX 17-20 le paramétrage suivant est à écrire :

\$FC=C

et

\$CPRED15=200

2- Programmation

Le programme est composé de deux folios « EXFC.GR7 » pour le programme principal et « EXFC2.GR7 » pour la tâche rapide.

Programme principal :

```
// Exemple d'utilisation du comptage rapide
```

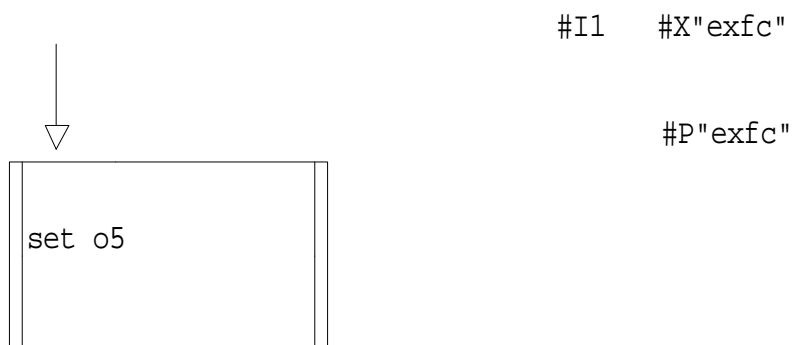
```
// sur un automate TSX 17-20
```

```
#L"EXFC2"
```

0	<pre>#begin_machine_code ; Désactiver le bit sy19 L sy19 R sy19 ; RAZ du bit sy15 L sy15 R sy15 #end_machine_code</pre>
---	--

Tâche rapide :

```
// Tâche rapide de l'exemple EXFC
```



Cet exemple se trouve dans le sous-répertoire « EXPL71 » du répertoire où est installé AUTOMGEN. Il porte les noms « EXFC » pour le programme principal et « EXFC2 » pour la tâche rapide.

CHAPITRE N°12.

Exemple complet

Ce chapitre contient le dossier de documentation de l'exemple « PL71 ».

Cet exemple est présent dans le sous-répertoire « EXPL71 » du répertoire où est installé AUTOMGEN.

PL71



I R A I

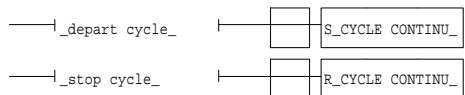
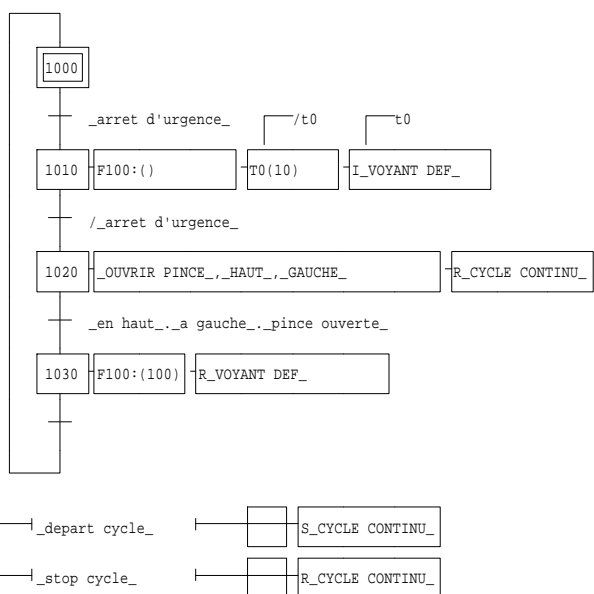
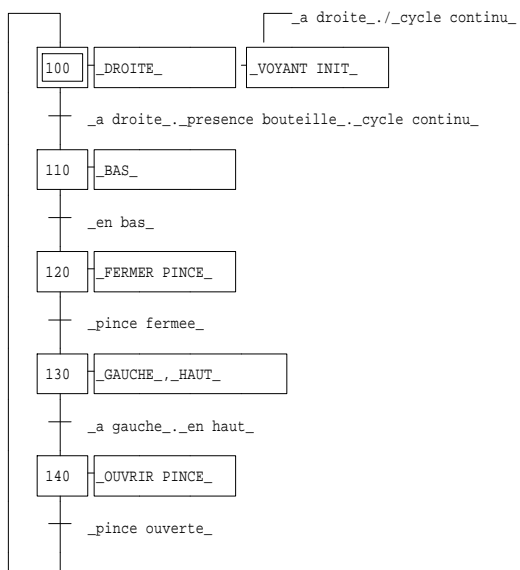
BP 14 - 30110 LA GRAND COMBE

PL71

Cible : TSX 17-10 ou TSX 17-20

Date de création : 11/08/1999 14:48 50

/* Programme d'exemple pour PL71*/



AUTOMGEN	API N°1	API N°2	SYMBOLE	COMMENTAIRE OU NOM DES FOLIOS OU LA VARIABLE EST UTILISEE
I0000		I0,0	a droite	capteur verin 1 sorti
I0001		I0,1	a gauche	capteur verin 1 rentre
I0002		I0,2	en bas	capteur verin 2 sorti
I0003		I0,3	en haut	capteur verin 2 rentre
I0004		I0,4	pince fermee	capteur pince fermee
I0005		I0,5	pince ouverte	capteur pince ouverte
I0006		I0,6	presence bouteille	capteur bouteille en attente
I0007		I0,7	depart cycle	bp depart de cycle
I0008		I0,8	stop cycle	bp stop
I0009		I0,9	arret d'urgence	bp arret d'urgence
O0000	O0,0		DROITE	sortir le verin 1
O0001	O0,1		GAUCHE	rentrer le verin 1
O0002	O0,2		BAS	sortir le verin 2
O0003	O0,3		HAUT	rentrer le verin2
O0004	O0,4		FERMER PINCE	fermer la pince
O0005	O0,5		OUVRIR PINCE	ouvrir la pince
O0006	O0,6		VOYANT INIT	voyant vert
O0007	O0,7		VOYANT DEF	voyant rouge
T0000	B26	B34		
U0101	B20	B21		
U1000	B20	B21	cycle continu	etat cycle continu
X0100	B2	B3		
X0110	B4	B5		
X0120	B6	B7		
X0130	B8	B9		
X0140	B10	B11		
X1000	B12	B13		
X1010	B14	B15		
X1020	B16	B17		
X1030	B18	B19		
	B18	B19		

Fichier « PL71.T71 »

```
; ***** Variables systèmes ***** VSY
; pas de fichier PL71.VSY pour l'application, utilisation du fichier C:\AUTOMV6\defpl71.VSY
; Fichier .VSY pour PL71 - AUTOMGEN V6

; Paramètres de l'UC
$TSX=1720 ; Modifier suivant le type d'automate : 1710 ou 1720
$RAM=8 ; Modifier suivant la taille de la RAM : 8 ou 24 (Ko)

; Paramétrage des modules d'extension
$EXT1=0 ; module 1 : 0=pas de module, 1=local, 2=reseau
$EXT2=0 ; module 2 : 0=pas de module, 1=local, 2=reseau

; Configuration pour le TSX 17-20
$IF TSX=1720
$I0,0=NORMAL ; entrée I0,0 : NORMAL ou RUN/STOP
$O0,0=NORMAL ; sortie O0,0 : NORMAL ou SECURITY
$I0,24=NORMAL ; entrée I0,24 : NORMAL ou EVENT
$I0,25=NORMAL ; entrée I0,24 : NORMAL ou EVENT
$FC=C ; FC : C ou T
$CPRED15=200 ; prédisposition compteur rapide
$IFT=NO ; entrées pour la tâche rapide NO ou I0,0-I0,7 ou I0,8-I0,15 ou I0,16-I0,23
$OFT=NO ; sorties pour la tâche rapide NO ou O0,0-O0,7 ou O0,8-O0,15
$ENDIF

#tmp=b0
#tr=b1

; ***** Variables utilisateur ***** VUS
; pas de fichier PL71.VUS pour l'application, utilisation du fichier C:\AUTOMV6\defpl71.VUS
; Fichier .VUS pour PL71 - AUTOMGEN V6

; E/S pour les TSX 17-20 et 17-10 (module de base)
##32:bi0=i0,0
##16:o0=o0,0

; 32 temporisations
##32:tempo=0

; bits : b2 à b255
###i&bo&x&bx&bb&bu&b&u&t&bt&ux&dx&ui&di&uo&do&du&uu&ub&db&tt&ut&dt=b2:255
; bits : x1 à x62
###i&bo&x&bx&bb&bu&b&u&t&bt&ux&dx&ui&di&uo&do&du&uu&ub&db&tt&ut&dt=x1:62

; 15 compteurs
##15:c0=c0

; ***** Correspondance bits/étapes
#b100=_x100_
#bb100=_bx100_
#b103=_x110_
#bb103=_bx110_
#b105=_x120_
#bb105=_bx120_
#b107=_x130_
#bb107=_bx130_
#b109=_x140_
#bb109=_bx140_
#b102=_x1000_
```

```

#bb102=_bx1000_
#b104=_x1010_
#bb104=_bx1010_
#b106=_x1020_
#bb106=_bx1020_
#b108=_x1030_
#bb108=_bx1030_

; ***** Correspondance bits/bits utilisateur
#b101=_u1000_
#bb101=_bu1000_
; ***** Programme de démarrage ***** SRT
; pas de fichier PL71.SRT pour l'application, utilisation du fichier C:\AUTOMV6\defpl71.SRT
; Fichier .SRT pour PL71 - AUTOMGEN V6
L sy0
O sy1
O sy2
= _b0_
L sy2
R sy2
L sy0
R sy0
LN _tr_
S _tr_
; ***** Prédispositions
$TPRED_tempo0_=10
; ***** Code généré par le post-processeur *****

; ***** Module : C:\AUTOMV6\EXPL71\PL71.GR7
L _bi0_
A _bi6_
A _u1000_
R _bx100_
L _b0_
S _bx100_
L _x140_
A _bi5_
S _bx100_
L _bi9_
R _bx1000_
L _b0_
S _bx1000_
L _x1030_
S _bx1000_
L _bi2_
R _bx110_
L _x100_
A _bi0_
A _bi6_
A _u1000_
S _bx110_
LN _bi9_
R _bx1010_
L _x1000_
A _bi9_
S _bx1010_
L _bi4_
R _bx120_
L _x110_
A _bi2_

```



```

S _bx120_
L _bi3_
A _bi1_
A _bi5_
R _bx1020_
L _x1010_
AN _bi9_
S _bx1020_
L _bi1_
A _bi3_
R _bx130_
L _x120_
A _bi4_
S _bx130_
L _tr_
R _bx1030_
L _x1020_
A _bi3_
A _bi1_
A _bi5_
S _bx1030_
L _bi5_
R _bx140_
L _x130_
A _bi1_
A _bi3_
S _bx140_
L _bi7_
= _b110_
L _bi8_
= _b112_
L _bi0_
AN _u1000_
= _b115_
LN _t0_
= _b118_
L _t0_
= _b120_
LN _b116_
JMP _line75_
L _tr_
R _bx130_
L _tr_
R _bx120_
L _tr_
R _bx110_
L _tr_
R _bx100_
L _tr_
R _bx140_
@LAB line75
LN _b121_
JMP _line82_
L _tr_
R _bx130_
L _tr_
R _bx120_
L _tr_
R _bx110_
L _tr_

```

```

S _bx100_
L _tr_
R _bx140_
@LAB line82
; ***** Module : (ACTION)
L _b115_
A _bx100_
= _b114_
L _b118_
A _bx1010_
= _b117_
L _b120_
A _bx1010_
= _b119_
L _bx100_
= _o0_
L _b114_
= _o6_
L _bx110_
= _o2_
L _bx1010_
= _b116_
L _b117_
= _bt0_
L _b119_
= _tmp_
LN _tmp_
JMP _label1_
LN _o7_
= _o7_
@LAB label1
L _bx120_
= _o4_
L _bx1020_
O _bx140_
= _o5_
L _bx1020_
O _bx130_
= _o3_
L _bx1020_
O _bx130_
= _o1_
L _bx1020_
O _b112_
R _bu1000_
L _bx1030_
= _b121_
L _bx1030_
R _o7_
L _b110_
S _bu1000_
; ***** Evolution des bits d'étapes
L _bx100_
= _x100_
L _bx110_
= _x110_
L _bx120_
= _x120_
L _bx130_
= _x130_

```

```

L _bx140_
= _x140_
L _bx1000_
= _x1000_
L _bx1010_
= _x1010_
L _bx1020_
= _x1020_
L _bx1030_
= _x1030_
;***** Evolution des bits utilisateurs
L _bu1000_
= _u1000_
;***** Evolution des temporisations
L _bt0_
= T_tempo0_
L T_tempo0_
= _t0_
;***** Programme de fin          ***** END
; pas de fichier PL71.END pour l'application, utilisation du fichier C:\AUTOMV6\defpl71.END
; Fichier .END pour PL71 - AUTOMGEN V6

; (néant)

```

Fichier « PL71.LAP »

```

8300 : L1 L SY0
8308 : L2 O SY1
8310 : L3 O SY2
8318 : L4 = B22
8320 : L5 L SY2
8328 : L6 R SY2
8330 : L7 L SY0
8338 : L8 R SY0
8340 : L9 LN B1
8348 : L10 S B1

; MODULE : C:\AUTOMV6\EXPL71\PL71.GR7

8350 : L11 L I0,0
8358 : L12 A I0,6
8360 : L13 A B20
8368 : L14 R B3
8370 : L15 L B22
8378 : L16 S B3
8380 : L17 L B10
8388 : L18 A I0,5
8390 : L19 S B3
8398 : L20 L I0,9
83A0 : L21 R B13
83A8 : L22 L B22
83B0 : L23 S B13
83B8 : L24 L B18
83C0 : L25 S B13
83C8 : L26 L I0,2
83D0 : L27 R B5
83D8 : L28 L B2
83E0 : L29 A I0,0
83E8 : L30 A I0,6
83F0 : L31 A B20

```

83F8 : L32 S B5
 8400 : L33 LN I0,9
 8408 : L34 R B15
 8410 : L35 L B12
 8418 : L36 A I0,9
 8420 : L37 S B15
 8428 : L38 L I0,4
 8430 : L39 R B7
 8438 : L40 L B4
 8440 : L41 A I0,2
 8448 : L42 S B7
 8450 : L43 L I0,3
 8458 : L44 A I0,1
 8460 : L45 A I0,5
 8468 : L46 R B17
 8470 : L47 L B14
 8478 : L48 AN I0,9
 8480 : L49 S B17
 8488 : L50 L I0,1
 8490 : L51 A I0,3
 8498 : L52 R B9
 84A0 : L53 L B6
 84A8 : L54 A I0,4
 84B0 : L55 S B9
 84B8 : L56 L B1
 84C0 : L57 R B19
 84C8 : L58 L B16
 84D0 : L59 A I0,3
 84D8 : L60 A I0,1
 84E0 : L61 A I0,5
 84E8 : L62 S B19
 84F0 : L63 L I0,5
 84F8 : L64 R B11
 8500 : L65 L B8
 8508 : L66 A I0,1
 8510 : L67 A I0,3
 8518 : L68 S B11
 8520 : L69 L I0,7
 8528 : L70 = B23
 8530 : L71 L I0,8
 8538 : L72 = B24
 8540 : L73 L I0,0
 8548 : L74 AN B20
 8550 : L75 = B25
 8558 : L76 LN B26
 8560 : L77 = B27
 8568 : L78 L B26
 8570 : L79 = B28
 8578 : L80 LN B29
 8580 : L81 JMP 1
 8588 : L82 L B1
 8590 : L83 R B9
 8598 : L84 L B1
 85A0 : L85 R B7
 85A8 : L86 L B1
 85B0 : L87 R B5
 85B8 : L88 L B1
 85C0 : L89 R B3
 85C8 : L90 L B1
 85D0 : L91 R B11

85D8 : L92 LAB 1
 85E0 : L93 LN B30
 85E8 : L94 JMP 2
 85F0 : L95 L B1
 85F8 : L96 R B9
 8600 : L97 L B1
 8608 : L98 R B7
 8610 : L99 L B1
 8618 : L100 R B5
 8620 : L101 L B1
 8628 : L102 S B3
 8630 : L103 L B1
 8638 : L104 R B11
 8640 : L105 LAB 2

; MODULE : (ACTION)

8648 : L106 L B25
 8650 : L107 A B3
 8658 : L108 = B31
 8660 : L109 L B27
 8668 : L110 A B15
 8670 : L111 = B32
 8678 : L112 L B28
 8680 : L113 A B15
 8688 : L114 = B33
 8690 : L115 L B3
 8698 : L116 = O0,0
 86A0 : L117 L B31
 86A8 : L118 = O0,6
 86B0 : L119 L B5
 86B8 : L120 = O0,2
 86C0 : L121 L B15
 86C8 : L122 = B29
 86D0 : L123 L B32
 86D8 : L124 = B34
 86E0 : L125 L B33
 86E8 : L126 = B0
 86F0 : L127 LN B0
 86F8 : L128 JMP 3
 8700 : L129 LN O0,7
 8708 : L130 = O0,7
 8710 : L131 LAB 3
 8718 : L132 L B7
 8720 : L133 = O0,4
 8728 : L134 L B17
 8730 : L135 O B11
 8738 : L136 = O0,5
 8740 : L137 L B17
 8748 : L138 O B9
 8750 : L139 = O0,3
 8758 : L140 L B17
 8760 : L141 O B9
 8768 : L142 = O0,1
 8770 : L143 L B17
 8778 : L144 O B24
 8780 : L145 R B21
 8788 : L146 L B19
 8790 : L147 = B30
 8798 : L148 L B19

87A0 : L149 R O0,7
87A8 : L150 L B23
87B0 : L151 S B21
87B8 : L152 L B3
87C0 : L153 = B2
87C8 : L154 L B5
87D0 : L155 = B4
87D8 : L156 L B7
87E0 : L157 = B6
87E8 : L158 L B9
87F0 : L159 = B8
87F8 : L160 L B11
8800 : L161 = B10
8808 : L162 L B13
8810 : L163 = B12
8818 : L164 L B15
8820 : L165 = B14
8828 : L166 L B17
8830 : L167 = B16
8838 : L168 L B19
8840 : L169 = B18
8848 : L170 L B21
8850 : L171 = B20
8858 : L172 L B34
8860 : L173 = T0
8868 : L174 L T0
8870 : L175 = B26