



Manuel de référence post-processeurs
mise à jour numéro 4

GENERALITES	7
CONFIGURATION	8
LES FICHIERS DE CONFIGURATION	8
<i>Système</i>	8
<i>Correspondances de variables</i>	8
<i>Code constructeur démarrage</i>	8
<i>Code constructeur fin</i>	8
CONFIGURATION PAR DEFAUT	8
<i>Modifier les déclarations par défaut</i>	9
<i>Utiliser les déclarations par défaut</i>	9
VISUALISER ET MODIFIER LES ELEMENTS DE CONFIGURATION	9
SYSTEME	9
<i>Configuration matérielle</i>	9
<i>Configuration logicielle</i>	9
<i>Options de génération de code</i>	9
<i>Déclarations de variables</i>	10
<i>Autres éléments</i>	10
<i>Voir l'élément « Système » sous forme de textes</i>	10
<i>Afficher les éléments système</i>	11
CORRESPONDANCES DE VARIABLES	11
<i>L'affectation unitaire</i>	12
<i>L'affectation linéaire</i>	12
<i>L'affectation automatique</i>	12
<i>Types de variables supprimer AUTOMGEN</i>	13
<i>Afficher les éléments de correspondances de variables</i>	15
<i>Modifier un élément de correspondance de variables</i>	17
<i>Ajouter un élément de correspondance de variables</i>	17
<i>Supprimer un élément de correspondances de variable</i>	19
<i>Associer un bit d'AUTOMGEN à un bit système d'une cible</i>	19
<i>Associer une table de mots d'AUTOMGEN à une table de mots fixes de la cible</i>	20
<i>Associer des mots d'AUTOMGEN à des entrées ou des sorties analogiques d'une cible</i>	20
<i>Associer une table de bits d'AUTOMGEN à une table de bits d'une cible</i>	21
<i>Voir les correspondances de variables sous forme de textes</i>	21
CODE CONSTRUCTEUR DEMARRAGE, CODE CONSTRUCTEUR DE FIN	22
<i>Référence à une variable AUTOMGEN</i>	22
<i>Référence à un symbole de l'application AUTOMGEN</i>	22
<i>Définition et référence à un label</i>	22
INSERER DU CODE CONSTRUCTEUR DANS UNE APPLICATION	22
CHOIX DES OPTIONS DE CONNEXION	22
CHOIX D'UN MODE DE CONNEXION	23
PARAMETRAGE DU MODULE DE COMMUNICATION	23
POST-PROCESSEUR PL7	24
MODULE DE COMMUNICATION	24
MODE DE GENERATION D'UN FICHIER EXECUTABLE.....	25
<i>Mode de génération directe du fichier binaire</i>	25
<i>Mode de génération d'un fichier « .FEF »</i>	27
UTILISATION DE TACHES D'INTERRUPTIONS.....	29
EXEMPLES SPECIFIQUES	29
<i>Entrées / sorties analogiques</i>	29
<i>Compteur rapide TSX 37-10</i>	29
<i>Compteur rapide TSX 37-10 utilisé en décomptage</i>	29
<i>Compteur rapide TSX 37-22</i>	29
<i>ASI</i>	30
<i>MAGELIS</i>	30
POST-PROCESSEUR PL72	31
CHOIX DU TYPE DE L' AUTOMATE	31
ELEMENTS SYNTAXIQUES SPECIFIQUES	31

<i>Appel des blocs fonction PL72</i>	31
<i>Utilisation de la tâche rapide</i>	33
MODULE DE COMMUNICATION	33
EXEMPLES SPECIFIQUES	33
<i>Entrées / sorties analogiques</i>	34
<i>Compteur rapide</i>	34
<i>Blocs texte et xbt</i>	35
<i>Blocs texte et UNITELWAY</i>	37
POST-PROCESSEUR S7200	40
CHOIX DU TYPE DE CPU	40
MODULE DE COMMUNICATION	40
EXEMPLE SPECIFIQUE.....	40
POST-PROCESSEUR ABB	41
CHOIX DU TYPE D' AUTOMATE.....	41
<i>Automate AC31</i>	41
<i>Automate CS31</i>	41
MODULE DE COMMUNICATION	41
UTILITAIRE	41
EXEMPLES SPECIFIQUES	41
<i>Entrées / sorties analogiques</i>	41
<i>Interruptions</i>	41
POST-PROCESSEUR GE-FANUC / ALSPA	42
CHOIX DU TYPE D' AUTOMATE.....	42
MODULE DE COMMUNICATION	42
UTILITAIRE	42
POST-PROCESSEUR STEP5	43
MODULE DE COMMUNICATION	43
STRUCTURE DE L' APPLICATION.....	43
<i>Choix des blocs de programmes à utiliser</i>	45
<i>Choix du bloc de données</i>	45
CHOIX DU TYPE DE PROCESSEUR	46
ASSOCIATION DU CODE ECRIT SUR UN FOLIO A UN BLOC PROGRAMME.....	46
SYNTAXES SPECIFIQUES	46
<i>Définition de blocs</i>	46
POST-PROCESSEUR TSX 07	49
MODULE DE COMMUNICATION	49
POST-PROCESSEUR PS3-PS4	50
MODULE DE COMMUNICATION	50
POST-PROCESSEUR PS4	51
MODULE DE COMMUNICATION	51
TRANSFERT DES PROGRAMMES VERS LE LOGICIEL SUCOSOFT S40 DE MOELLER.....	51
<i>Démarche à suivre pour importer le fichier généré par AUTOMGEN dans le logiciel MOELLER puis l'injecter dans l'automate</i>	52
POST-PROCESSEUR RPX	56
CHOIX DU TYPE D' AUTOMATE.....	56
MODULE DE COMMUNICATION	56
UTILITAIRE	56
POST-PROCESSEUR PL71	57
CHOIX DU TYPE D' AUTOMATE.....	57
MODULE DE COMMUNICATION	57
TACHE COMPTEUR RAPIDE	57

EXEMPLES SPECIFIQUES	57
<i>Comptage</i>	57
<i>Compteur rapide</i>	58
POST-PROCESSEUR PB	59
CHOIX DU TYPE D' AUTOMATE.....	59
MODULE DE COMMUNICATION	59
SYNTAXES SPECIFIQUES	59
POST-PROCESSEUR SMC	61
CHOIX DU TYPE D' AUTOMATE.....	61
MODULE DE COMMUNICATION	61
SYNTAXES SPECIFIQUES	61
POST-PROCESSEUR S7300	62
MODULE DE COMMUNICATION	62
SYNTAXES SPECIFIQUES	62
<i>Définition des variables d' un bloc</i>	63
<i>Appel des blocs</i>	63
IMPORTATION DANS LE LOGICIEL SIMATIC DE SIEMENS	64
STRUCTURE DU CODE GENERE.....	66
<i>Choix des blocs de programmes à utiliser</i>	68
ASSOCIATION DU CODE ECRIT SUR UN FOLIO A UN BLOC PROGRAMME.....	68
EXEMPLES SPECIFIQUES	68
<i>Appel d' un bloc STEP7</i>	68
<i>Utilisation d' un bloc OB</i>	68
POST-PROCESSEUR OMRON	69
CHOIX DU TYPE D' AUTOMATE.....	69
MODULE DE COMMUNICATION	69
TRANSFERT DES APPLICATIONS DANS LE LOGICIEL CX-PROGRAMMER	69
SYNTAXE SPECIFIQUE.....	71
ASSOCIATION DU CODE ECRIT SUR UN FOLIO A UN BLOC PROGRAMME.....	72
EXEMPLE SPECIFIQUE.....	72
POST-PROCESSEUR ALSPA	73
MODULE DE COMMUNICATION	73
POST-PROCESSEUR ZELIO	74
MODULE DE COMMUNICATION	74
POST-PROCESSEUR FESTO	75
MODULE DE COMMUNICATION	75
GENERATION DE FICHIER BINAIRE.....	75
IMPORTATION DANS LES ATELIERS LOGICIEL FESTO.....	75
POST-PROCESSEUR ALLEN-BRADLEY	77
MODULE DE COMMUNICATION	77
TRANSFERT DES PROGRAMMES VERS LE LOGICIEL RS-LOGIX 500 DE ROCKWELL SOFTWARE.....	77
POST-PROCESSEUR MITSUBISHI	79
CHOIX DU TYPE D' AUTOMATE.....	79
MODULE DE COMMUNICATION	79
TRANSFERT DES PROGRAMMES VERS LE LOGICIEL FX-WIN DE MITSUBISHI.....	79
TRANSFERT DES PROGRAMMES VERS LE LOGICIEL GX-DEVELOPPER DE MITSUBISHI	80
POST-PROCESSEUR TWIDO	81
CHOIX DE LA CONFIGURATION DE L' AUTOMATE	81
MODULE DE COMMUNICATION	81

POST-PROCESSEUR ZELIO 2	82
INITIALISATION DE L' AUTOMATE	82
CONFIGURATION DE L' AUTOMATE	82
MODULE DE COMMUNICATION	82

Généralités

Les post-processeurs sont des modules logiciels permettant de traduire les fichiers de code pivot générés par le compilateur AUTOMGEN en fichiers exécutables sur une cible ainsi que d'assurer la connexion dynamique à la cible.

Le mot « Cible » désigne de façon générique un système programmable capable d'exécuter une application.

Un post-processeur d'AUTOMGEN permet de programmer un type ou un ensemble de type de cibles (généralement une famille d'automates partageant le même langage de programmation est programmable avec le même post-processeur dans AUTOMGEN).

Ce manuel contient en première partie des notions fondamentales communes à tous les post-processeurs. Viennent ensuite des informations spécifiques aux implémentations faites par chaque post-processeur.

Configuration

Nous vous invitons à apporter la plus grande attention aux explications de ce chapitre.

Les fichiers de configuration

Quatre éléments de configurations sont utilisés par chaque post-processeur. Chacun d'eux est utilisé de façon spécifique.

Système

Contient la configuration matérielle de la cible, la configuration logicielle, des options permettant de modifier la façon dont le post-processeur génère le code ainsi que des déclarations de variables réservées (pour l'usage interne du post-processeur). Généralement, vous serez amenés, suivant la cible, à modifier la configuration matérielle contenue dans cet élément (par exemple un type d'UC ou une configuration de type de cartes d'entrées / sorties).

Correspondances de variables

La maîtrise de la correspondance des variables est un des éléments essentiels à la maîtrise de l'utilisation des post-processeurs.

Lorsque le post-processeur traduit un fichier du langage pivot d'AUTOMGEN vers un langage cible spécifique, il doit attribuer les variables d'AUTOMGEN à des variables de la cible.

Cet élément contient la description précise de l'attribution des variables. En modifiant cet élément, vous avez le contrôle total sur l'utilisation de l'espace des variables de la cible.

Code constructeur démarrage

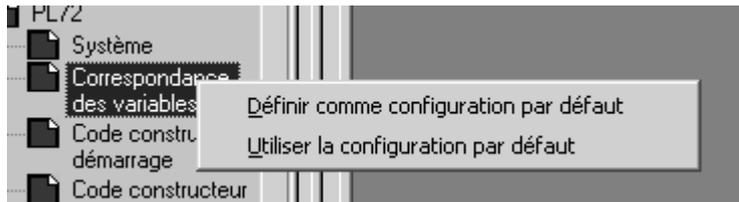
Cet élément contient du langage machine propre à la cible qui sera placé au début du code exécutable généré par le post-processeur (exécuté au début du cycle).

Code constructeur fin

Cet élément contient du langage machine propre à la cible qui sera placé à la fin du code exécutable généré par le post-processeur (exécuté à la fin du cycle).

Configuration par défaut

A la création du projet, des éléments de configuration par défaut sont dupliqués dans le projet. Les modifications apportées aux éléments de configuration du projet n'affecteront pas les déclarations par défaut.



Modifier les déclarations par défaut

Cliquez avec le bouton droit de la souris sur l'élément « Configuration / Post-processeur / <nom de la cible> / ... » et choisissez « Définir comme configuration par défaut ». L'élément de configuration du projet est alors défini comme configuration par défaut (écrasement de la configuration par défaut).



Attention, cette opération est irréversible. Seule la réinstallation du post-processeur permet de restaurer l'élément de configuration.

Utiliser les déclarations par défaut.

Cliquez avec le bouton droit de la souris sur l'élément « Configuration / Post-processeur / <nom de la cible> / ... » et choisissez « Utiliser la configuration par défaut ». L'élément de configuration du projet en cours est écrasé par la configuration par défaut.

Visualiser et modifier les éléments de configuration

Vous accédez à ces fichiers de configurations en double cliquant sur l'élément « Configuration / Post-processeurs / <nom de la cible> / ... ». S'ouvre alors une fenêtre permettant de visualiser et de modifier l'élément de configuration.

Systeme

Cet élément de configuration est très spécifique à chaque post-processeur.

Configuration matérielle

Cette zone (optionnelle) doit être modifiée pour déterminer la configuration matérielle d'une cible (type de CPU, cartes d'entrées / sorties par exemple).

Configuration logicielle

Cette zone (optionnelle) doit être modifiée pour déterminer des caractéristiques propres à la configuration de l'application (la valeur du chien de garde par exemple).

Options de génération de code

Cette zone contient des réglages concernant la méthode de traduction que doit utiliser le post-processeur (réservés aux spécialistes). Le nombre d'options peut

différer d'un post-processeur à l'autre. Ci dessous se trouve la liste des options communes à tous les post-processeurs :

« Optimiser le code généré »

Généralement réglé sur « Oui ». Le réglage sur « Non » peut permettre une analyse plus aisée du code généré.

« Ne pas générer le code d'évolution des étapes Grafcet »

Réglé par défaut sur « Non ». Si réglé sur « Oui », vous devez écrire dans l'élément « Code constructeur de fin » les instructions permettant la recopie des états immédiats de variables booléennes vers les états passés (voir le chapitre Gestion des variables booléennes d'AUTOMGEN)

« Ne pas générer le code d'évolution des bits utilisateurs »

Identique à l'option précédente mais appliquée au bits utilisateurs (variables « U ») d'AUTOMGEN.

Déclarations de variables

Ce sont des déclarations de variables utilisées en interne par le post-processeur. Seuls des spécialistes peuvent avoir à modifier cet élément.

Autres éléments

D'autres éléments spécifiques à chaque post-processeurs peuvent exister.

Voir l'élément « Système » sous forme de textes

En cliquant sur l'icône  dans la barre d'outils, vous basculez du mode « arborescence » au mode « texte » (format des anciennes versions d'AUTOMGEN). Dans le format « Texte » vous pouvez copier et coller des informations entre les fichiers de configuration.



Les modifications en mode « texte » doivent être réservées aux spécialistes, toute modification intempestive peut entraîner des erreurs de compilation difficiles à localiser pour un néophyte.

Afficher les éléments système

En double cliquant sur « Configuration / Post-processeurs / <nom du post-processeur> / Système » vous ouvrez une fenêtre qui à l'aspect suivant.

Eléments	Valeurs	Commentaires
<input type="checkbox"/> Configuration matérielle		
..... Type de l'automate	10	RFX 10
..... Module d'extension présent	NO	
..... Présence batterie		
<input type="checkbox"/> Configuration logicielle		
..... Code compatible RFX C3	NO	
..... Chien de garde en ms		
<input type="checkbox"/> Options de génération de code (attention, modifier avec précaution)		
..... Optimiser le code généré	Oui	
..... Ne pas générer le code d'évolution des étapes Grafcet	Non	
..... Ne pas générer le code d'évolution des bits utilisateur	Non	
..... Utiliser un seul bit automate pour chaque bit utilisateur AUTOMGEN	Non	
<input type="checkbox"/> Déclaration de variables		
<input checked="" type="checkbox"/> Affectation unitaire (une variable AUTOMGEN à une variable automate)		
..... Affectation linéaire (une table de variables AUTOMGEN à une table de variables automate)		
..... Affectation automatique (un ou plusieurs types de variables AUTOMGEN à une table de variables automate)		

Exemple de configuration système

Correspondances de variables

La maîtrise de la correspondance des variables est un des éléments essentiel à la maîtrise de l'utilisation des post-processeurs.

Lorsque le post-processeur traduit un fichier du langage pivot d'AUTOMGEN vers un langage cible spécifique, il doit attribuer les variables d'AUTOMGEN à des variables de la cible.

Par exemple, si vous avez utilisé le mot 200 d'AUTOMGEN dans votre application (il s'appelle M200 ou %MW200 dans AUTOMGEN) ce mot doit avoir une existence dans la mémoire de la cible et donc être repéré par un nom propre à cette cible.

AUTOMGEN propose trois types de déclaration de correspondance de variables :

- affectation unitaire,
- affectation linéaire
- affectation automatique.

La correspondance des variables pour un projet sera constituée de « n » affectations utilisant chacune un de ces trois types.

L'affectation unitaire

Elle permet d'associer une variable AUTOMGEN à une variable de la cible. Elle est la plus simple des déclarations.

Elle doit être utilisée uniquement si une seule déclaration est nécessaire.

Cette déclaration utilise deux informations : le nom de la variable AUTOMGEN et le nom de la variable de la cible.

« Associes cette variable d'AUTOMGEN à cette variable de la cible », ainsi peut être résumée l'affectation unitaire.

L'affectation linéaire

C'est une forme plus évoluée de l'affectation unitaire.

Elle permet d'associer une série de variables consécutives (plusieurs variables de même type dont les numéros se suivent) d'AUTOMGEN à une série de variables consécutives de la cible.

Cette affectation est typiquement utilisée pour :

- la déclaration des variables d'entrées / sorties,
- la déclaration de tables de bits ou de mots devant avoir une adresse fixe (pour un lien avec un pupitre opérateur par exemple).

Cette déclaration nécessite trois informations : le nom de la première variable AUTOMGEN, le nom de la première variable de la cible et la dimension de la table en nombre de variables.

« Associes dans l'ordre cette table de variables AUTOMGEN à cette table de variables de la cible », ainsi peut être résumé l'affectation linéaire.

L'affectation automatique

C'est le type de déclaration le plus complexe et le plus puissant. Il permet d'associer un ou plusieurs types de variables AUTOMGEN à une plage de variables de la cible.

Cette affectation laisse le soin au compilateur de trouver une affectation à chaque variable présente dans le code généré (sous réserve quelle corresponde à l'un des types) de la déclaration.

Ce type de déclaration est typiquement utilisé pour toutes les variables de l'application AUTOMGEN dont l'adresse de la variable associée dans la cible n'a pas besoin d'être précisément fixée.

Cette déclaration nécessite trois informations :

- les type de variables AUTOMGEN (voir le chapitre Types de variables supprimer AUTOMGEN),
- le nom de la première variable de la plage de la cible,
- le numéro de la dernière variable (incluse) de la plage de la cible.

L'affectation automatique n'est utilisée par un post-processeur que si aucune autre déclaration n'a été trouvée pour une variable. Si par exemple une directive d'affectation linéaire définit l'attribution pour les mots 200 à 210 d'AUTOMGEN, alors le post-processeur n'utilisera pas l'affectation automatique pour essayer d'allouer ces mots.

Si plusieurs affectations automatiques existent pour un même type de variable AUTOMGEN, alors le post-processeur utilise la première plage de variables de la cible jusqu'à saturation puis la seconde jusqu'à saturation, puis la troisième, etc...

Si au terme de l'utilisation de toutes les affectations automatiques une variable ne peut être allouée, alors un message d'erreur est généré par le compilateur indiquant que la variable n'est pas définie.

« Lorsque tu rencontres un de ces types de variables, utilises une variable de la cible de cette zone », ainsi peut être résumé l'affectation automatique.

Types de variables supprimer AUTOMGEN

Utilisés pour déclarer les correspondances de variables, ils sont un sur ensemble (car plus d'une variable de la cible peut être nécessaire pour loger une variable d'AUTOMGEN) des types de variables AUTOMGEN.

Gestion des variables booléennes d'AUTOMGEN

Un des principes de base de la traduction des langages booléens par le compilateur AUTOMGEN est de pouvoir accéder à deux états pour une même variable booléenne.

Ce concept fait référence à la notion de « cycle d'exécution » : entité représentant l'action réalisée par la cible consistant à lire les instructions de l'application de façon linéaire (du début jusqu'à la fin) et à accomplir les traitements qui leurs correspondent.

Ces deux états sont définis comme suit :

- 1- L'état immédiat de la variable : l'état écrit par la dernière instruction exécutée par la cible se reportant à cette variable, ou, à défaut celui qu'avait la variable à la fin du dernier cycle d'exécution, ou à défaut, si c'est le premier cycle d'exécution l'état d'initialisation de la variable.
- 2- L'état passé de la variable : l'état qu'avait la variable à la fin du dernier cycle d'exécution.

Remarques : ces deux états n'ont de validité que pour la tâche principale de l'application. Seul l'état immédiat a un sens pour les tâches asynchrones.

Le code généré par le compilateur AUTOMGEN assume ce qui suit :

- une affectation de variable booléenne se fait sur son état immédiat,
- un test de variable booléenne se fait sur son état passé.

Ces deux règles permettent de garantir une cohérence d'évolution des applications booléennes et notamment le respect des règles d'évolution des programmes générés par une description en langage Grafcet.

Le code généré par le post-processeur gère la recopie des états immédiats de variables vers les états passés en fin de cycle.

Lorsqu'une variable booléenne est utilisée dans AUTOMGEN deux variables booléennes sont utilisées sur la cible.

Trois exceptions existent :

- 1- pour une entrée tout ou rien, si aucun test de front n'est utilisé, seul l'état passé (« bi ») est utilisé (économie d'une variable booléenne),
- 2- pour une sortie tout ou rien, si aucun test de front n'est utilisé seul l'état immédiat (« o ») est utilisé.

(ceci explique pourquoi seules les variables « bi » et « o » se trouvent dans les directives d'attribution de variables).

- 3- pour le post-processeur ZELIO, compte tenu de la gestion temporelle des variables (quasi identique à celle d'AUTOMGEN) seuls les états immédiats sont utilisés dans le programme en langage ZELIO.

Syntaxe des éléments standards

« <nom de variable AUTOMGEN> » fait référence à l'état immédiat d'une variable booléenne ou à une variable numérique.

« b<nom de variable AUTOMGEN> » fait référence à l'état passé d'une variable booléenne.

Syntaxes spéciales pour les fronts

« u<nom de variable AUTOMGEN> » fait référence à l'état « front montant » d'une variable booléenne.

« d<nom de variable AUTOMGEN> » fait référence à l'état « front descendant » d'une variable booléenne.

Syntaxes spéciales pour les temporisations

« tempo <numéro> » fait référence au numéro d'une temporisation.

« tconsi<numéro> » fait référence à la consigne d'une temporisation.

« tcompt<numéro> » fait référence au compteur de temps d'une temporisation.

Autres syntaxes spéciales (réservés aux spécialistes)

« ac » fait référence à l'accumulateur 16 bits.

« al » fait référence à l'accumulateur 32 bits.

« af » fait référence à l'accumulateur flottant.

« cf » fait référence au drapeau de retenue.

« zf » fait référence au drapeau de résultat nul.

« sf » fait référence au drapeau de résultat négatif.

« of » fait référence au drapeau de débordement.

Afficher les éléments de correspondances de variables

En double cliquant sur « Configuration / Post-processeurs / <nom du post-processeur> / Correspondance des variables » vous ouvrez une fenêtre qui a l'aspect suivant.

Eléments	Valeurs	Commentaires
<input type="checkbox"/> Déclaration de variables		
Affectation unitaire (une variable AUTOMGEN à une variable automate)		
<input type="checkbox"/> Affectation linéaire (une table de variables AUTOMGEN à une table de variables automate)		
..... <-32-> bi0	i0,0	
..... <-16-> o0	o0,0	
..... <-32-> tempo	0	
..... <-15-> c0	c0	
<input type="checkbox"/> Affectation automatique (un ou plusieurs types de variables AUTOMGEN à une table de variables automate)		
..... i&bo&x&bx&bb&bu&b&u&t&bt&ux;	b2:255	
..... i&bo&x&bx&bb&bu&b&u&t&bt&ux;	x1:62	

Exemple de correspondances de variables

Remarque : dans le cas où un même post-processeur peut générer du code pour plusieurs types de cible (plusieurs types de CPUs d'automate par exemple) les

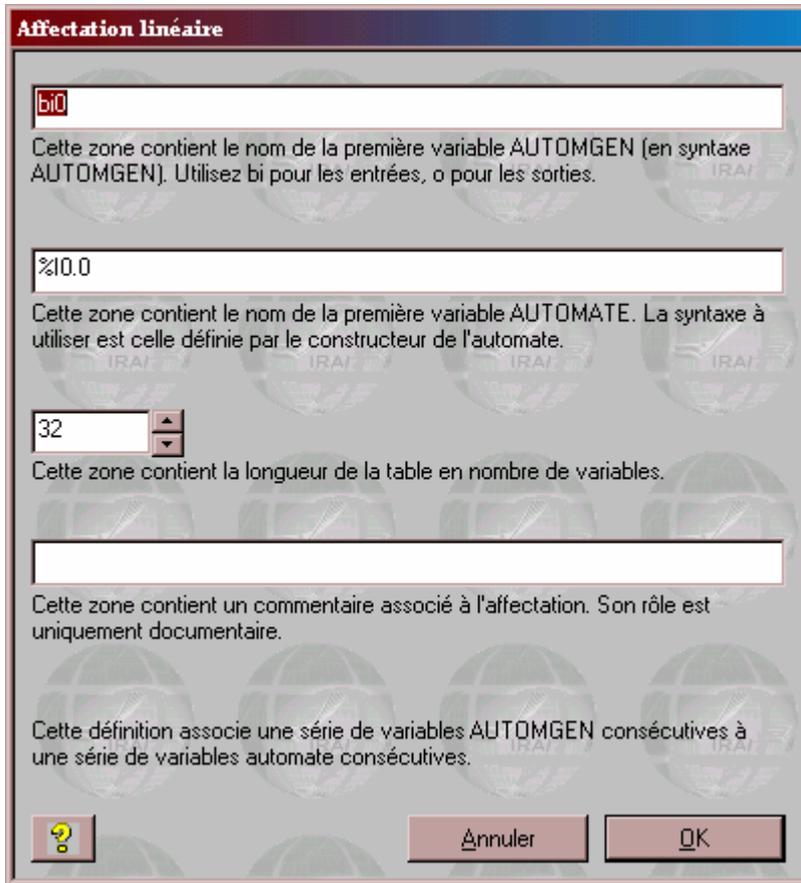
différents éléments peuvent être conditionnés pour l'ensemble des types de cibles ou pour un type de cible en particulier. Si les éléments sont conditionnés, ils sont rattachés à des lignes « Seulement pour xxx ». Voir exemple ci après.

Eléments	Valeurs	Commentaires
- Déclaration de variables		
- Affectation unitaire (une variable AUTOMGEN à une variable automate)		
- Affectation linéaire (une table de variables AUTOMGEN à une table de variables automate)		
- Seulement pour 1720		
- <-32-> bi0	i0,0	
- <-16-> o0	o0,0	
+ Seulement pour 47		
+ Seulement pour 4720		
+ Seulement pour 27		
- <-32-> tempo	0	
- Affectation automatique (un ou plusieurs types de variables AUTOMGEN à une table de variables automate)		
- i&bo&x&bx&bb&bu&b&u&t&bt&uxi	b0:255	
- i&bo&x&bx&bb&bu&b&u&t&bt&uxi	x0:95	
- m&c	w1:1023	

En cliquant sur les éléments « + » dans l'arborescence, vous développez les branches, « - » les referme.

Modifier un élément de correspondance de variables

En double cliquant sur chacun des éléments, vous pouvez les modifier.

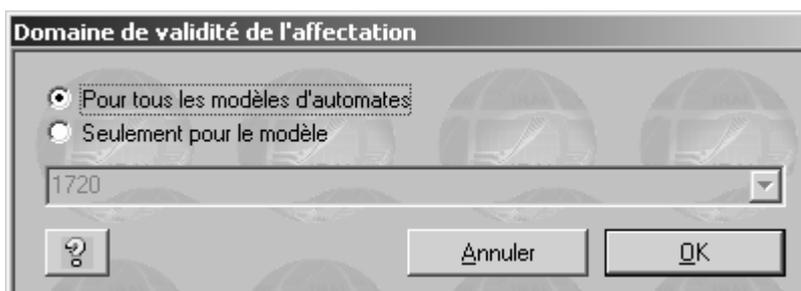


Exemple de boîte de dialogue de configuration d'une affectation linéaire.

Ajouter un élément de correspondance de variables

En cliquant avec le bouton droit de la souris sur les éléments « Affectation ... » de l'arborescence et en choisissant « Ajouter » dans le menu , vous pouvez ajouter une nouvelle affectation.

Si plusieurs types de cibles sont gérés par le post-processeur, la boîte de dialogue suivante permet de déterminer si la nouvelle affectation est seulement pour un type en particulier ou pour tous les types.



Affectation unitaire

Cette zone contient le nom de la variable AUTOMGEN (en syntaxe AUTOMGEN). Utilisez bi pour les entrées, o pour les sorties.

Cette zone contient le nom de la variable AUTOMATE. La syntaxe à utiliser est celle définie par le constructeur de l'automate.

Cette zone contient un commentaire associé à l'affectation. Son rôle est uniquement documentaire.

Cette définition associe une variable d'AUTOMGEN à une variable de l'automate.



Affectation unitaire

Affectation linéaire

Cette zone contient le nom de la première variable AUTOMGEN (en syntaxe AUTOMGEN). Utilisez bi pour les entrées, o pour les sorties.

Cette zone contient le nom de la première variable AUTOMATE. La syntaxe à utiliser est celle définie par le constructeur de l'automate.



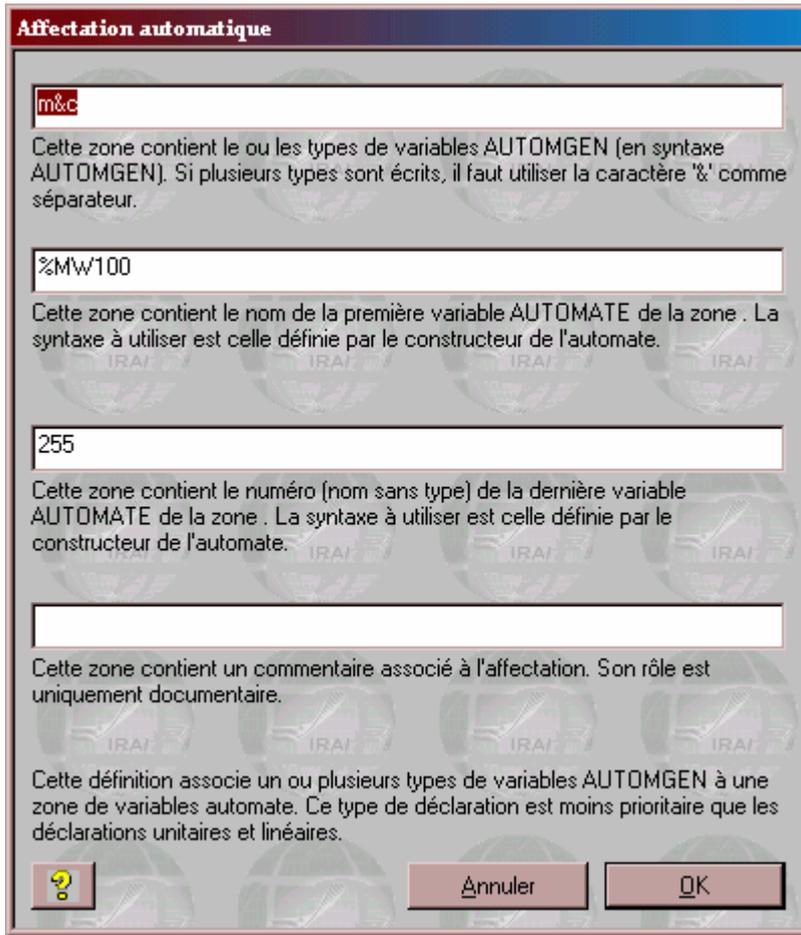
Cette zone contient la longueur de la table en nombre de variables.

Cette zone contient un commentaire associé à l'affectation. Son rôle est uniquement documentaire.

Cette définition associe une série de variables AUTOMGEN consécutives à une série de variables automate consécutives.



Affectation linéaire



Affectation automatique

Notez que les types de variables AUTOMGEN doivent être séparés en utilisant le caractère « & ».

Supprimer un élément de correspondances de variable

Cliquez avec le bouton droit de la souris sur l'élément de correspondance de variables et choisissez « Supprimer » dans le menu.

Associer un bit d'AUTOMGEN à un bit système d'une cible

Deux déclarations sont nécessaires, il faut associer les deux variables d'état d'un bit « U » (« u » et « bu ») au bit système de la cible. Vous devez créer deux affectations unitaires, par exemple :

Eléments	Valeurs
<input type="checkbox"/> Déclaration de variables	
<input type="checkbox"/> Affectation unitaire (une variable AUTOMGEN à une variable automate)	
<input type="checkbox"/> u100	sy5
<input type="checkbox"/> bu100	sy5
<input type="checkbox"/> Affectation linéaire (une table de	

Attention, en affectant la variable u et bu d'AUTOMGEN au même bit système de la cible, vous annulez la possibilité de réaliser un test de front montant ou descendant dans l'application. Vous pouvez contourner ce problème en utilisant la syntaxe

« $\uparrow(u<n>)$ » ou « $\downarrow(u<n>)$ » (avec « $<n>$ » représentant le numéro du bit) dans l'application (cette syntaxe génère un bit intermédiaire sur lequel le front sera évalué correctement).

Associer une table de mots d'AUTOMGEN à une table de mots fixes de la cible

Pour ceci, une seule déclaration linéaire suffit, par exemple :

Affectation linéaire (une table de variables AUTOMGEN à une table de variables automate)	
$\langle -64 \rangle$ bi0	e0.0
$\langle -64 \rangle$ o0	a0.0
+ Seulement pour 212	
+ Seulement pour 214	
+ Seulement pour 215	
+ Seulement pour 216	
+ Seulement pour 221	
+ Seulement pour 222	
+ Seulement pour 224	
$\langle -10 \rangle$ m200	Vw128



Les mots de la cible ainsi alloués doivent être enlevés d'éventuelles autres affectations sous peine d'affecter deux fois de mêmes variables de la cible à plusieurs variables AUTOMGEN différentes.

Associer des mots d'AUTOMGEN à des entrées ou des sorties analogiques d'une cible

Utilisez des déclarations linéaires, par exemple :

Affectation linéaire (une table de variables AUTOMGEN à une table de variables automate)		
+ Seulement pour 1720		
+ Seulement pour 47		
+ Seulement pour 4720		
+ Seulement pour 27		
$\langle -32 \rangle$ tempe	0	
$\langle -2 \rangle$ m200	Iw1.0	entrées analogiques
$\langle -2 \rangle$ m202	Ow2.0	sorties analogiques
Affectation automatique (un ou		

Associer une table de bits d'AUTOMGEN à une table de bits d'une cible

Deux affectations linéaires sont nécessaires. Par exemple :

Affectation linéaire (une table de variables AUTOMGEN à une table de variables automate)	
..... <-16-> tconsi0	kd0,1
..... <-16-> bi0	e62,0
..... <-16-> a0	a62,0
..... <-10-> u100	m2,0
..... <-10-> bu100	m2,0
Affectation automatique fun ou	

Cet exemple (les états immédiats et les états passés sont associés aux mêmes variables de la cible) interdit l'utilisation de tests de front sur les bits AUTOMGEN u100 à u109.

Pour contourner le problème deux solutions sont possibles :

- utiliser la syntaxe « $\uparrow(u<n>)$ » ou « $\downarrow(u<n>)$ » (avec « <n> » représentant le numéro du bit) dans l'application,
- associer les bits états immédiats et passés à deux tables de bits différents dans la cible. Dans ce cas, les accès extérieurs à l'application qui pourraient être réalisés sur ces bits (par un terminal de dialogue ou un logiciel de supervision par exemple) doivent respecter la philosophie d'AUTOMGEN : accès en lecture sur les états passés, accès en écriture sur les états immédiats (un accès en lecture sur les états immédiats est dans la pratique possible).



Les bits de la cible ainsi alloués doivent être enlevés d'éventuelles autres affectations sous peine d'affecter deux fois de mêmes variables de la cible à plusieurs variables AUTOMGEN différentes.

Voir les correspondances de variables sous forme de textes

En cliquant sur l'icône  dans la barre d'outils, vous basculez du mode « arborescence » au mode « texte » (format des anciennes versions d'AUTOMGEN). Dans le format « Texte » vous pouvez copier et coller des informations entre les fichiers de configuration.



Les modifications en mode « texte » doivent être réservées aux spécialistes, toute modification intempestive peut entraîner des erreurs de compilation difficiles à localiser pour un néophyte.

Code constructeur démarrage, code constructeur de fin

Ces éléments de configuration contiennent du code machine propre à chaque cibles sous forme textuel.

La syntaxe à utiliser dans ces sections est proche des langages de bas niveau utilisables sur chaque cibles. L'observation du code généré en passe 1 par chaque post-processeur vous permet de visualiser la syntaxe à utiliser.

Référence à une variable AUTOMGEN

Utilisez la syntaxe « `_<nom de variable AUTOMGEN>_` » pour faire référence à une variable AUTOMGEN (pensez à ajouter le caractère « `b` » en tête de variable pour accéder à l'état passé d'une variable booléenne. Par exemple « `_bu100_` »).

Référence à un symbole de l'application AUTOMGEN

Syntaxe :

`_|nom du symbole|_`

Le caractère « `|` » est généralement associé à la touche 6 du clavier.

Définition et référence à un label

« `@<nom de label>` » marque une destination de saut,

« `_<nom de label>_` » fait référence au label.

Insérer du code constructeur dans une application

Les mots clés « `#BEGIN_MACHINE_CODE` » et « `#END_MACHINE_CODE` » permettent d'insérer du code constructeur dans une boîte de code AUTOMGEN.

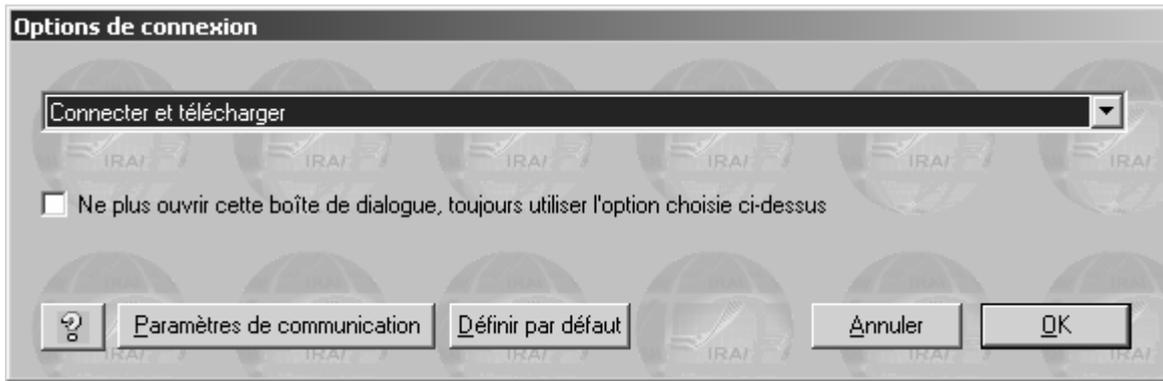
Ces deux directives doivent se trouver en début d'une ligne, aucun autre caractère ne doit se trouver sur la même ligne.

Les lignes se trouvant entre ces deux directives définissent une zone nommée « Section de langage constructeur ».

La syntaxe à utiliser dans une section de langage constructeur est la même que celle utilisée dans les éléments « Code de démarrage » et « Code de fin ».

Choix des options de connexion

Double cliquez sur l'élément « Configuration / Post-processeur / <nom du post-processeur> / Options de connexion ».



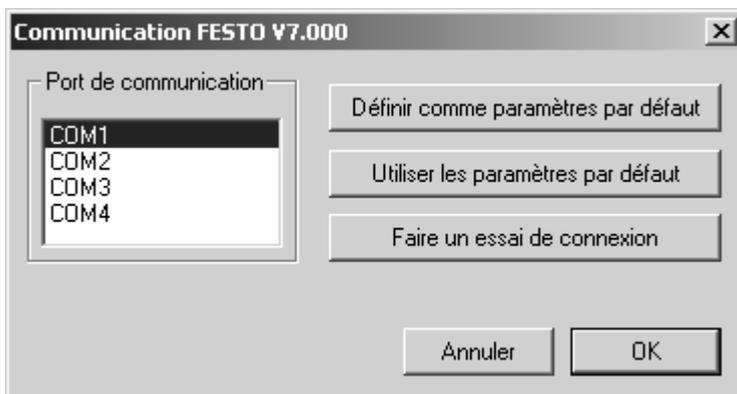
Choix d'un mode de connexion

Les modes de connexions possibles sont plus ou moins nombreux en fonction du post-processeur. Le mode « Seulement connecté » est typiquement utilisé pour créer une application de supervision.

Cette boîte de dialogue s'ouvre automatiquement lorsqu'une connexion à une cible est demandée. En cochant la case « Ne plus ouvrir ... », cette ouverture n'est plus automatique. Pour l'ouvrir de nouveau, laissez enfoncée la touche [Shift] du clavier en lançant la commande de connexion ou la commande « Go ».

Paramétrage du module de communication

Double cliquez sur l'élément « Configuration / Post-processeur / <nom du post-processeur> / Module de communication ».



Exemple de paramétrage d'un module de communication.

La configuration en cours peut être définie comme configuration par défaut (pour les nouveaux projets) ou rétablie par défaut.

Un essai de connexion peut être réalisé.

Post-processeur PL7

Ce post-processeur permet de programmer les automates MODICON TELEMECANIQUE SCHNEIDER TSX 37, (TSX MICRO) et TSX 57 (TSX PREMIUM).

Module de communication

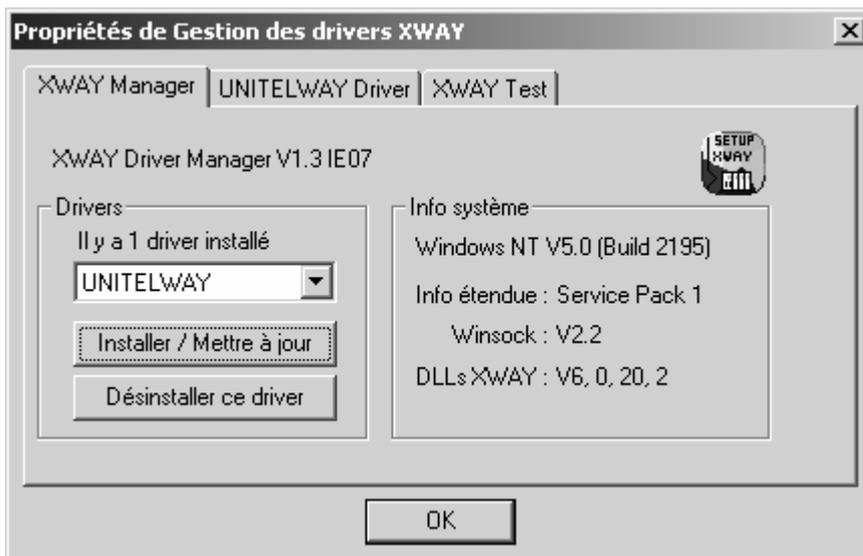


Le driver UNITELWAY SCHNEIDER doit impérativement être installé sur l'ordinateur (en local) pour pouvoir communiquer avec les automates SCHNEIDER TSX 37 et TSX 57. Des drivers adaptés à une ou plusieurs versions de WINDOWS se trouvent sur le CD-ROM et peuvent être téléchargés sur le site d'IRAI : www.iraicom.com.

Le module de communication utilise le driver de communication conçu par SCHNEIDER AUTOMATION. Cliquer sur « Paramétrer et tester ... » vous permet d'accéder directement au menus du driver de communication SCHNEIDER.



Paramétrage du module de communication



Propriétés du module de communication UNITELWAY

Mode de génération d'un fichier exécutable

Le post-processeur peut générer soit un fichier binaire directement téléchargeable dans l'automate (disponible uniquement sur TSX 37, non disponible sur TSX 57), soit un fichier importable dans les outils SCHNEIDER (disponible pour TSX 37 et TSX 57). La première solution est nettement préférable (gain de temps, simplicité d'utilisation).

Le choix du mode s'effectue dans la configuration logicielle du post-processeur.

Mode de génération directe du fichier binaire

Ce mode est fortement recommandé pour TSX 37.

Il engendre les restrictions suivantes :

- pas d'utilisation d'instruction de sous-programmes,
- pas de support d'instructions spécifiques (communication ou PID par exemple).

Si votre application doit utiliser des éléments très spécifiques, utilisez une des méthodes d'importations décrite ci-après.

Eléments	Valeurs
+ Configuration matérielle	
- Configuration logicielle	
Générer directement le fichier binaire sans passer par le logiciel SCHNEIDER	YES

Sélection de la génération automatique d'un fichier binaire.

Fichier de configuration de l'automate

Pour les automates TSX 37-05 et TSX 37-08 en version de base (avec une seule carte d'entrées / sorties), des fichiers 3705.stx et 3708.stx sont fournis en standard dans le répertoire d'installation d'AUTOMGEN.



Si le fichier de configuration n'est pas créé, alors les sorties TOR de l'automate ne seront pas activées.

Une fois le fichier créé ou téléchargé (voir ci-après), donnez le chemin d'accès au fichier dans l'élément de configuration suivant :

- Configuration matérielle	
Fichier contenant la configuration matérielle de l'automate	<AUTOM7DIR>\3708.stx

Le nom du fichier contenant la configuration.

Quatre méthodes sont possibles pour obtenir un fichier de configuration :

Utiliser le configurateur fourni par IRAI

Ce configurateur est au téléchargement sur le site d'IRAI (www.irai.com rubrique « Téléchargement / AUTOMGEN7 / utilitaires ») et permet de créer directement les fichiers .STX correspondant à la configuration de votre API. Ce configurateur comporte certaines restrictions : pas de support des cartes de communications, pas de relecture de fichier .STX existants.

Téléchargement du fichier de configuration sur le site IRAI

- 1- téléchargez un fichier correspondant à la configuration de votre automate sur le site d'IRAI : www.irai.com, rubrique « Téléchargement / AUTOMGEN7 / fichiers de configuration pour automate TSX 37 » (recommandé si le fichier de configuration est présent sur le site),
- 2- recopiez le fichier ainsi téléchargé sans le décompresser (les fichiers « .STX » sont des fichiers compressés) dans le répertoire d'installation d'AUTOMGEN ou insérez le dans les ressources du projet AUTOMGEN.



Décompresser le fichier « .STX » empêchera son utilisation par le post-processeur.

Création d'un fichier avec un des outils de programmation SCHNEIDER

Les outils logiciel SCHNEIDER (PL7MICRO V3.1, PL7JUNIOR V3.1 ou PL7PRO V3.4) sont utilisables. Les fichiers créés avec d'autres versions peuvent ne pas fonctionner, dans ce cas, l'automate passe en mode erreur à l'issue du téléchargement de l'application (Voyant « ERR » allumé sur l'automate).

Pour créer le fichier « .STX » :

- 1- lancer un des outils SCHNEIDER, créer une application en suivant les règles suivantes :
 - o choisissez le type de la CPU de votre automate en sélectionnant toujours la version 1.0 de la CPU,
 - o sélectionnez la ou les cartes d'entrées / sorties installées sur votre automate et au besoin paramétrez les,
- 2- sauvegarder le fichier ainsi créé dans le répertoire d'installation d'AUTOMGEN ou intégrez le dans les ressources du projet AUTOMGEN.

Communiquez la communication par email à IRAI pour obtenir un fichier de configuration

- 1- envoyez un email à IRAI en demandant un fichier de configuration en précisant :
 - o le type de CPU TSX 37-05, 37-08, 37-10, 37-21 ou 3722,
 - o la position et le type précis des cartes d'entrées / sorties (DMZ ...).
- 2- à la réception du fichier par mail, recopiez le dans le répertoire d'installation d'AUTOMGEN (sans le décompresser) ou intégrez le dans les ressources du projet AUTOMGEN.

Mode de génération d'un fichier « .FEF »

Dans ce mode, l'importation dans les outils de programmation SCHNEIDER (PL7 MICRO (TSX 37), PL7 JUNIOR (TSX 37 ou TSX 57) ou PL7 PRO (TSX 37 ou TSX 57) peut être automatique ou manuelle.

Importation manuelle

Configuration logicielle	
Générer directement le fichier binaire sans passer par le logiciel SCHNEIDER	NO
Lancer automatiquement le logiciel SCHNEIDER (Seulement si BUILD BIN=NO)	NO

Sélection du mode d'import manuel

Vous devez choisir un nom de fichier qui sera exporté depuis AUTOMGEN :

Configuration logicielle	
Générer directement le fichier binaire sans passer par le logiciel SCHNEIDER	NO
Lancer automatiquement le logiciel SCHNEIDER (Seulement si BUILD BIN=NO)	YES
Version du logiciel SCHNEIDER (seulement si BUILD BIN=NO et RUNPL7SOFT=YES)	TOPL7PRO.EXE
Fichier à importer dans PL7Micro ou PL7Junior ou PL7PRO après compilation (BUILD BIN=NO)	c:\export.fef

Choix d'un fichier pour l'export vers l'atelier logiciel SCHNEIDER

Procédure :

- 1- Compilez l'application dans AUTOMGEN en utilisant la commande « Compile » du menu « Programme » ou en cliquant sur le bouton  de la barre d'outils,

- 2- Lancez un atelier logiciel SCHNEIDER, créez un nouveau projet et utilisez la commande « Importer une application » du menu « Fichier »,
- 3- A la fin de l'importation, transférez l'application dans l'automate,
- 4- Pour obtenir la visualisation dynamique dans AUTOMGEN, cliquez sur le bouton « Go » de la barre d'outils et choisissez comme mode de connexion « Seulement connecter ».

Importation automatique

L'outil logiciel SCHNEIDER sera lancé automatiquement. Seul un nombre restreint de versions des logiciels SCHNEIDER sont utilisables. Le type et la version du logiciel SCHNEIDER doit être réglé dans la configuration logicielle.

Configuration logicielle		
Générer directement le fichier binaire sans passer par le logiciel SCHNEIDER	NO	
Lancer automatiquement le logiciel SCHNEIDER (Seulement si BUILD BIN=NO)	YES	
Version du logiciel SCHNEIDER (seulement si BUILD BIN=NO et RUNPL7SOFT=YES)	TOPL7JU2.EXE	PL7 Junior version 3.1

Sélection du type et de la version du logiciel SCHNEIDER

Le fonctionnement de la procédure d'import automatique avec d'autres versions des logiciels SCHNEIDER n'est pas garanti.

Procédure à réaliser une seule fois :

- 1- Lancer un outil de programmation SCHNEIDER et créez une nouvelle application,
- 2- Configurez l'application : type d'automate, cartes d'entrées / sorties, etc...
- 3- Sauvegardez le fichier ainsi créé,
- 4- Donnez le chemin d'accès complet à ce fichier sous la rubrique « Configuration matérielle » de l'élément « Système », par exemple :

Configuration matérielle	
Fichier contenant la configuration matérielle de l'automate	c:\pl7user\automgen.stx

Procédure à réaliser à chaque exécution une application :

- 1- Lancez l'outil logiciel SCHNEIDER (si il ne l'est pas déjà),
- 2- Cliquez sur le bouton « GO » dans la barre d'outils d'AUTOMGEN.

Utilisation de tâches d'interruptions

En définissant un folio de type tâche, vous pouvez insérer du langage bas niveau d'AUTOMGEN ou du langage constructeur à une tâche de l'automate. Le tableau ci-dessous donne la correspondance entre le numéro de tâche et le type de tâche d'interruption de l'automate.

Numéro de tâche (folio AUTOMGEN)	Type de tâche automate TSX 37	Type de tâche automate TSX 57
0	Tâche rapide	Tâche rapide
1	EVT1	EVT0
2	EVT2	EVT1
3	EVT3	EVT2
etc...		

Exemples spécifiques

Ces exemples se trouvent dans le répertoire « <répertoire d'installation d'AUTOMGEN> /Exemples/Post-processeurs/PL7 ». Les fichiers portent le même nom que le titre des chapitres qui suivent.

Entrées / sorties analogiques

Cet exemple illustre l'utilisation des entrées / sorties analogiques.

Déclaration de variables	
Affectation unitaire (une variable AUTOMGEN à une variable automate)	
..... m62	%SW30
..... b8	%S11
..... m200	%qw0.10
Affectation linéaire (une table de variables AUTOMGEN à une table de variables automate)	
..... <-32-> bi0	%I1.0
..... <-32-> o0	%Q2.0
..... <-64-> tempe	0
..... <-8-> m201	%iw0.2

Déclaration des entrées / sorties analogiques sur un automate TSX 37-22.

Compteur rapide TSX 37-10

Cet exemple illustre l'utilisation du compteur rapide sur un automate TSX 37-10.

Compteur rapide TSX 37-10 utilisé en décomptage

Cet exemple illustre l'utilisation du compteur rapide sur un automate TSX 37-10 en mode décomptage.

Compteur rapide TSX 37-22

Cet exemple illustre l'utilisation du compteur rapide sur un automate TSX 37-22.

ASI

Exemple d'utilisation d'entrées / sorties ASI

MAGELIS

Exemple d'utilisation d'un terminal MAGELIS

Post-processeur PL72

Ce post-processeur permet la programmation des automates TELEMECANIQUE SCHNEIDER TSX 17-20 (avec cartouche PL72), TSX 27, TSX 47 et TSX 47-20.

Choix du type de l'automate

L'élément « Configuration / Post-processeur / PL72 / Système / Configuration matérielle » du navigateur permet de choisir le type de l'automate.

Configuration matérielle	
Type de l'automate	1720

Éléments syntaxiques spécifiques

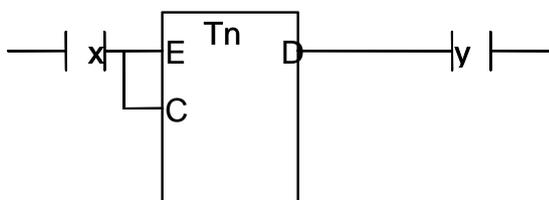
Appel des blocs fonction PL72

Les syntaxes suivantes permettent d'appeler les blocs temporisation, texte et compteur rapide (TSX 17-20) sous la forme textuel utilisé dans les éléments « Code constructeur de démarrage », « Code constructeur de fin » et les sections en langage constructeur.

Bloc temporisation

$x.Tn=y$

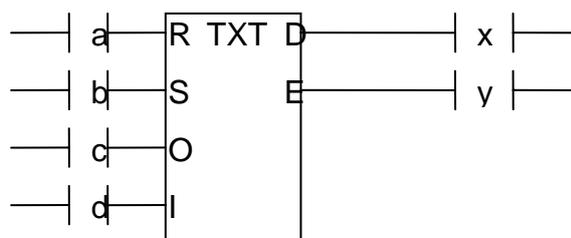
Equivalent PL72 :



Bloc texte

$a+b+c+d.TXTn=x:y$

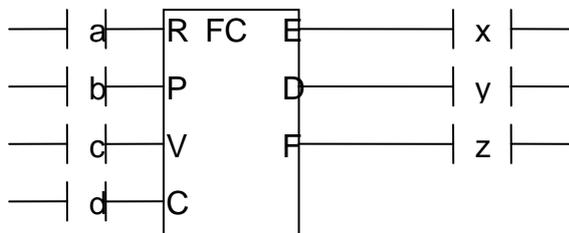
Equivalent PL72 :



Bloc compteur rapide

$a+b+c+d.FC=x:y:z$

Equivalent PL72 :



Bloc horodateur type WEEK (sur TSX 17-20 uniquement)

$a.H,W,(jours),(heure\ de\ début),(heure\ de\ fin)=x :y :z$

« jours » représente les jours de la semaine, c'est une valeur décimale codée sur 7 bits, chaque bit représente un jour de la semaine. Le jour est actif si le bit est à 1. b0 correspond à Dimanche et b6 à Samedi. Par exemple pour valider le lundi et le mercredi il faut écrire la valeur $2^1 + 2^3 : 2 + 8 = 10$. Pour valider les 7 jours de la semaine : la valeur est 127.

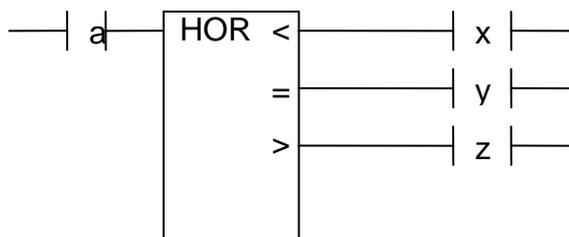
« heure de début » et « heure de fin » sont exprimées sous la forme HH:MM : heures et minutes.

Bloc horodateur type YEAR (sur TSX 17-20 uniquement)

$a.H,Y,(date\ de\ début),(date\ de\ fin)=x :y :z$

« date de début » et « date de fin » sont exprimées sous la forme JJ/MM : jours et mois.

Equivalent PL72 :



Utilisation de la tâche rapide

Un folio de type « Tâche » portant le numéro « 1 » permet d'associer le code littéral ou le code PL72 écrit sur le folio à la tâche rapide. Le folio ne peut contenir que du code littéral bas niveau ou du code PL72 écrit dans un rectangle d'organigramme.

Module de communication

Port de communication	
Port	Vitesse
COM1	1200
COM2	2400
COM3	4800
COM4	9600

Mode

Prise console Esclave

Unitelway 1

Toujours essayer une connexion à 19200 bauds

Définir comme paramètres par défaut Utiliser les paramètres par défaut Annuler

Faire un essai de connexion OK

Paramétrage du module de communication

Si vous connectez le PC sur la prise console de l'automate sélectionnez impérativement « Prise console ».

Ne cochez « Toujours essayer une connexion à 19200 bauds » que si votre automate est un TSX 17-20 récent (cette option permet un dialogue plus rapide entre le PC et l'automate).

Le mode « UNITELWAY » permet de connecter le PC à un coupleur UNITELWAY. Dans ce cas, la vitesse doit être en accord avec la configuration du coupleur.



Si vous cochez « Toujours essayer une connexion à 19200 bauds » et que votre TSX 17-20 ne supporte pas la communication à 19200 bauds alors la connexion échouera.

Si le mode n'est pas en accord avec la connexion (mode UNITELWAY sélectionné et connexion sur ma prise console par exemple) alors la connexion échouera.

Exemples spécifiques

Ces exemples se trouvent dans le répertoire « <répertoire d'installation d'AUTOMGEN> / Exemples / Post-processeurs / PL72 ». Les fichiers portent le même nom que le titre des chapitres qui suivent.

Entrées / sorties analogiques

Pour pouvoir utiliser les sorties analogiques sur un automate TSX 17-20 il faut :

- déclarer le ou les blocs d'entrées / sorties analogiques dans l'élément « Système » de la configuration,
- associer une ou plusieurs variables AUTOMGEN aux mots d'entrées / sorties TELEMECANIQUE (IW et OW).

Exemple :

- automate TSX 17-20 utilisant un bloc de 4 entrées analogiques (code 27) en position 1 et un bloc de 2 sorties analogiques (code 21) en position 2.
- le programme recopiera simplement l'état de la première entrée analogique sur la première sortie analogique. Il comparera également la deuxième entrée analogique avec la valeur 500 (valeur arbitraire) et positionnera deux sorties booléennes : O0 si l'entrée est inférieur à 500, O1 si l'entrée est supérieur ou égale à 500.

Configuration matérielle	
Type de l'automate	1720
Code de l'extension numéro 1 (0 si pas d'extension)	27 : TSX AEG 4111
Code de l'extension numéro 2 (0 si pas d'extension)	21 : TSX ASG 2001, TSX ASG 2000
Code de l'extension numéro 3 (0 si pas d'extension)	0 : aucune

La déclaration des deux modules d'extension

Affectation linéaire (une table de variables AUTOMGEN à une table de variables automate)	
Seulement pour 1720	
<-32-> bi0	i0,0
<-16-> o0	o0,0
<-4-> M200	Iw1,0
<-2-> M204	OW2,0

L'affectation des variables

Ces deux déclarations associent les mots M200 à M203 d'AUTOMGEN aux variables IW1,0 à IW1,3 de l'automate ainsi que les variables M204 et M205 d'AUTOMGEN aux variables OW2,0 et OW2,1 de l'automate.

Compteur rapide

Le but est de compter 200 impulsions sur le compteur rapide. La valeur courante du compteur rapide sera recopiée dans le mot M200 d'AUTOMGEN. La sortie O5 sera activée par la tâche rapide en fin de comptage.

Configuration matérielle	
Configuration logicielle	
Capacité mémoire en Ko	8
Compteur rapide	C
Prédisposition compteur rapide	200
Modification compteur rapide	YES
Entrées lues dans la tâche rapide	NO

Le paramétrage du compteur rapide

Blocs texte et xbt

Le but est de dialoguer avec un XBT connecté sur le port console d'un automate TSX 17-20.

Les entrées I0 à I3 déclencheront l'affichage des messages numéro 0 à numéro 3 enregistrés dans l'XBT.

Le bloc texte TXT1 sera utilisé pour dialoguer sur le port console.

Le format du message à envoyer à l'XBT pour afficher un message est le suivant :
ESC V xxx LF CR

xxx représente le numéro de message codé en décimal sur trois caractères.

Eléments	Valeurs
Déclaration de variables	
Affectation unitaire (une variable AUTOMGEN à une variable automate)	
Affectation linéaire (une table de variables AUTOMGEN à une table de variables automate)	
+ Seulement pour 1720	
+ Seulement pour 47	
+ Seulement pour 4720	
+ Seulement pour 27	
<-32-> tempo	0
<-4-> m200	w1
Affectation automatique (un ou plusieurs types de variables AUTOMGEN à une table de variables automate)	
i&bo&x&bx&bb&bu&b&u&t&bt&ux&dx&ui&di&uo&do&du&u	b0:255
i&bo&x&bx&bb&bu&b&u&t&bt&ux&dx&ui&di&uo&do&du&u	w0:95
m&c	w5:1023

Allocation d'une table de mots pour les échanges

Blocs textes	
Type de bloc texte 0	
Type de bloc texte 1	TER
Type de bloc texte 2	
Type de bloc texte 3	
Type de bloc texte 4	
Type de bloc texte 5	
Type de bloc texte 6	
Type de bloc texte 7	
Adresse du buffer pour le bloc texte 0 (CW0 à CW127 ou W0 à W1023)	
Adresse du buffer pour le bloc texte 1 (CW0 à CW127 ou W0 à W1023)	W1[0]
Adresse du buffer pour le bloc texte 2 (CW0 à CW127 ou W0 à W1023)	
Adresse du buffer pour le bloc texte 3 (CW0 à CW127 ou W0 à W1023)	
Adresse du buffer pour le bloc texte 4 (CW0 à CW127 ou W0 à W1023)	
Adresse du buffer pour le bloc texte 5 (CW0 à CW127 ou W0 à W1023)	
Adresse du buffer pour le bloc texte 6 (CW0 à CW127 ou W0 à W1023)	
Adresse du buffer pour le bloc texte 7 (CW0 à CW127 ou W0 à W1023)	
TXT0,L	
TXT1,L	7
TXT2,L	
TXT3,L	
TXT4,L	
TXT5,L	
TXT6,L	
TXT7,L	
TXT0,C	
TXT1,C	\$FD
TXT2,C	
TXT3,C	
TXT4,C	
TXT5,C	
TXT6,C	
TXT7,C	
TXT0,M	
TXT1,M	\$00
TXT2,M	
TXT3,M	
TXT4,M	
TXT5,M	
TXT6,M	
TXT7,M	
TXT0,A	
TXT1,A	\$FE
TXT2,A	
TXT3,A	
TXT4,A	
TXT5,A	
TXT6,A	
TXT7,A	
TXT0,T	
TXT1,T	0

Paramétrage du bloc texte

Blocs texte et UNITELWAY

Le but est d'utiliser un coupleur UNITELWAY pour effectuer l'acquisition d'une table de 3 mots sur un l'automate cible. Le coupleur UNITELWAY est installé comme première extension, il sera configuré en maître pour utiliser deux esclaves. L'automate lu sera l'esclave numéro 1.

Eléments	Valeurs
<input type="checkbox"/> Configuration matérielle	
..... Type de l'automate	1720
..... Code de l'extension numéro 1 (0 si pas d'extension)	63
..... Code de l'extension numéro 2 (0 si pas d'extension)	0
..... Code de l'extension numéro 3 (0 si pas d'extension)	0

Configuration du coupleur UNITELWAY

Blocs textes	
Type de bloc texte 0	
Type de bloc texte 1	CPL
Type de bloc texte 2	CPL
Type de bloc texte 3	
Type de bloc texte 4	
Type de bloc texte 5	
Type de bloc texte 6	
Type de bloc texte 7	
Adresse du buffer pour le bloc texte 0 (CW0 à CW127 ou W0 à W1023)	
Adresse du buffer pour le bloc texte 1 (CW0 à CW127 ou W0 à W1023)	CW0
Adresse du buffer pour le bloc texte 2 (CW0 à CW127 ou W0 à W1023)	W1[10]
Adresse du buffer pour le bloc texte 3 (CW0 à CW127 ou W0 à W1023)	
Adresse du buffer pour le bloc texte 4 (CW0 à CW127 ou W0 à W1023)	
Adresse du buffer pour le bloc texte 5 (CW0 à CW127 ou W0 à W1023)	
Adresse du buffer pour le bloc texte 6 (CW0 à CW127 ou W0 à W1023)	
Adresse du buffer pour le bloc texte 7 (CW0 à CW127 ou W0 à W1023)	
TXT0.L	
TXT1.L	10
TXT2.L	6
TXT3.L	
TXT4.L	
TXT5.L	
TXT6.L	
TXT7.L	
TXT8.C	
TXT1.C	\$40
TXT2.C	\$0736
TXT3.C	
TXT4.C	
TXT5.C	
TXT6.C	
TXT7.C	
TXT0.M	
TXT1.M	\$100
TXT2.M	\$165
TXT3.M	
TXT4.M	
TXT5.M	
TXT6.M	
TXT7.M	
TXT0.A	
TXT1.A	\$FE
TXT2.A	\$FE
TXT3.A	

Paramétrage des deux blocs texte

Affectation linéaire (une table de variables AUTOMGEN à une table de variables automate)	
+ Seulemment pour 1720	
+ Seulemment pour 47	
+ Seulemment pour 4720	
+ Seulemment pour 27	
<-32-> temps	0
<-10-> m200	w1
Affectation automatique (un ou plusieurs types de variables AUTOMGEN à une table de variables automate)	
i&bo&x&bx&bb&bu&bt&ux&dx&ui&di&uo&do&du&u	b0:255
i&bo&x&bx&bb&bu&bt&ux&dx&ui&di&uo&do&du&u	x0:95
m&c	w11:1023

Attribution d'une table de mots pour les échanges

Module d'extension TOR

Cet exemple illustre la configuration d'un module d'extension TOR. Nous prenons comme hypothèses un module de base équipé de 16 entrées et de 12 sorties et un module d'extension équipé de 10 entrées et de 8 sorties.

Eléments	Valeurs	Commentaires
Configuration matérielle		
Type de l'automate	1720	automate TSX17-20 avec cartouche PL72
Code de l'extension numéro 1 (0 si pas d'extension)	15	TSX DMF 401, TSX DMF 400, TSX DMF 342A, TSX DMF 344A
Code de l'extension numéro 2 (0 si pas d'extension)	0	aucune
Code de l'extension numéro 3 (0 si pas d'extension)	0	aucune
Entrée I/O 0	NORMAL	entrée normale

La définition du module d'extension

Affectation linéaire (une table de variables AUTOMGEN à une table de variables automate)	
Seulemment pour 1720	
<-16-> bi0	i0,0
<-12-> o0	o0,0
<-10-> bi16	i1,0
<-8-> o12	o1,0
Seulemment pour 47	

L'affectation des variables

Conversion

Montre comment appeler les fonctions de conversion du langage PL72.

Horodateur

Exemple d'utilisation du bloc fonctionnel horodateur.

Post-processeur S7200

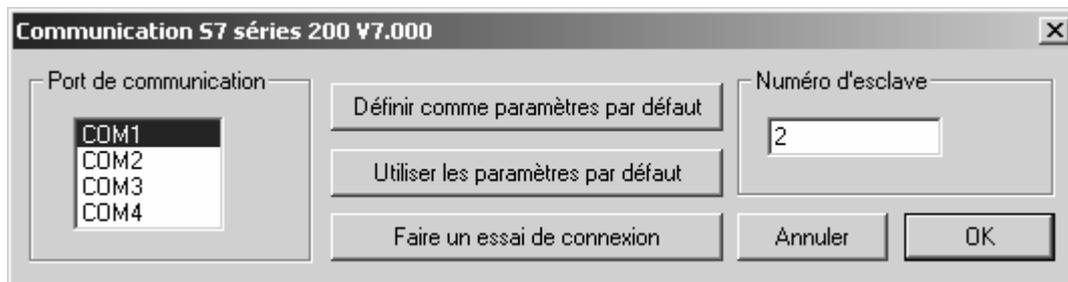
Ce post-processeur permet de programmer les automates SIEMENS S7200 (toutes les CPU 2xx).

Choix du type de CPU

L'élément « Configuration / Post-processeur / STEP7 (S7200) / Système / Configuration matérielle » du navigateur permet de choisir le type de la CPU.

Configuration matérielle	
Type de l'automate	224

Module de communication



Paramétrage du module de communication

Veillez à régler le numéro d'esclave en accord avec celui configuré sur l'automate.

Exemple spécifique

Cet exemple se trouve dans le répertoire « <répertoire d'installation d'AUTOMGEN> / Exemples / Post-processeurs / S7200 ». Le fichier porte le même nom que le titre du chapitre qui suit.

Tâche d'interruption

Exemple d'appel d'une tâche d'interruption

Post-processeur ABB

Ce post-processeur permet de programmer les automates ABB CS31 et AC31.

Choix du type d'automate

L'élément « Configuration / Post-processeur / ABB / Système / Configuration matérielle » du navigateur permet de choisir le type de l'automate.

Automate AC31

Configuration matérielle	
Type de l'automate	AC
Code compatible AC31	Oui

Automate CS31

Configuration matérielle	
Type de l'automate	CS
Code compatible AC31	Non

Module de communication

Paramétrage du module de communication

Utilitaire

L'élément « Configuration / Post-processeur / ABB / Terminal emulator » du navigateur permet d'accéder à un émulateur de terminal utilisable pour dialoguer avec l'automate.

Exemples spécifiques

Ces exemples se trouvent dans le répertoire « <répertoire d'installation d'AUTOMGEN> / Exemples / Post-processeurs / ABB ». Les fichiers portent le même nom que les titres des chapitres qui suivent.

Entrées / sorties analogiques

Exemple illustrant l'utilisation d'un module d'extension analogique sur un automate AC31.

Interruptions

Exemple illustrant l'utilisation des tâches d'interruptions sur un automate AC31.

Post-processeur GE-FANUC / ALSPA

Ce post-processeur permet de programmer les automates GE-FANUC 90 MICRO et 9030 ou ALSPA 8005 et 8035.

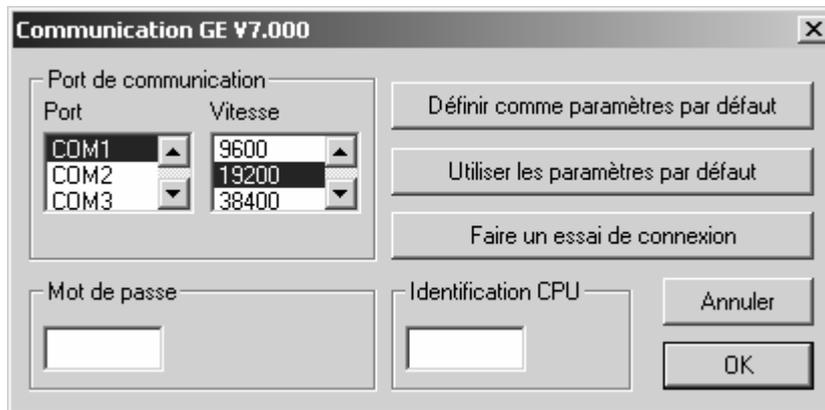
Choix du type d'automate

L'élément « Configuration / Post-processeur / GE-FANUC / Système / Configuration matérielle » du navigateur permet de choisir le type de l'automate.



Choisir standard pour les CPUs autres que 350, 351 ou VERSAMAX.

Module de communication



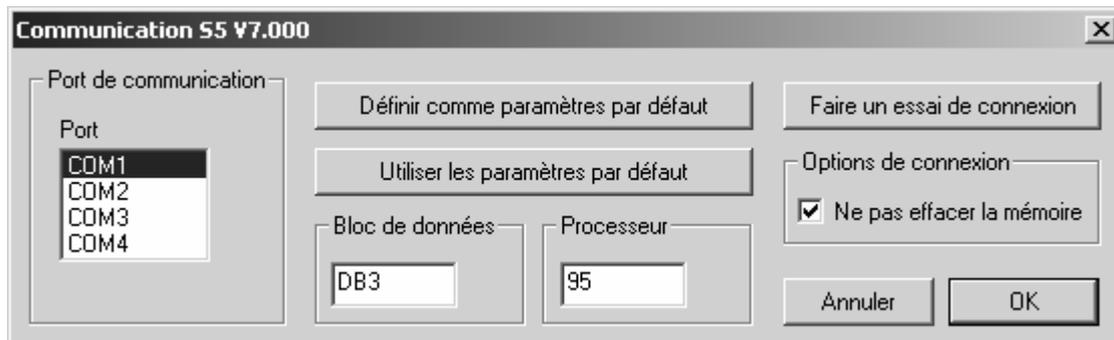
Paramétrage du module de communication

Utilitaire

L'élément « Configuration / Post-processeur / GE-FANUC / Hardware configuration & diagnostic » du navigateur permet d'accéder à un utilitaire de configuration et de diagnostic.

Post-processeur STEP5

Module de communication



Paramétrage du module de communication

Structure de l'application

Le langage STEP5 de SIEMENS est organisé en blocs de programmes et de données. Les applications AUTOMGEN traduites par le post-processeur STEP5 sont découpées en plusieurs blocs. Par défaut, le post-processeur utilise les blocs suivants :

- OB1 bloc d'organisation : ce bloc appelle tous les blocs qui doivent être traités de façon cyclique.
- OB20, OB21, OB22 : blocs exécutés au démarrage de l'automate. Ces blocs arment un bit pour activer les étapes initiales de Grafset.

Des blocs PB sont utilisés pour le traitement des prédispositions, pour gérer l'évolution des variables booléennes et des temporisations.

Des blocs FB ou FX sont utilisés pour le code issu de l'application et pour le code écrit dans les fichiers « .SRT » et « .END ». Un bloc FB ou FX est créé pour chaque folio de l'application.

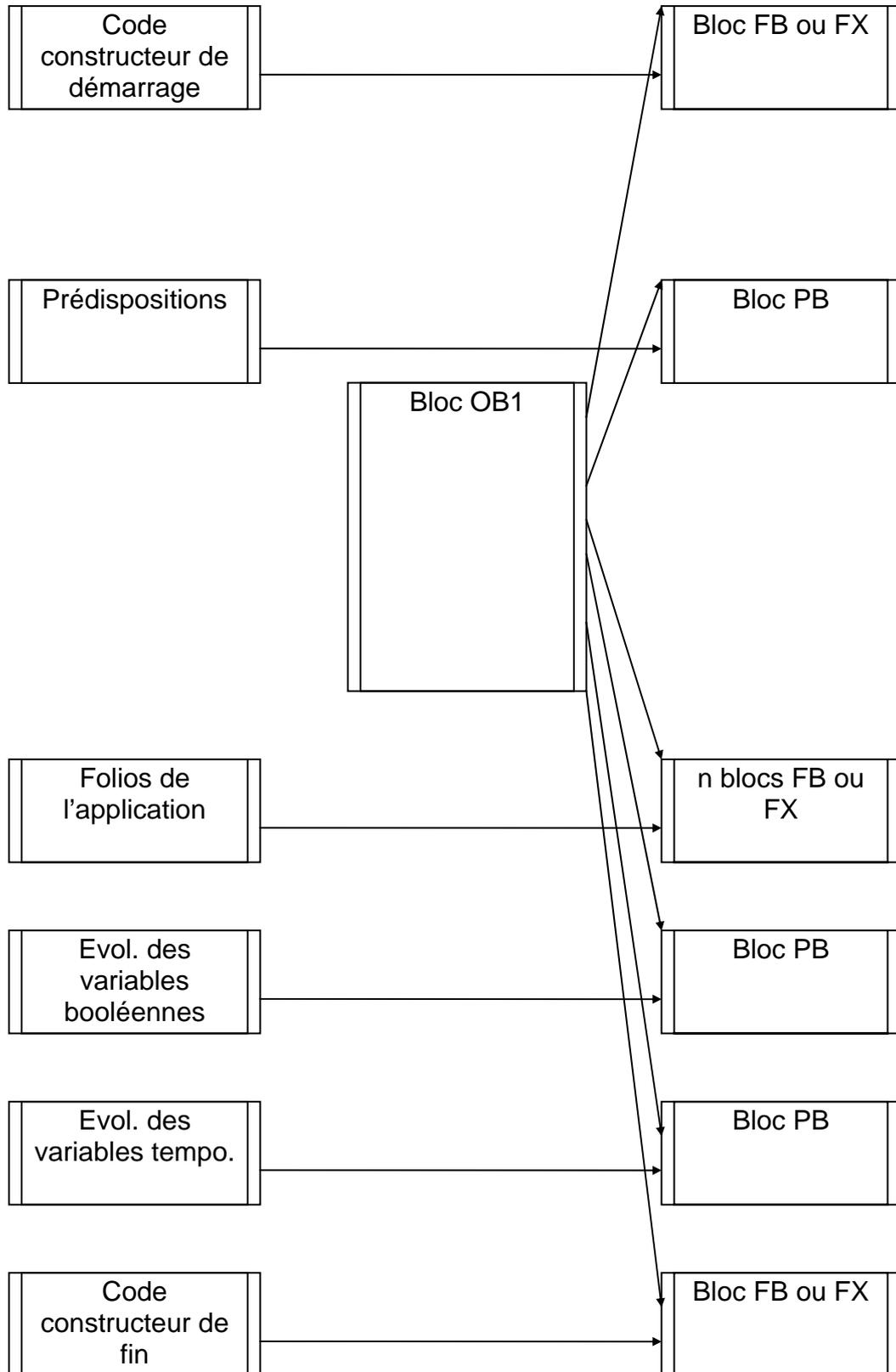
De plus, des folios peuvent directement être associés à un bloc de codes ou de données.

Si un volume de code trop important est généré pour un bloc (code issu d'un folio contenant un programme volumineux par exemple), alors le post-processeur utilise automatiquement un nouveau bloc.

Par défaut, le post-processeur utilise selon ses besoins les blocs PB1 à PB255 et FB1 à FB239.

Ces valeurs peuvent être modifiées (voir chapitre Choix des blocs de programmes à utiliser).

La figure suivante illustre la structure du code généré par le post-processeur SIEMENS :



Choix des blocs de programmes à utiliser

Par défaut, les blocs PB 1 à PB 255 et FB 1 à FB 239 sont utilisés. Trois éléments de configuration permettent de choisir d'autres blocs.

Options de génération de code (attention, modifier avec précaution)	
Type de CPU	95
Optimiser le code généré	Oui
Ne pas générer le code d'évolution des étapes Grafcet	Non
Ne pas générer le code d'évolution des bits utilisateur	Non
Utiliser un seul bit automate pour chaque bit utilisateur AUTOMGEN	Non
Bloc de données utilisé pour les variables AUTOMGEN	DB3
Blocs programme	FB1-199
Blocs programme auxiliaires	PB1-255
Blocs pour les sous-programmes	FP200-239

Choix du bloc de données

Par défaut le bloc DB 3 est utilisé pour les variables numériques. Cette directive permet d'utiliser un autre bloc.

Options de génération de code (attention, modifier avec précaution)	
Type de CPU	95
Optimiser le code généré	Oui
Ne pas générer le code d'évolution des étapes Grafcet	Non
Ne pas générer le code d'évolution des bits utilisateur	Non
Utiliser un seul bit automate pour chaque bit utilisateur AUTOMGEN	Non
Bloc de données utilisée pour les variables AUTOMGEN	DB3
Blocs programme	FB1-199
Blocs programme auxiliaires	PB1-255
Blocs pour les sous-programmes	FP200-239

Le changement de bloc de données implique deux autres modifications :

- dans le code constructeur de démarrage, il faut créer le bloc de données correspondant,
- il faut sélectionner le bloc de données choisi dans le paramétrage du module de dialogue.

Choix du type de processeur

Options de génération de code (attention, modifier avec précaution)	
Type de CPU	95
Optimiser le code généré	Oui
Ne pas générer le code d'évolution des étapes Grafcet	Non
Ne pas générer le code d'évolution des bits utilisateur	Non
Utiliser un seul bit automate pour chaque bit utilisateur AUTOMGEN	Non
Bloc de données utilisé pour les variables AUTOMGEN	DB3
Blocs programme	FB1-199
Blocs programme auxiliaires	PB1-255
Blocs pour les sous-programmes	FP200-239

Association du code écrit sur un folio à un bloc programme

En écrivant du code littéral bas niveau ou du code constructeur écrit dans un organigramme sur un folio de type « Tâche », on associe ce code à un bloc STEP5.

Le numéro de tâche détermine le type et le numéro du bloc.

Le code généré par ce folio doit tenir compte du type du bloc et des instructions utilisables dans ce type de bloc (jeu d'instructions limité dans les blocs OB et PB).

La table ci-dessous donne la correspondance entre la valeur de « n » et le bloc.

Numéro de tâche	Bloc STEP5
0 à 255	OB 0 à OB 255
256 à 511	PB 0 à PB 255
512 à 767	FB 0 à FB 255
768 à 1023	FX 0 à FX 255
1024 à 1279	DB 0 à DB 255
1280 à 1535	DX 0 à DX 255

Syntaxes spécifiques

Définition de blocs

La directive « \$BLOC <type de bloc> <numéro> » permet de définir le début d'un bloc de programme ou de données.

Le type de bloc peut être « OB », « FB », « FX », « PB » pour le code ou « DB », « DX » pour les données.

Le numéro du bloc est une valeur comprise entre 0 et 255. Les blocs « FX » et « DX » ne peuvent être utilisés que sur les automates 135U et 155U. La directive « BE » marque la fin d'un bloc.

Exemple :

\$BLOC DB1

...

\$BE

\$BLOC OB1

...

\$BE

les directives « KH= », « KY= », « KC= », « KM= » et « KF= » insèrent des constantes dans les blocs de données DB.

- « KH= » insère une constante 16 bits exprimée en hexadécimal.
- « KY= » insère une constante 16 bits exprimée sous la forme de deux valeurs comprises entre 0 et 255 séparées par une virgule.
- « KC= » insère une suite de caractères entourés par des caractères « ' » (apostrophe).
- « KM= » insère une constante 16 bits exprimée en binaire.
- « KF= » insère une constante 16 bits exprimée en décimal signé.

Exemple :

\$BLOC DB 4

KH= 1234

KY=100,5

KC= 'Ceci est un texte'

KM=11111111 00000000

KF=-200

\$BE

Blocs FB et FX

Les blocs FB et FX du langage SIEMENS peuvent recevoir des paramètres.

Appel

Si des paramètres doivent être passés à un bloc fonctionnel, alors il faut utiliser la syntaxe suivante :

- l'appel doit être suivi du caractère « * »,
- la ligne suivante doit contenir une instruction de saut « SPA » vers la ligne située au-delà des paramètres,
- les lignes suivantes doivent contenir les paramètres précédés d'un mnémotique « U » pour les bits ou « L » pour les mots. Les constantes doivent être écrites sous la forme « Kx=valeur » (x est le type de constante voir le chapitre Définition de blocs).

Exemple d'appel à un bloc fonctionnel sans paramètre :

```
SPA FB 5
```

Exemples d'appel de bloc fonctionnel avec paramètres :

```
SPA FB 242*  
SPA=_label_  
L MW10      ; premier paramètre  
L MW12      ; deuxième paramètre  
U M0.0      ; troisième paramètre  
L MW14      ; quatrième paramètre  
L MW16      ; cinquième paramètre  
@label  
SPA FB200*  
SPA=_label2_  
KY=0,4      ; Exemple de paramètre constant  
@label2
```

Écriture

Dans les blocs FB et FX, il faut utiliser les mnémoniques se terminant par le caractère '=' suivi d'un numéro de paramètre (1=premier paramètre).

Exemple de bloc fonctionnel à deux paramètres (recopie du premier paramètre dans le deuxième) :

```
$BLOC FB 100  
L=1  
T=2  
$BE
```

Post-processeur TSX 07

Ce post-processeur permet de programmer les automates MODICON TELEMECANIQUE SCHNEIDER TSX 07.

Module de communication

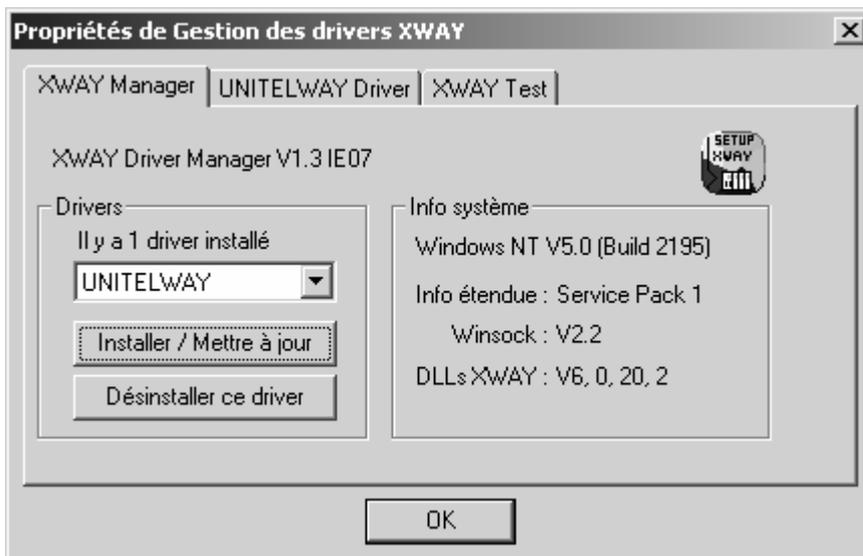


Le driver UNITELWAY SCHNEIDER doit impérativement être installé sur l'ordinateur (en local) pour pouvoir communiquer avec les automates SCHNEIDER TSX 07. Des drivers adaptés à une ou plusieurs versions de WINDOWS se trouvent sur le CD-ROM et peuvent être téléchargés sur le site d'IRAI : www.irai.com. Attention, certaines versions de WINDOWS sont incompatibles avec certains type d'automates TSX 07 (TSX 0720... et TSX 0721... incompatibles avec WINDOWS ME, WINDOWS 2000 ou WINDOWS XP).

Le module de communication utilise le driver de communication conçu par SCHNEIDER AUTOMATION. Cliquer sur « Paramétrer et tester ... » vous permet d'accéder directement au menus du driver de communication SCHNEIDER.



Paramétrage du module de communication

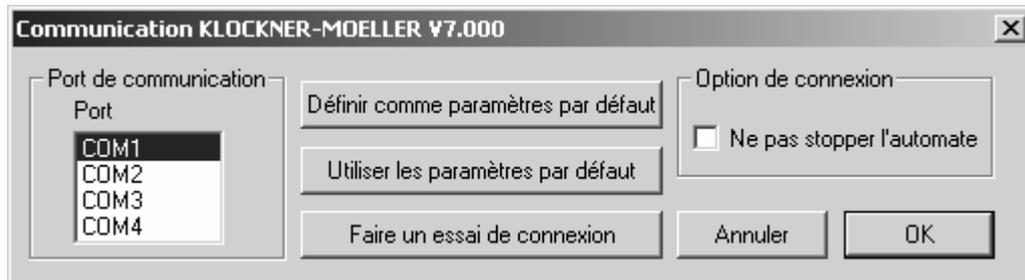


Propriétés du module de communication UNITELWAY

Post-processeur PS3-PS4

Ce post-processeur permet de programmer les automates KLOCKNER-MOELLER PS3 et PS4-100.

Module de communication

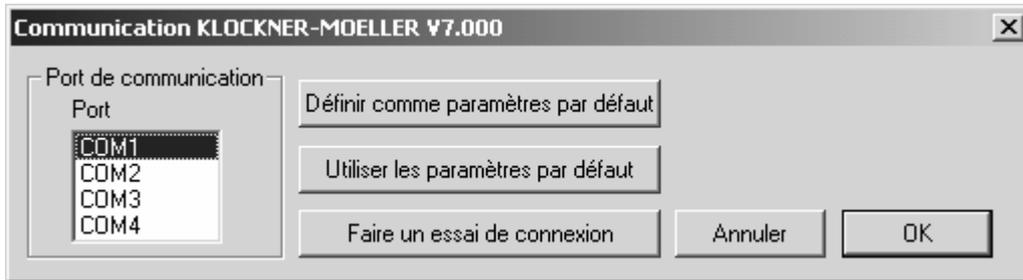


Paramétrage du module de communication

Post-processeur PS4

Ce post-processeur permet de programmer les automates MOELLER PS4-200, PS4-300 et PS416. Le logiciel SUCOSOFT S40 V5 ou supérieure de MOELLER doit être utilisé (la version de démonstration de ce logiciel peut être utilisée).

Module de communication



Paramétrage du module de communication

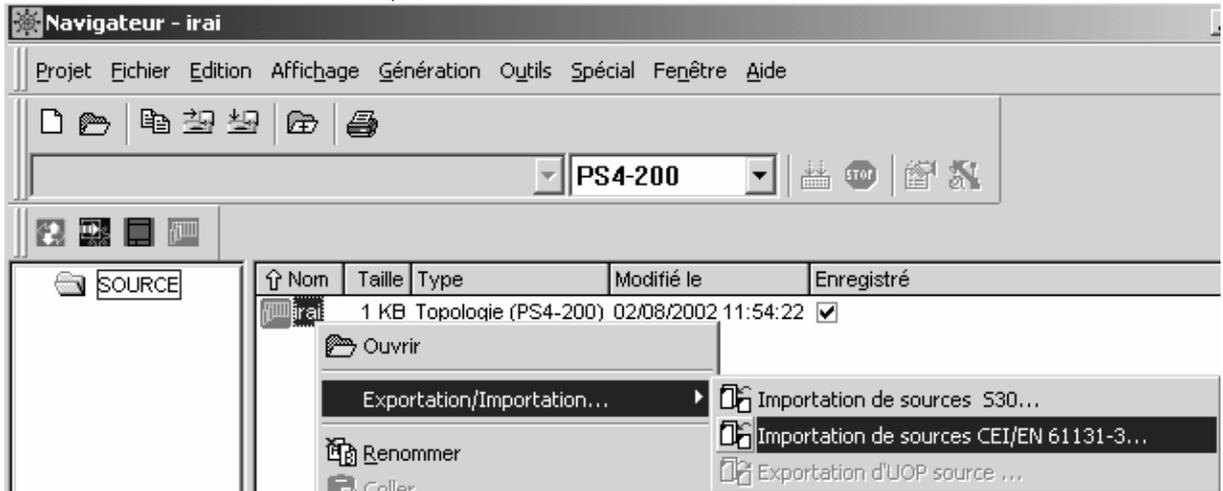
Transfert des programmes vers le logiciel SUCOSOFT S40 de MOELLER

Dans l'élément ci-dessous, désignez un fichier qui servira d'échange entre AUTOMGEN et SUCOSOFT. A la fin de la compilation dans AUTOMGEN, ce fichier sera généré.

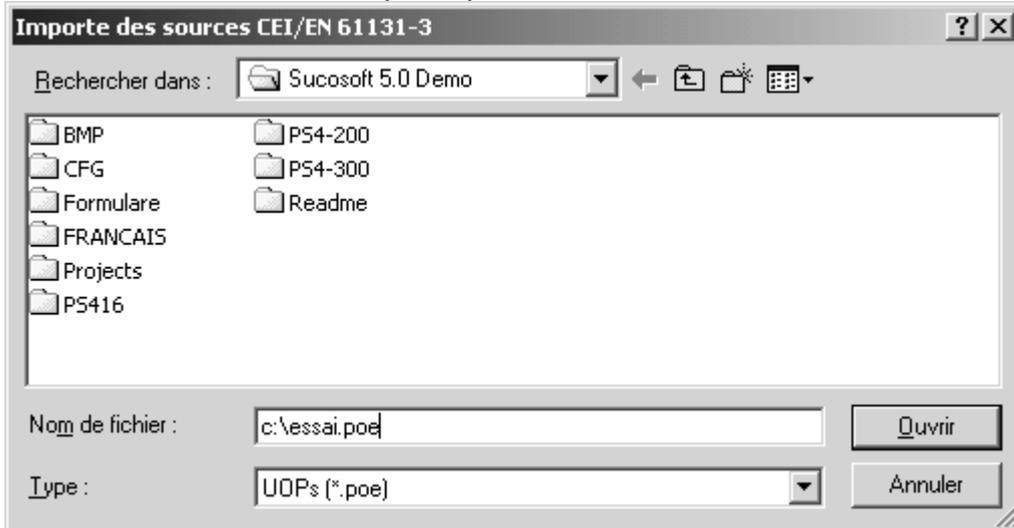
+	Configuration matérielle	
-	Options de génération de code (attention, modifier avec précaution)	
	Optimiser le code généré	Oui
	Ne pas générer le code d'évolution des étapes Grafcet	Non
	Ne pas générer le code d'évolution des bits utilisateur	Non
	Utiliser un seul bit automate pour chaque bit utilisateur AUTOMGEN	Non
	Fichier à importer dans le logiciel MOELLER après compilation	c:\essai.poe

Démarche à suivre pour importer le fichier généré par AUTOMGEN dans le logiciel MOELLER puis l'injecter dans l'automate

- lancez SUCOSOFT,
- créez un nouveau projet,
- cliquez avec le bouton droit de la souris sur la configuration topologique dans SUCOSOFT et choisissez l'option « Exportation/Importation / Importation de sources CEI/EN 61131-3 »,



- entrez le nom du fichier exporté par AUTOMGEN,



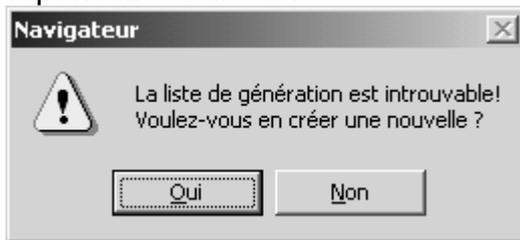
- double cliquez sur l'élément « _MAIN » qui vient d'apparaître dans la liste,

↑	Nom	Taille	Type	Modifié le	Enregi:
+	MAIN	1 KB	Programme	05/08/2002 10:09:30	<input checked="" type="checkbox"/>
	irai	1 KB	Topologie (PS4-200)	02/08/2002 11:54:22	<input checked="" type="checkbox"/>

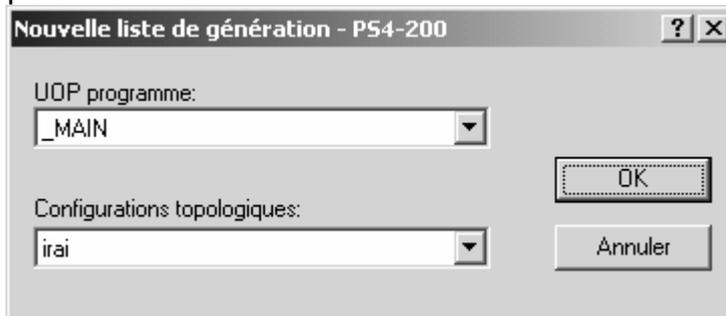
- dans le menu « Spécial », choisissez « Génération de code »,



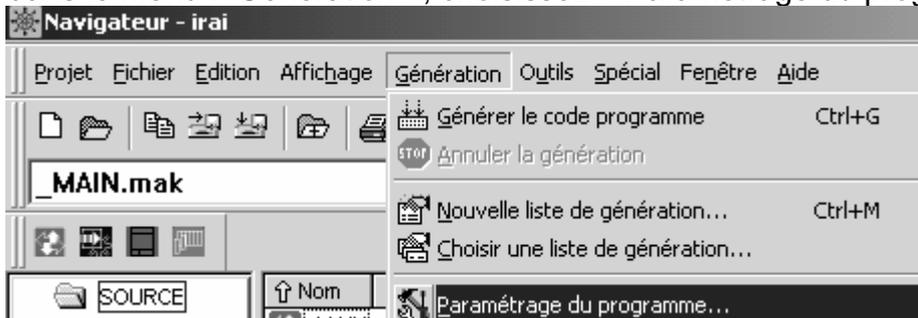
- répondez « Oui » à :



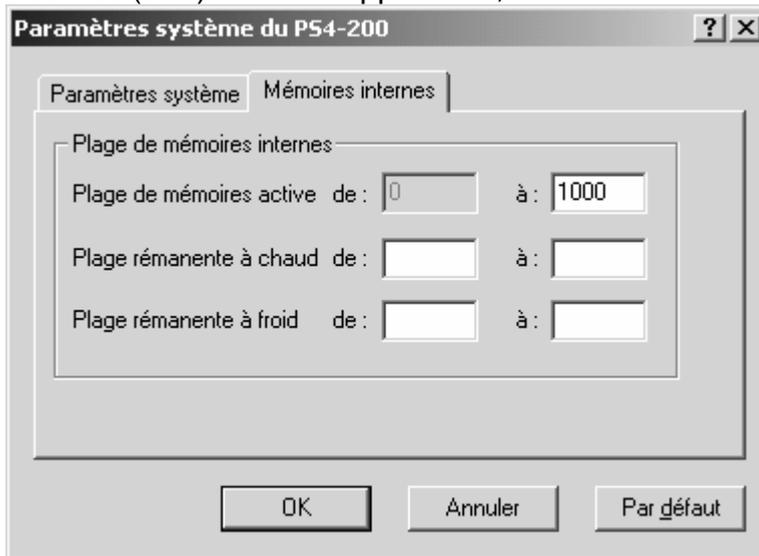
- puis « OK » à :



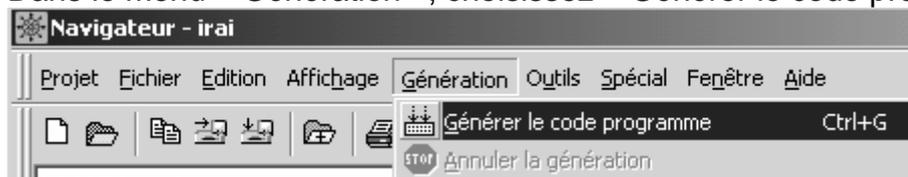
- dans le menu « Génération », choisissez « Paramétrage du programme ... »,



- Sélectionner une taille appropriée pour stocker l'ensemble des variables internes (%M) de votre application,



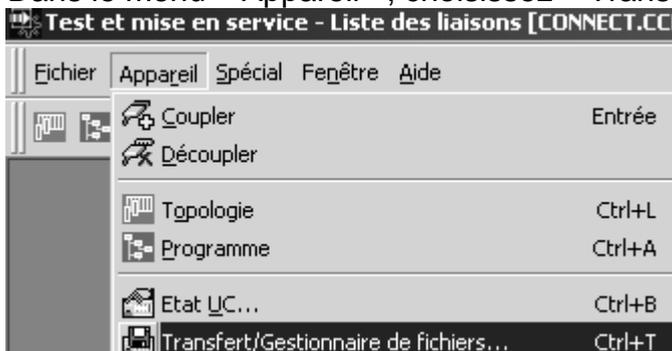
- Dans le menu « Génération », choisissez « Générer le code programme » ,



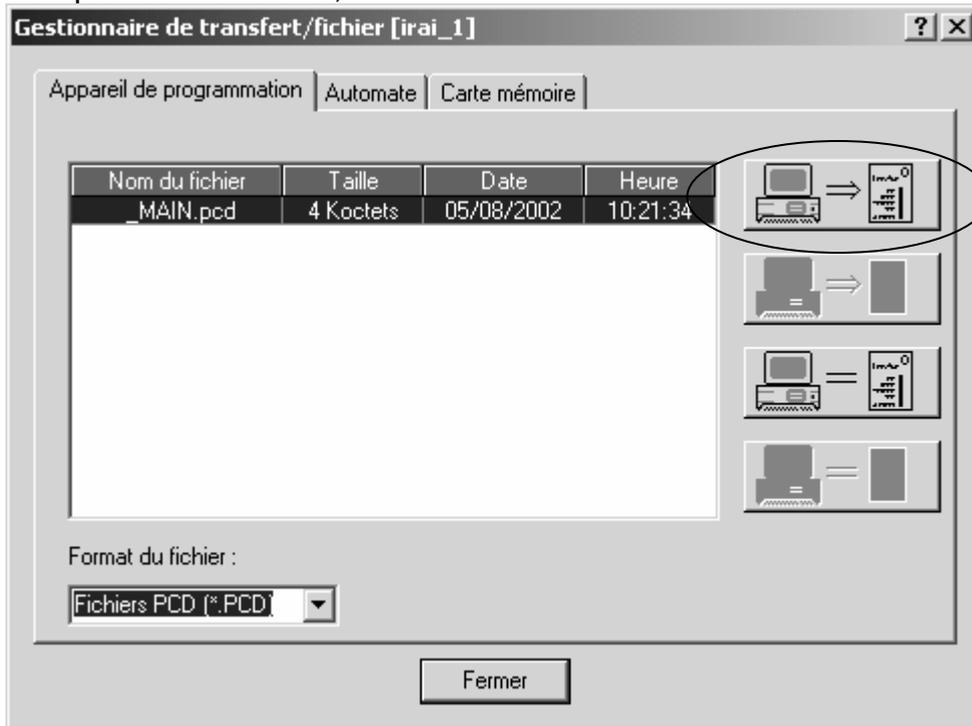
- Si il n'y a pas d'erreur de compilation, vous pouvez procéder au transfert de l'application vers l'automate. Dans le menu « Outils », choisissez « Test et mise en service » ,



- Dans le menu « Appareil », choisissez « Transfert / Gestionnaire de fichiers » ,



- Cliquez sur transférer,



- A la fin du transfert, déconnecter le logiciel SUCOSOFT pour pouvoir connecter AUTOMGEN à l'automate et activer le mode de mise au point dynamique.

Post-processeur RPX

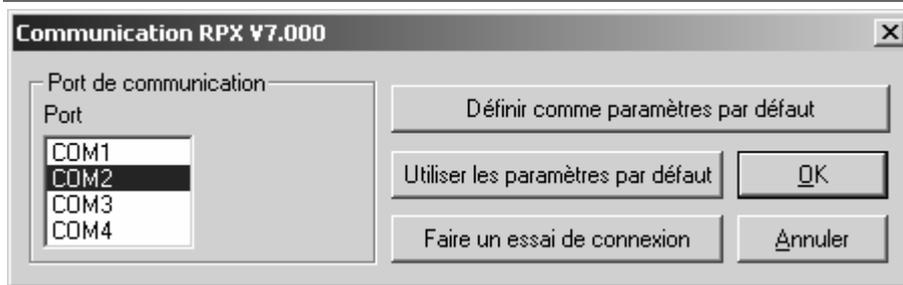
Ce post-processeur permet de programmer les automates CROUZET RPX.

Choix du type d'automate

L'élément « Configuration / Post-processeur / RPX / Système / Configuration matérielle » du navigateur permet de choisir le type de l'automate.

Configuration matérielle	
Type de l'automate	10
Module d'extension présent	NO

Module de communication



Paramétrage du module de communication

Utilitaire

L'élément « Configuration / Post-processeur / RPX / Terminal emulator » du navigateur permet d'accéder à un émulateur de terminal utilisable pour configurer les coupleurs de communication de l'automate.

Post-processeur PL71

Ce post-processeur permet de programmer les automates SCHNEIDER TSX 17-10 et TSX 17-20 (sans cartouche PL72).

Choix du type d'automate

L'élément « Configuration / Post-processeur / PL71 / Système / Configuration matérielle » du navigateur permet de choisir le type de l'automate.

Configuration matérielle		
Type de l'automate	1720	automate TSX17-20 sans cartouche PL72

Module de communication

Communication PL71 V7.000

Port de communication

Port	Vitesse
COM1	300
COM2	600
COM3	1200
COM4	2400

Mode

Prise console Esclave

Unitelway 14

Options

Toujours essayer une connexion à 19200 bauds

Définir comme paramètres par défaut

Utiliser les paramètres par défaut

Faire un essai de connexion

OK Annuler

Paramétrage du module de communication

(Pour plus d'informations, reportez-vous à la configuration du Module de communication PL72).

Tâche compteur rapide

Un folio de type tâche portant le numéro 1 sera associé à la tâche compteur rapide de l'automate.

Exemples spécifiques

Ces exemples se trouvent dans le répertoire « <répertoire d'installation d'AUTOMGEN> / Exemples / Post-processeurs / PL71 ». Les fichiers portent le même nom que le titre des chapitres qui suivent.

Comptage

Les incréments et décréments de compteurs en PL71 étant limités (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 (voir le contenu de l'exemple)

Compteur rapide

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

Post-processeur PB

Ce post-processeur permet de programmer les automates SCHNEIDER APRIL PB.

Choix du type d'automate

L'élément « Configuration / Post-processeur / PB / Système / Configuration matérielle » du navigateur permet de choisir le type de l'automate.



Module de communication



Paramétrage du module de communication

Syntaxes spécifiques

La directive « \$ORG=xxxx » permet de définir le début de l'adresse d'assemblage, au départ l'adresse d'assemblage est fixée à 0C30,

Exemple :

\$ORG=1C30

La directive « \$TOP=xxx » définit l'adresse maximale pour le saut de page. Elle précise les trois digits de poids faible des adresses au dessus desquelles un saut de page sera automatiquement généré par l'assembleur.

La directive « \$CONST=xxxx,yyyy » définit l'adresse de départ et de fin pour le stockage des constantes. Les constantes sont logées dans une table en dehors du programme.

La directive « WORD xxxx » insère la valeur xxxx (quatre digits en hexadécimal) dans le programme.

La directive « ADR xxxx » insère l'adresse de la variable xxxx (quatre digits en hexadécimal) dans le programme.

La syntaxe #nnnn permet de faire référence à une valeur constante.

Par exemple :

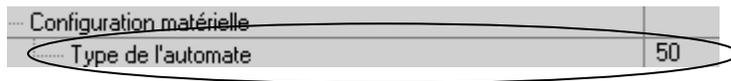
apl #1234 ; place la constante 1234 (hexadécimal) dans l'accumulateur.

Post-processeur SMC

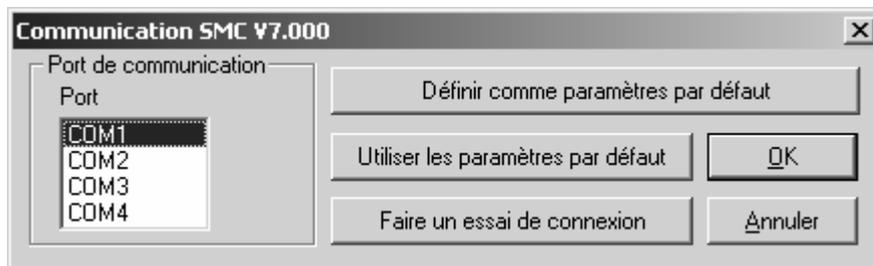
Ce post-processeur permet de programmer les automates SCHNEIDER APRIL SMC.

Choix du type d'automate

L'élément « Configuration / Post-processeur / SMC / Système / Configuration matérielle » du navigateur permet de choisir le type de l'automate.



Module de communication



Paramétrage du module de communication

Syntaxes spécifiques

La directive « \$SEQ » marque le début d'une zone booléenne.

La directive « \$CAL » débute une zone de calcul.

La directive « \$PRD » débute une zone de prédisposition de variables.

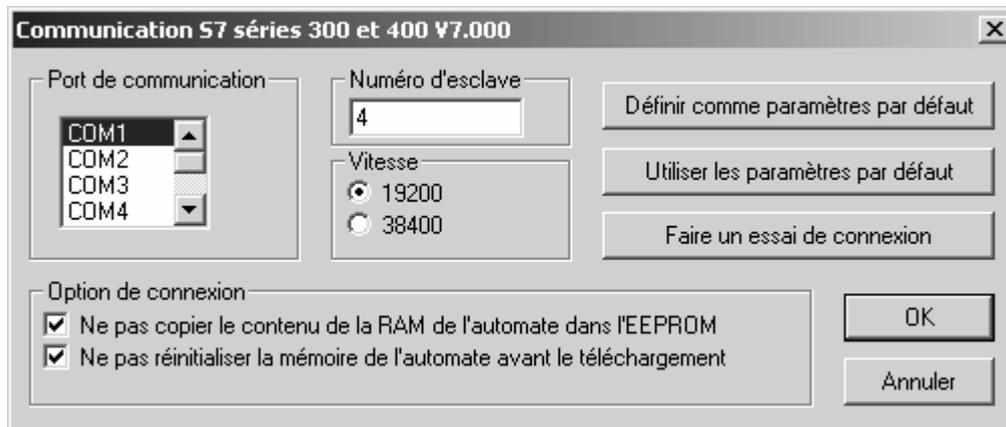
Les variables booléennes peuvent être utilisées en bistable ou en monostable indépendamment des conventions du langage SMC. Le caractère « ! » placé après le signe « = » force la variable à être bistable (mise à un ou mise à zéro), le caractère « ? » placé après le signe « = » force la variable à être monostable (affectation ou affectation complémentée).

La syntaxe « SS.ccccccc » permet d'écrire une séquence de sécurité (nécessaire sur les automates SMC 25 et 600), « ccccccc » représente un nom de programme sur 8 caractères maximum.

Post-processeur S7300

Ce post-processeur permet de programmer les automates SIEMENS S7300.

Module de communication



Paramétrage du module de communication

Le numéro d'esclave doit être en accord avec celui paramétré dans l'automate.

Syntaxes spécifiques

La directive « \$BLOC <type de bloc> <numéro> » permet de définir le début d'un bloc de programme ou de données.

Le type de bloc peut être « OB », « FB », « FC », « SFC », « SFB », pour le code ou « DB » pour les données. Le numéro du bloc est une valeur comprise entre 0 et 65535. La directive « \$ENDBLOC » marque la fin d'un bloc.

Exemple :

\$BLOC DB1

...

\$ENDBLOC

\$BLOC OB1

...

\$ENDBLOC

La syntaxe suivante permet de déclarer les variables pour un bloc :

Pour une variable d'entrée :

\$VAR-nature type {:=initialisation}

ou

\$VAR-nature symbole : {type :=initialisation}

« nature » peut être :

- « IN » pour une variable d'entrée,
- « OUT » pour une variable de sortie,
- « INOUT » pour une variable d'entrée / sortie,
- « TEMP » pour une variable temporaire,
- « STAT » pour une variable statique.

« type » peut être un des types de variable du langage STEP7 : BOOL, INT, WORD, STRUCT, ENDSTRUCT, etc ...

« symbole » permet d'associer un mnémonique à une variable.

« initialisation » est optionnel et fixe la valeur par défaut d'une variable.

Les blocs DB n'autorisent que des variables de type statique.

Les blocs OB n'autorisent que des variables de type temporaire.

Les blocs FC et SFC n'autorisent pas des variables de type statique.

Comme dans le logiciel SIEMENS, les déclarations de variables doivent apparaître dans l'ordre suivant : entrée, sortie, entrée / sortie, statique et temporaire.

Définition des variables d'un bloc

La syntaxe « £D bloc déclaration » permet de définir une déclaration associée à un bloc particulier. Lorsque ce bloc est généré par le compilateur, alors la déclaration est utilisée.

Appel des blocs

La syntaxe « CALL nom de bloc {,DB d'instance} (liste des paramètres) » permet d'appeler un bloc FC, FB, SFC ou SFB.

Exemple :

```
$BLOC FC1
$VAR-IN entree1 :BOOL :=FALSE ;
$VAR-IN entree2 :BOOL :=FALSE ;
$VAR-OUT sortie :BOOL ;
u_entree1_
u_entree2_
=_sortie_
$ENDBLOC

$BLOC OB1
CALL FC1(_entree1_ :=e0.0,_entree2_ :=e0.1,_sortie_ :=a0.0)
$ENDBLOC
```

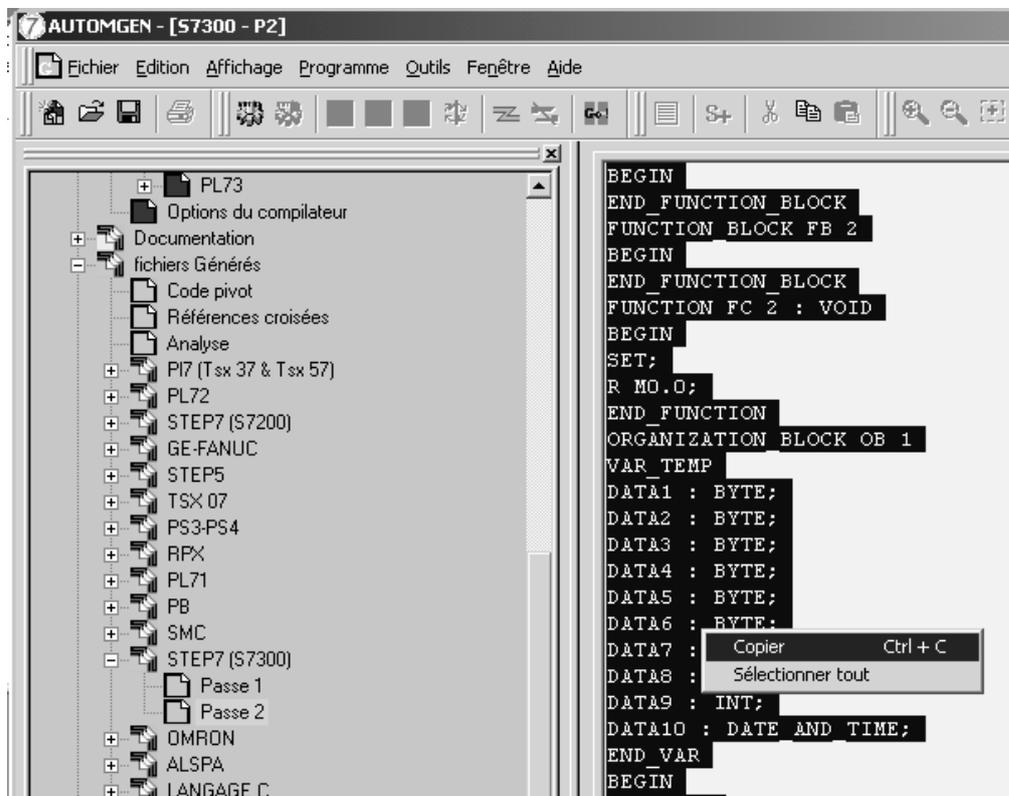
Importation dans le logiciel SIMATIC de SIEMENS

Pour importer le code généré par AUTOMGEN dans le logiciel SIMATIC de SIEMENS, suivre la procédure suivante :

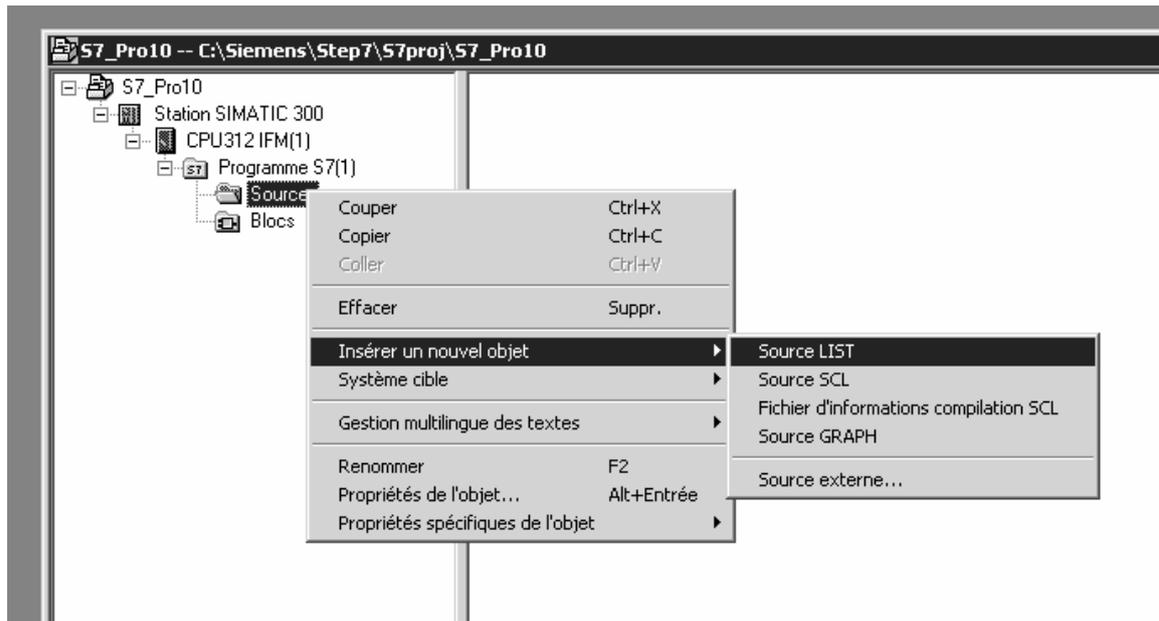
- 1- Dans la partie « Système » de la configuration du post-processeur S7300, sélectionnez SIMATIC dans l'élément suivant

Eléments	Valeurs	Commentaires
Options de génération de code (attention, modifier avec précaution)		
Optimiser le code généré	Non	
Ne pas générer le code d'évolution des étapes Grafset	Non	
Ne pas générer le code d'évolution des bits utilisateur	Non	
Utiliser un seul bit automate pour chaque bit utilisateur AUTOMGEN	Non	
Blocs programme	FC1-239	
Blocs programme auxiliaires	FB1-255	
Génération de code	SIMATIC	Génération d'un source pour SIMATIC de SIEMENS
Détection de variables		

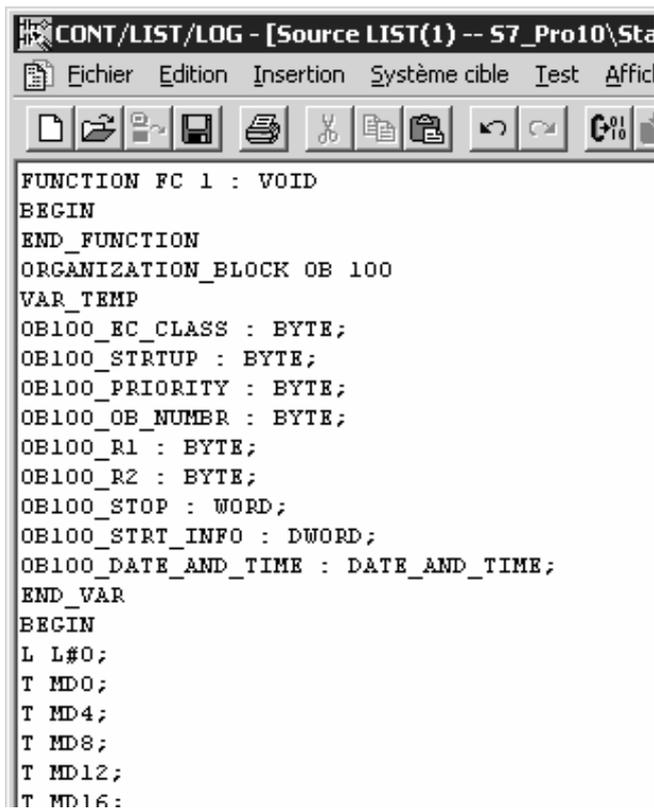
- 2- Compilez l'application,
- 3- Dans AUTOMGEN, ouvrez l'élément « Code généré / Post-processeur S7300 / Passe 2 », sélectionnez l'ensemble des lignes puis la commande « Copier » dans le menu « Edition ».



- 4- Dans le logiciel SIMATIC, créez un élément de type « Source LIST ».



5- Dans SIMATIC, collez le code dans la fenêtre contenant le source LIST avec la commande « Coller » du menu « Edition »,



```

FUNCTION FC 1 : VOID
BEGIN
END_FUNCTION
ORGANIZATION_BLOCK OB 100
VAR_TEMP
OB100_EC_CLASS : BYTE;
OB100_STARTUP : BYTE;
OB100_PRIORITY : BYTE;
OB100_OE_NUMBR : BYTE;
OB100_R1 : BYTE;
OB100_R2 : BYTE;
OB100_STOP : WORD;
OB100_STRT_INFO : DWORD;
OB100_DATE_AND_TIME : DATE_AND_TIME;
END_VAR
BEGIN
L L#0;
T MD0;
T MD4;
T MD8;
T MD12;
T MD16;

```

6- Dans SIMATIC, compilez le source en cliquant sur .

L'importation est alors terminée.

Structure du code généré

Le langage STEP7 de SIEMENS est organisé en blocs de programmes et de données. Les applications AUTOMGEN traduites par le post-processeur STEP7 sont découpées en plusieurs blocs. Par défaut, le post-processeur utilise les blocs suivants :

- OB1 bloc d'organisation : ce bloc appelle tous les blocs qui doivent être traités de façon cyclique,
- OB100 : blocs exécutés au démarrage de l'automate. Ce bloc arme un bit pour activer les étapes initiales de Grafset.

Des blocs FB sont utilisés pour le traitement des prédispositions, pour gérer l'évolution des variables booléennes et des temporisations.

Des blocs FC sont utilisés pour le code issu de l'application et pour le code constructeur de démarrage et de fin.

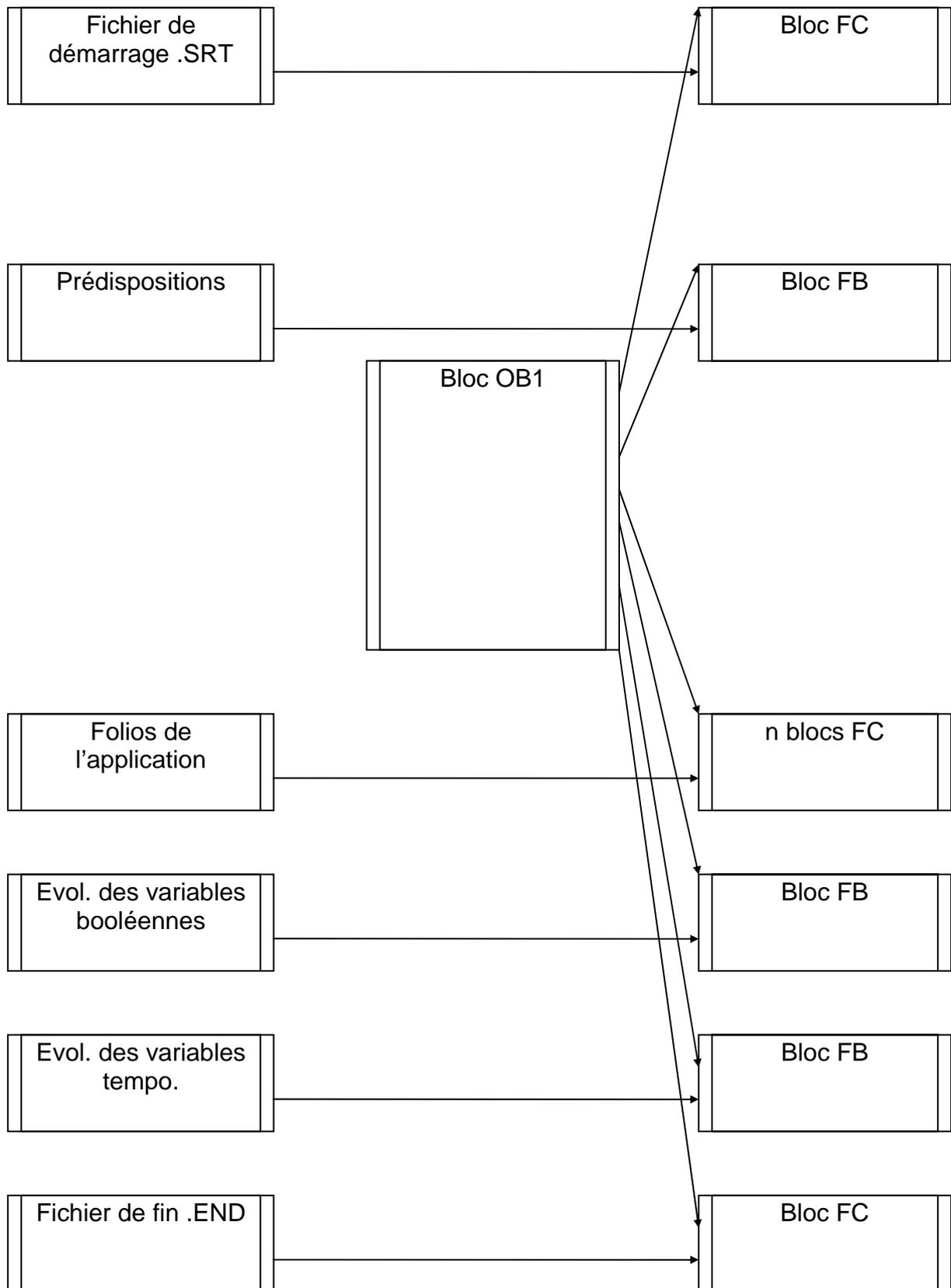
Un bloc FB pour chaque folio de l'application.

De plus, des folios peuvent directement être associés à un bloc (voir chapitre Association du code écrit sur un folio à un bloc programme).

Par défaut, le post-processeur utilise selon ses besoins les blocs FB1 à FB255 et FC1 à FC239.

Ces valeurs peuvent être modifiées dans l'élément « Système » de la configuration.

La figure suivante illustre la structure du code généré par le post-processeur SIEMENS :



Choix des blocs de programmes à utiliser

Par défaut, les blocs FC 1 à FC 239 et FB 1 à FB 255 sont utilisés. Deux éléments de configuration permettent de choisir d'autres blocs.

Options de génération de code (attention, modifier avec précaution)	
Optimiser le code généré	Non
Ne pas générer le code d'évolution des étapes Grafcet	Non
Ne pas générer le code d'évolution des bits utilisateur	Non
Utiliser un seul bit automate pour chaque bit utilisateur AUTOMGEN	Non
Blocs programme	FC1-239
Blocs programme auxiliaires	PB1-255

Association du code écrit sur un folio à un bloc programme

En écrivant du code littéral bas niveau ou du code constructeur écrit dans un organigramme sur un folio de type « Tâche », on associe ce code à un bloc STEP7.

Le numéro de tâche détermine le type et le numéro du bloc.

Le code généré par ce folio doit tenir compte du type du bloc et des instructions utilisables dans ce type de bloc (jeu d'instructions limité dans les blocs OB et PB).

La table ci-dessous donne la correspondance entre le numéro de tâche et le bloc STEP7.

Numéro de tâche	Bloc STEP7
0 à 255	OB 0 à OB 255
256 à 511	FC 0 à FC 255
512 à 767	FB 0 à FB 255

Exemples spécifiques

Ces exemples se trouvent dans le répertoire « <répertoire d'installation d'AUTOMGEN> / Exemples / Post-processeurs / S7300 ». Les fichiers portent le même nom que les titres des chapitres qui suivent.

Appel d'un bloc STEP7

Exemple d'appel des blocs fonctions STEP7.

Utilisation d'un bloc OB

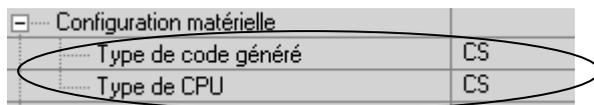
Exemple d'association du code écrit sur un folio à un bloc OB.

Post-processeur OMRON

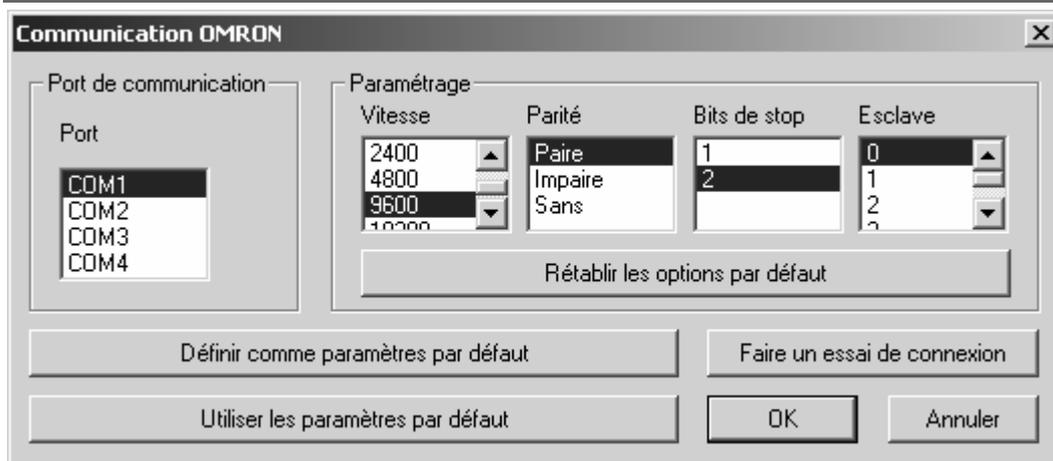
Ce post-processeur permet de programmer les automates OMRON série C, CS et CV. La génération directe des fichiers binaire et le transfert des applications dans l'automate sont supportés pour les automates série C. Pour les automates séries CS et CV, le logiciel CX-PROGRAMMER d'OMRON version 2.0 ou supérieur doit être utilisé.

Choix du type d'automate

L'élément « Configuration / Post-processeur / OMRON / Système / Configuration matérielle » du navigateur permet de choisir le type de l'automate.



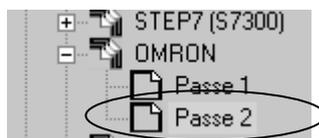
Module de communication



Paramétrage du module de communication

Transfert des applications dans le logiciel CX-PROGRAMMER

- dans AUTOMGEN, à la fin de la compilation, double cliquez sur l'élément « Fichiers générés / OMRON / passe 2 »,



- sélectionnez l'ensemble des lignes,

```
LD A20011
BSET #0 HO H511
LD 0
RSET H1
LD A20011
SET H1
LD H2
AND 1
SET H1
LD 1
RSET H3
LD HO
AND 0
SET H3
LD H1
OUT 1000
LD H3
OUT 1001
LD H1
OUT HO
LD H3
OUT H2
END
```

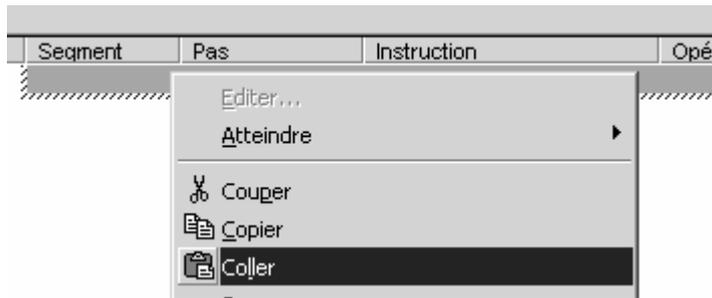
- sélectionnez la commande « Copier » du menu « Edition »,



- dans CX-PROGRAMMER, créer une application vierge, affichez la zone programme sous forme de mnémoniques,



- sélectionnez la zone programme puis collez les lignes,



Segment	Pas	Instruction	Opérande
0	0	LD	P_First_Cycle
	1	BSET(071)	#0
			H0
			H511
1	2	LD	0.00
	3	RSET	H0.01
2	4	LD	P_First_Cycle
	5	SET	H0.01
3	6	LD	H0.02
	7	AND	0.01
	8	SET	H0.01
4	9	LD	0.01
	10	RSET	H0.03
5	11	LD	H0.00
	12	AND	0.00
	13	SET	H0.03
6	14	LD	H0.01
	15	OUT	10.00
7	16	LD	H0.03
	17	OUT	10.01
8	18	LD	H0.01
	19	OUT	H0.00
9	20	LD	H0.03
	21	OUT	H0.02
10	22	END(001)	

Vous pouvez télécharger l'application dans l'automate à partir de CX-PROGRAMMER puis revenir dans AUTOMGEN pour réaliser la mise au point du programme en mode connecté (pensez à déconnecter CX-PROGRAMMER de l'automate pour pouvoir communiquer à partir d'AUTOMGEN).

Syntaxe spécifique

La syntaxe suivante permet de fixer la valeur d'un mot de donnée :

\$DMn=valeur

« n » est le numéro du mot,

« valeur » est une valeur de 16 bits exprimée par défaut en décimal, ou en hexadécimal si elle est précédée du caractère 'H'.

Exemple :

\$DM10=50
\$DM100=HA000

Association du code écrit sur un folio à un bloc programme

En écrivant du code littéral bas niveau ou du code constructeur écrit dans un organigramme sur un folio de type « Tâche », on associe ce code à une tâche d'interruption. Le numéro de tâche est équivalent au numéro d'interruption.

Exemple spécifique

Cet exemple se trouve dans le répertoire « <répertoire d'installation d'AUTOMGEN> / Exemples / Post-processeurs / S7200 ». Le fichier porte le même nom que le titre du chapitre qui suit.

Tâche d'interruption

Exemple d'appel d'une tâche d'interruption

Post-processeur ALSPA

Ce post-processeur permet de programmer les automates CEGELEC ALSPA C50 et C100.

Module de communication

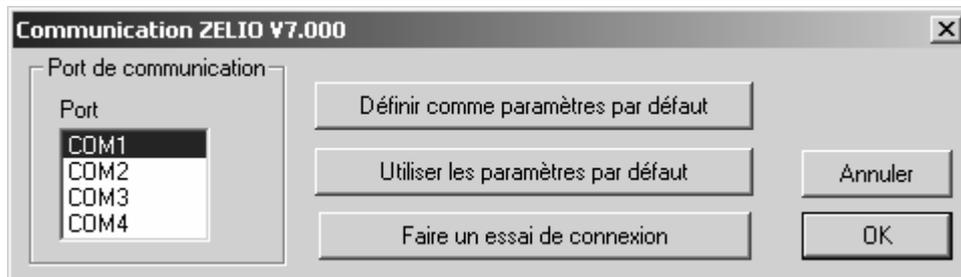


Paramétrage du module de communication

Post-processeur ZELIO

Ce post-processeur permet de programmer les modules SECHNEIDER ZELIO.

Module de communication

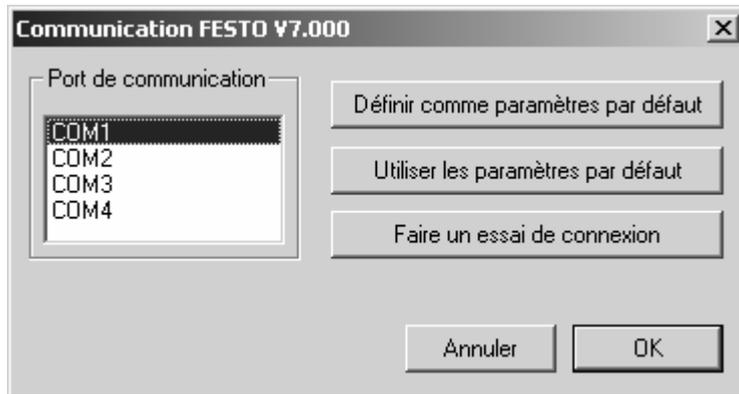


Paramétrage du module de communication

Post-processeur FESTO

Ce post-processeur permet de programmer les automates FPC 101, FPC 103 et FEC FESTO.

Module de communication



Paramétrage du module de communication

Génération de fichier binaire

Les fichiers binaires peuvent être directement générés et téléchargés dans les automates FPC 101 et FPC 103. L'atelier logiciel FST FESTO sous DOS ou sous WINDOWS est nécessaire pour les automates FEC.

Configuration matérielle		
Type de l'automate	FPC103	FPC103

Choix d'un type de CPU (génération directe d'un fichier binaire)

Importation dans les ateliers logiciel FESTO

Configuration matérielle		
Type de l'automate	NO	Génère un source .AWL mais pas de fichier binaire
Configuration logicielle		
Fichier à importer dans le logiciel FESTO après compilation (si type de l'automate=NO)	C:\export.AWL	

Génération d'un fichier .AWL compatible avec les ateliers FESTO

Si vous utilisez l'atelier logiciel FST FESTO sous DOS, relisez le fichier .AWL à partir de ce logiciel.

Si vous utilisez le logiciel FST FESTO sous WINDOWS, ouvrez le fichier généré dans AUTOMGEN en double cliquant sur l'élément « Fichiers générés / FESTO / Passe 2 », sélectionnez l'ensemble du fichier, utilisez la commande « Copier » du menu « Edition » puis utiliser la commande « Coller » dans le logiciel FESTO pour récupérer le code généré.

Transférer le programme vers l'automate avec le logiciel FESTO.

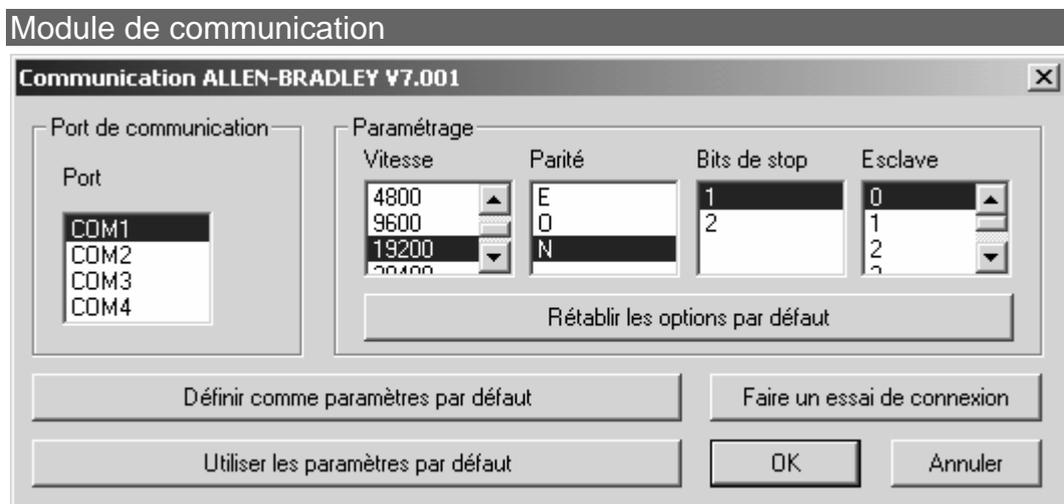
Vous pouvez ensuite vous connecter avec AUTOMGEN (après vous être déconnecté avec le logiciel FESTO) en utilisant comme mode de connexion « Seulement connecter ».

Post-processeur ALLEN-BRADLEY

Ce post-processeur permet de programmer les automates SLC de ROCKWELL. Le logiciel RSLogix 500 V5 ou supérieure de ROCKWELL est nécessaire.



La version STARTER de RSLogix 500 ne permet pas d'importer les fichiers générés par AUTOMGEN.



Paramétrage du module de communication

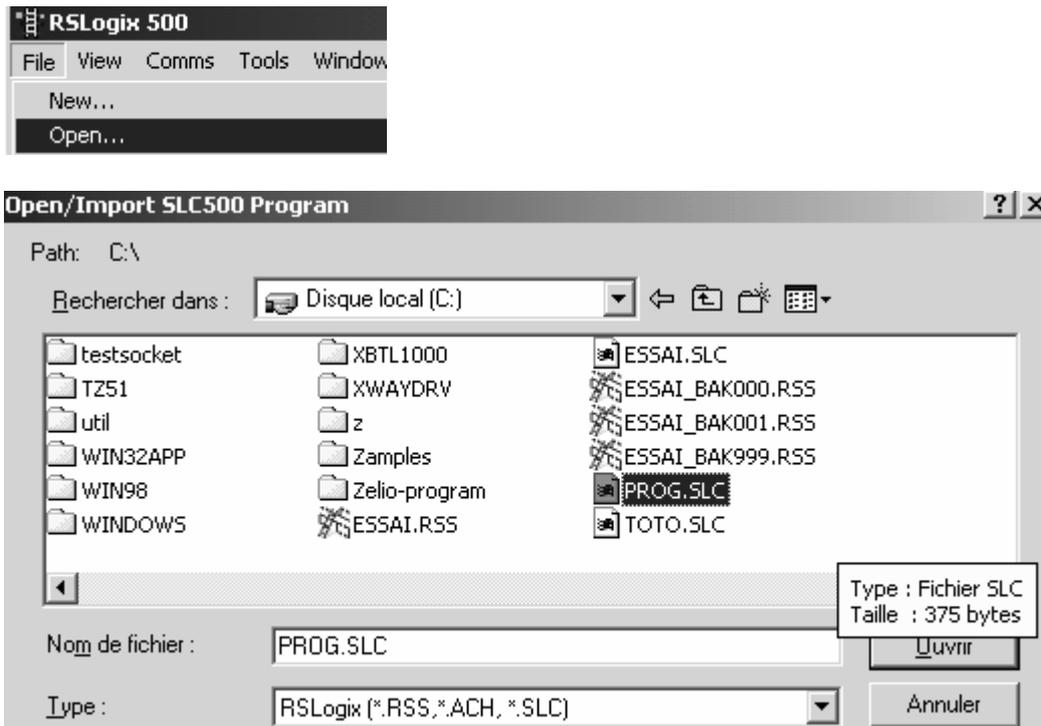
Transfert des programmes vers le logiciel RS-Logix 500 de ROCKWELL Software

Dans l'élément ci-dessous, désignez un fichier qui servira d'échange entre AUTOMGEN et RSLogix 500. A la fin de la compilation dans AUTOMGEN, ce fichier sera généré.

Options de génération de code (attention, modifier avec précaution)	
..... Optimiser le code généré	Non
..... Ne pas générer le code d'évolution des étapes Grafcet	Non
..... Ne pas générer le code d'évolution des bits utilisateur	Non
..... Utiliser un seul bit automate pour chaque bit utilisateur AUTOMGEN	Non
..... Utiliser la base de temps 1ms pour les temporisations si possible	Non
..... Fichier à importer dans le logiciel RSLOGIX de ROCKWELL après compilation	c:\prog.SLC

Génération d'un fichier .SLC compatible avec RSLogix 500

Lancez RSLogix500, puis ouvrez le fichier .SLC généré par AUTOMGEN.



Transférez le programme vers l'automate avec le logiciel RSLogix 500. Après avoir déconnecté RSLogix 500 de l'automate, vous pouvez réaliser la mise au point en mode connecté à partir d'AUTOMGEN.

Post-processeur MITSUBISHI

Ce post-processeur permet de programmer les automates MITSUBISHI de la gamme FX. Le code généré par AUTOMGEN peut être directement envoyé dans les automates MITSUBISHI de la gamme FX ou importer dans les logiciels MITSUBISHI FX-WIN ou GX-DEVELOPPER.

Choix du type d'automate

L'élément « Configuration / Post-processeur / MITSUBISHI / Système / Configuration matérielle » du navigateur permet de choisir le type de l'automate.

Éléments	Valeurs
[-] Configuration matérielle	
[-] Type de l'automate	FX0/FX0S
Options de génération de code	

Module de communication

Paramétrage du module de communication

Transfert des programmes vers le logiciel FX-WIN de MITSUBISHI

Dans l'élément ci-dessous, choisissez FXWIN.

Éléments	Valeurs	Commentaires
[-] Configuration matérielle		
[-] Type de l'automate	FX0/FX0S	
Options de génération de code (attention, modifier avec précaution)		
[-] Optimiser le code généré	Oui	
[-] Ne pas générer le code d'évolution des étapes Grafcet	Non	
[-] Ne pas générer le code d'évolution des bits utilisateur	Non	
[-] Utiliser un seul bit automate pour chaque bit utilisateur	Non	
AUTOMGEN		
[-] Option de génération de code	FXWIN	Génère un source pour FX WIN



La version Française de FXWIN doit impérativement être utilisée. L'importation a été validée avec la version 3.20 de FXWIN.

Après compilation dans AUTOMGEN, créez un projet dans FX-WIN. Ouvrez le programme en édition en mode liste d'instructions et choisissez « Insertion » dans le menu « Edition ».

Transfert des programmes vers le logiciel GX-DEVELOPPER de MITSUBISHI

Dans l'élément ci-dessous, choisissez GXDEVELOPPER.

+ Configuration matérielle		
- Options de génération de code (attention, modifier avec précaution)		
..... Optimiser le code généré	Oui	
..... Ne pas générer le code d'évolution des étapes Grafcet	Non	
..... Ne pas générer le code d'évolution des bits utilisateur	Non	
..... Utiliser un seul bit automate pour chaque bit utilisateur AUTOMGEN	Non	
..... Option de génération de code	GXDEVELOPPER	Génère un source pour GX DEVELOPPER
+ Déclaration de variables		

Lancez l'exécutable « A7TOGX.EXE » qui se trouve dans le répertoire d'installation d'AUTOMGEN. Après lancement, une icône « A7 → GX » apparaît dans la barre des icônes de WINDOWS.

Après compilation dans AUTOMGEN, créez un projet dans GX-DEVELOPPER. Ouvrez le programme en édition en mode liste d'instructions. Pressez simultanément les deux touches SHIFT du clavier pendant une seconde. Le programme est alors transféré dans GX-DEVELOPPER.

Pour désinstaller l'utilitaire « A7TOGX » cliquez sur l'icône avec le bouton droit de la souris. Vous pouvez laisser « A7TOGX » installé autant que nécessaire, nul n'est besoin de l'installer et de le désinstaller à chaque fois que vous désirez importer une application dans GX-DEVELOPPER.

Post-processeur TWIDO

Ce post-processeur permet de programmer les automates TWIDO de SCHNEIDER.

Choix de la configuration de l'automate

L'élément « Configuration / Post-processeur / TWIDO / Système / Configuration matérielle » du navigateur permet de choisir le type de l'automate.

Eléments	Valeurs
[-] Configuration matérielle	
Fichier contenant la configuration matérielle de l'automate	<AUTOM7DIR>\TWIDO\TWDLCAA10DRF.TWD

Le fichier « .TWD » est un fichier de configuration qui doit être généré avec l'atelier logiciel TWIDOSOFT de SCHNEIDER. Le sous-répertoire « TWIDO » du répertoire d'installation d'AUTOMGEN contient les fichiers de configuration pour plusieurs types d'automates TWIDO.

Module de communication

Communication TWIDO V7.000

Port de communication	
Port	Vitesse
COM1	4800
COM2	9600
COM3	14400
COM4	19200

Post-processeur ZELIO 2

Ce post-processeur permet de programmer les modules SECHNEIDER ZELIO 2.

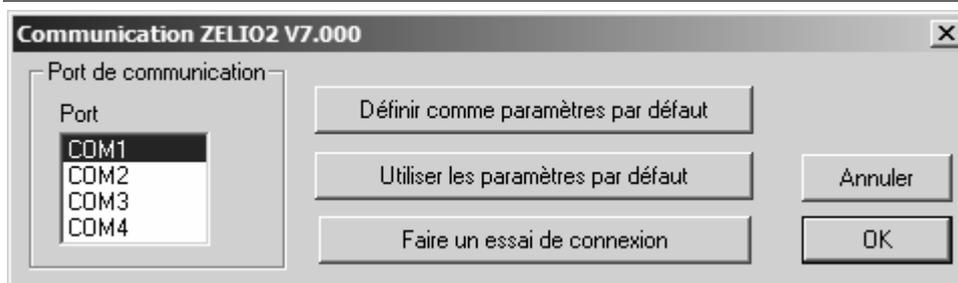
Initialisation de l'automate

Avant de pouvoir être utilisé avec AUTOMGEN, le firmware LD doit être téléchargé dans l'automate ZELIO 2 avec le logiciel ZELIOSOFT de SCHNEIDER.

Configuration de l'automate

La configuration de l'automate doit être réalisé avec le logiciel ZELIOSOFT de SCHNEIDER et téléchargée dans l'automate avec ce même logiciel avant de télécharger l'application avec AUTOMGEN.

Module de communication



Paramétrage du module de communication