



Dossier Ressource

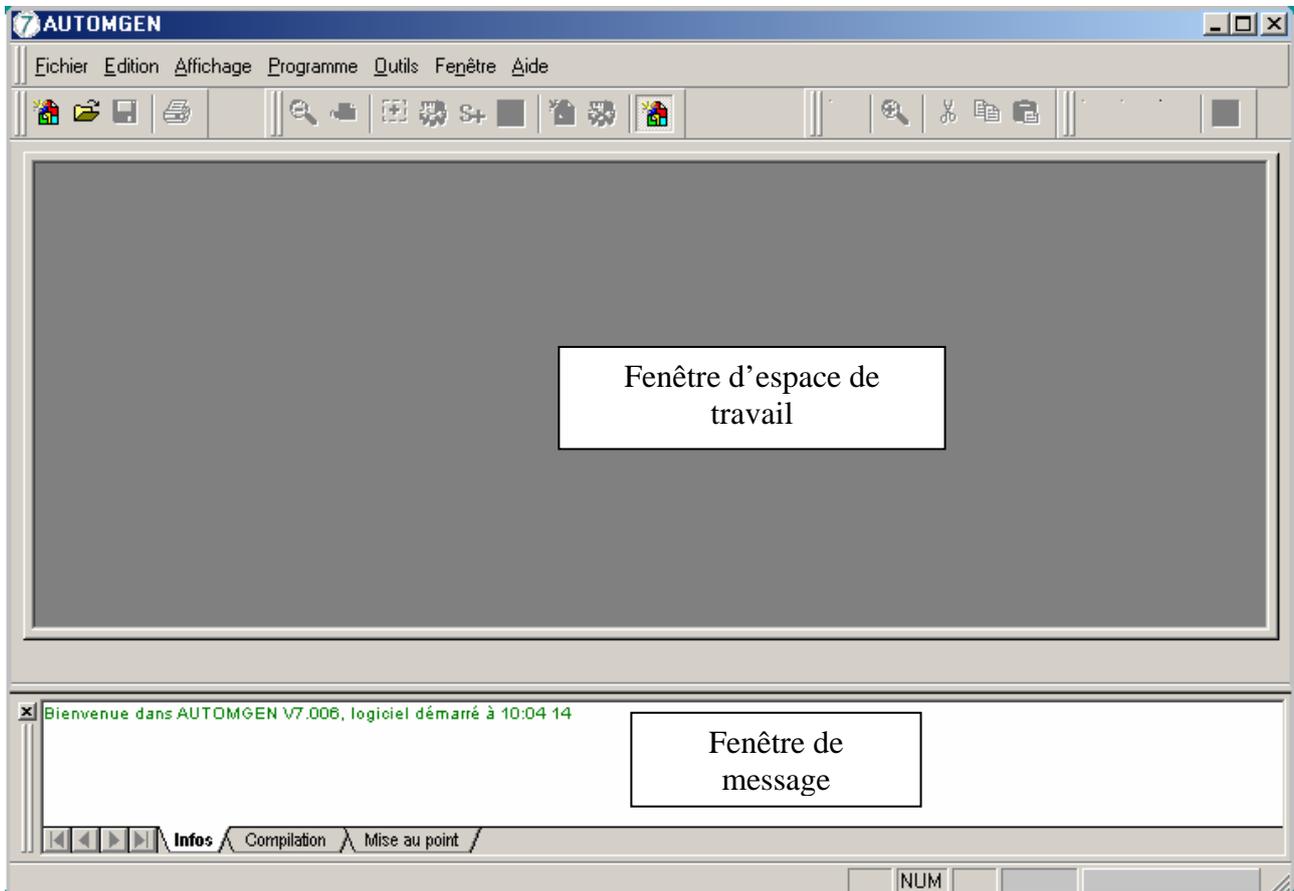
AUTOMGEN 7

CONTENU

- | | |
|--|--------------------------|
| 1 - LANCEMENT D'AUTOMGEN | Ressource page 2 sur 28 |
| 2 - OUVRIR UN NOUVEAU PROJET | Ressource page 2 sur 28 |
| 3 - LE NAVIGATEUR | Ressource page 5 sur 28 |
| 4 - LES VARIABLES | Ressource page 7 sur 28 |
| 5 - LES SYMBOLES | Ressource page 8 sur 28 |
| 6 - LES CONSTANTES | Ressource page 9 sur 28 |
| 7 - LES ACTIONS | Ressource page 9 sur 28 |
| 8 - LES TESTS | Ressource page 14 sur 28 |
| 9 - FOLIOS | Ressource page 15 sur 28 |
| 10 - CONFIGURATION | Ressource page 20 sur 28 |
| 11 - EXÉCUTER UNE APPLICATION | Ressource page 27 sur 28 |
| 12 - PROGRAMMATION DU RUN/STOP ET DU CdG | Ressource page 28 sur 28 |

1 - LANCEMENT D'AUTOMGEN

Pour démarrer AUTOMGEN, cliquez deux fois sur l'icône .
La fenêtre principale apparaît :



Cet écran contient 2 fenêtres :

- la fenêtre "**Espace de travail**",
- la fenêtre "**Messages**",

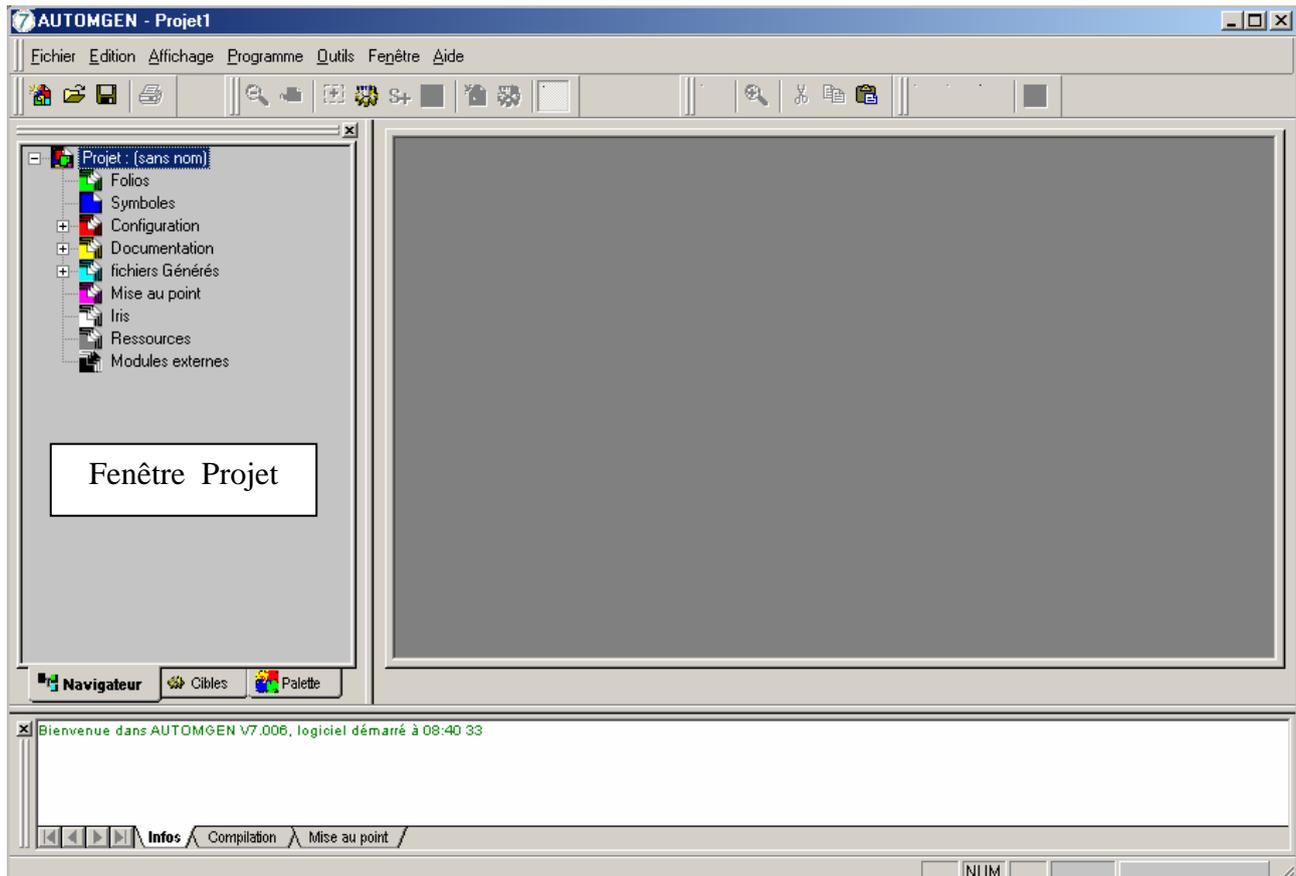
Pour accéder à une de ces 2 fenêtres, cliquez directement dans la fenêtre désirée.

2 - OUVRIR UN NOUVEAU PROJET

Pour ouvrir un nouveau projet :

- Sélectionnez la commande "**Nouveau**" du menu "**Fichier**"
- Ou appuyez sur l'icône 

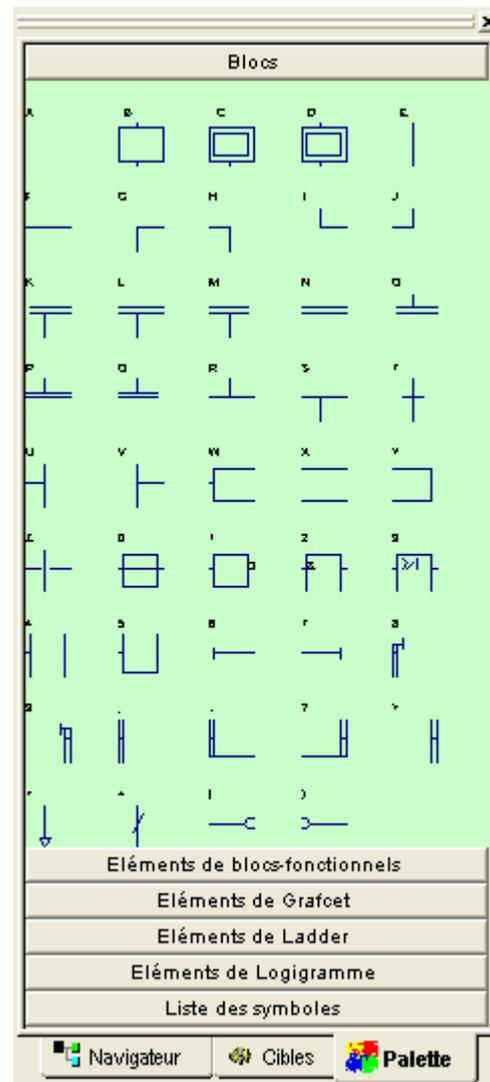
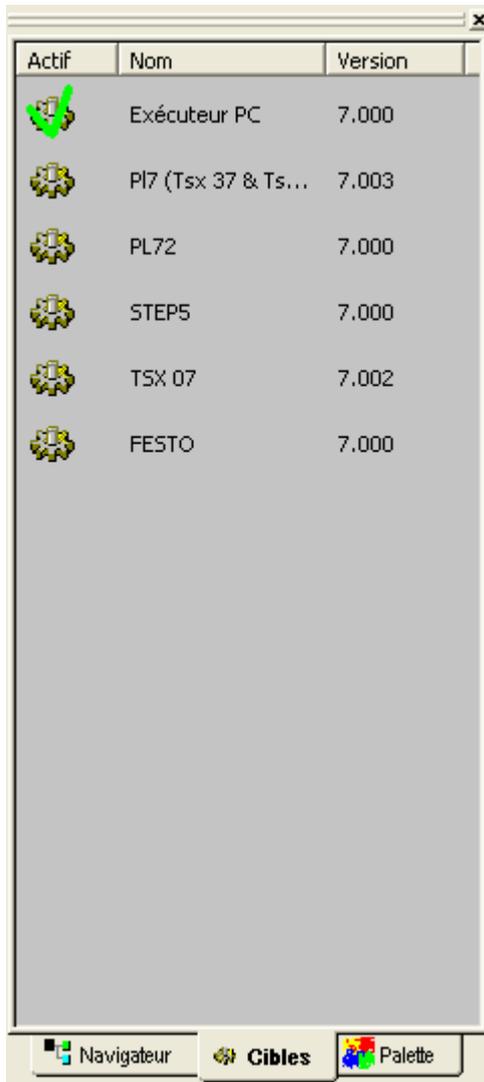
La fenêtre principale apparaît avec une troisième fenêtre « **Projet** » :



La fenêtre projet contient :

- L'onglet **Navigateur**, qui permet de créer les différents éléments d'une application et d'y accéder rapidement :
 - ✓ Folios,
 - ✓ Table de symbole,
 - ✓ Configuration du Post-processeur (adressage, ..),
 - ✓ Configuration matérielle,
 - ✓ Monitoring (Mise au point),
 - ✓ Modules IRIS 2D et 3D,
 - ✓ Ressources (objets IRIS 2D, objets 3D,),
 - ✓ Modules externes.

- L'onglet **Cibles** permet de sélectionner le type de traducteur (exécuteur PC ou automate programmable) avec laquelle on va effectuer la compilation du projet :
- L'onglet **Palette** permet d'accéder à des éléments de logigrammes, de blocs fonctionnels, de GRAFCET, de ladder, de bloc et la liste des symboles à utiliser lors de la création des programmes dans les folios :



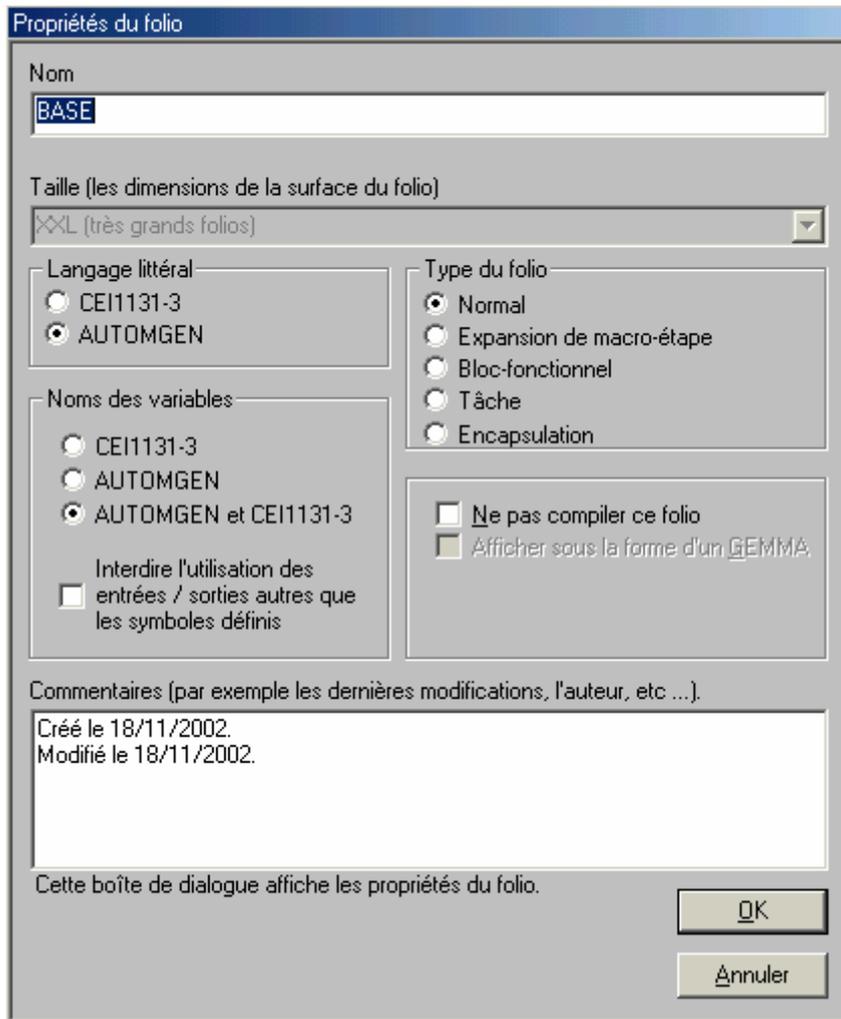
3 - LE NAVIGATEUR

FOLIOS

Un folio est une page sur laquelle est dessiné un programme ou une partie de programme.

Il faut cliquer avec le bouton droit de la souris pour créer un nouveau folio ou importer un ou plusieurs folios existants. Il est possible d'importer des folios au format AUTOMGEN (*.GR7) ou des GRAFCET (GRAFCETS rebouclés, sans renvoi) issus du logiciel CADEPA (*.GIG, format d'export de l'éditeur GRAFCET CADEPA Windows).

Lorsque le folio est ouvert, il est alors possible de définir ses propriétés : pour cela, sélectionner le folio puis cliquer sur le bouton droit de la souris et choisir dans le menu la commande **<propriétés>** pour faire apparaître la fenêtre suivante :



Les différents types de folios proposés sont les suivants :

- folio normal : pour la création de GRAFCET, de ladder, etc...
- folio contenant une expansion de macro-étape
- folio type bloc-fonctionnel
- folio contenant une tâche (par exemple tâche événementiel pour les automates télémechanique)
- folio encapsulation contenant les grafkets encapsulés

Cette fenêtre permet également de choisir le type de syntaxe AUTOMGEN ou CEI 1131-3 pour le langage littéral et les noms de variables de l'application.

Attention : le langage littéral est par défaut celui de la norme CEI1131-3 alors que le plus facile d'accès et celui de AUTOMGEN

SYMBOLES

La liste des symboles donne la correspondance entre des noms « symboliques » et des noms de variables. Un projet ne peut contenir qu'une seule table de symboles. Il faut cliquer avec le bouton droit de la souris pour avoir la possibilité de créer une table de symbole.

CONFIGURATION

POST-PROCESSEURS

Sous cette rubrique se trouve tous les éléments de configuration des post-processeurs, c'est à dire les différents types d'automates.

OPTIONS DU COMPILATEUR

Il permet de modifier le réglage des options du compilateur.

DOCUMENTATION

Il permet d'accéder à la fonction d'impression du dossier.

FICHIERS GÉNÉRÉS

CODE PIVOT

C'est un langage littéral de bas niveau résultant de la compilation du projet.

RÉFÉRENCES CROISÉES

Il affiche la liste des variables utilisées dans l'application avec leurs éventuelles variables automates associées ainsi que le nom du ou des folios où elles sont utilisées.

POST-PROCESSEURS

Il contient la compilation du projet dans le langage correspondant à l'automate sélectionné.

MISE AU POINT

Il regroupe des outils permettant la visualisation et la modification en dynamique de l'état des variables.

IRIS

IRIS 2D permet de créer des pupitres, des applications de supervision et des applications de simulation de parties opératives 2D.

IRIS 3D permet de créer des applications de simulations de parties opératives 3D.

RESSOURCES

Il permet d'ajouter tout type de fichier au projet.

Les fichiers ainsi ajoutés feront partie intégrante du projet et seront sauvegardés avec les autres éléments.

MODULES EXTERNES

Ils sont réservés à des modules exécutables développés par des tiers et interfacés avec automgen.

4 - LES VARIABLES

Automgen ne différencie pas les minuscules des majuscules.

Type de variable		Syntaxe		Plage valeur de n	Commentaire
		A7	CEI		
B	Entrées	In	%In	0 à 9999	Correspond ou non à des entrées physiques (dépend de la cible)
B	Sorties	On	%Qn	0 à 9999	Correspond ou non à des sorties physiques (dépend de la cible)
B	Bits Système	Un, Bn	%Mn	0 à 99	Bits réservés (exécuteur PC, automate...)
B	Bits Utilisateur	Un, Bn	%Mn	100 à 9999	Bits internes à usage général
B	Bits d'Étapes	Xn	%Xn	0 à 9999	Bits d'étapes du Grafcet
B	Bits de mots	Mn#m	%MWn:m	n : 0 à 9999 m : 0 à 15	Le numéro du bit est exprimé en décimal 0 est le bit de poids faible (LSB : Less Significant Bit), 15 est le bit de poids fort (MSB : More Significant Bit)
N	Mots Système	Mn	%MWn	0 à 199	Mots réservés (exécuteur PC, automate...)
N	Mots Utilisateur	Mn	%MWn	200 à 9999	Mots à usage général (entiers de 16 bits, caractères...)
N	Longs	Ln	%MDn	100 à 4998	Valeurs entières sur 32 bits
N	Flottants	Fn	%MFn	100 à	Valeurs réelles sur 32 bits (format IEEE)

				4998	
N	Compteur	Cn	%Cn	0 à 9999	Compteur 16 bits (de 0 à 65 535)
S	Temporisation	Tn	%Tn	0 à 9999	Temporisation 32 bits (de 0 à 4 294 967 295ms)

B : booléen N : Numérique S : Structuré (combinaison du type booléen et du type numérique)

Type de variable	Taille		Plage	Valeurs
Entier court	16 bits	Non signé	0.....2 ¹⁶ -1	0.....65 535
		Signé	-2 ¹⁶⁻¹2 ¹⁶⁻¹ -1	-32 768.....0.....+32 767
Entier long	32 bits	Non signé	0.....2 ³² -1	0.....4 294 967 295
		Signé	-2 ³²⁻¹2 ³²⁻¹ -1	-2 147 483 648.....0.....+2 147 483 647
Flottant (réel court)	32 bits		Plage en valeur absolue : 3,4.10 ⁻³⁸3,4.10 ³⁸	

Nombre signé (entier négatif) : $-n = \bar{n} + 1$ (complément à 2 de n)

(\bar{n} : complément à 1 de n)

Le signe est dans le bit de poids fort : 0 → + et 1 → -

Format IEEE des flottants (réels courts) 32 bits																
N° des bits	31	30														0
Signification	signe	exposant					D1	mantisse								
Taille	1 bit	8 bits					D23	23 bits								
Valeur	Offset = 127 (7FH)					Nombre flottant = signe . 2 ^{exposant-offset} . (1 + $\sum(D_k/2^k)$)										

5 - LES SYMBOLES

Un mnémonique (ou symbole) peut être associé à une variable afin d'améliorer la lisibilité du programme. Ces mnémoniques peuvent être utilisés en langage graphique (grafcet, ladder, logigramme) comme en langage littéral (bas niveau, étendu, structuré).

Automgen ne différencie pas les minuscules des majuscules

Premier(s) caractère(s) du symbole	Syntaxe	Commentaire	Exemples
Lettre (majuscule ou minuscule)	<code>_mmemo_</code> ou <code>mnemo</code>	Les underscores <code>_</code> sont optionnels	<code>vitesse</code> , <code>compteur...</code>
Variable réservée (Automgen)	<code>_VRmnemo_</code>	Les underscores <code>_</code> sont obligatoires	<code>_t1d_</code> , <code>_C03_</code>
Chiffre	<code>_5mnemo_</code>	Les underscores <code>_</code> sont obligatoires	<code>_1EVA_...</code>
Opérateur (+, - ...)	<code>+_mnemo_</code>	Les underscores <code>_</code> sont obligatoires	<code>+_V_</code> , <code>_-V_...</code>

Les espaces et les underscores `_` sont interdits dans la composition d'un symbole (`vanne_B`)

Éviter les accents, cédilles... car lors d'un copier-coller au format EMF d'un grafcet ou autre vers un traitement de texte (Word, par exemple), ces caractères sont susceptibles d'être modifiés.

Les underscores n'apparaissent pas à l'écran comme à l'impression ! Ils n'apparaissent qu'en mode édition.

6 - LES CONSTANTES

Type de constante	Taille	Syntaxe					
		En décimal	NDM	En hexadécimal	NDM	En binaire	NDM
Entière courte	16 bits	1000	5	\$3E8	4	%1111101000	16
				16#3E8		2#1111101000	
Entière longue	32 bits	100000L	10	\$186A0L	8	%110000011010100000L	32
				16#186A0L		2#110000011010100000L	
Réelle	32 bits	3.14R -2.83E- 5R					

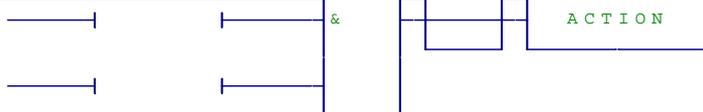
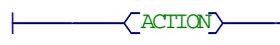
NDM : nombre de digits maxi

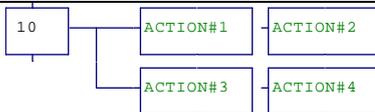
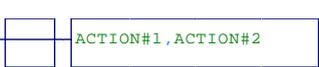
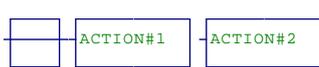
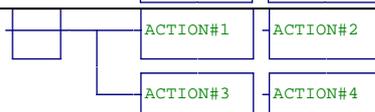
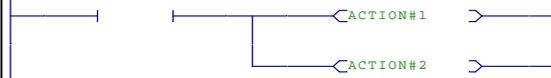
Type de constante	Taille	Syntaxe	NDC
Chaîne de caractères ASCII	courte	16 bits	'A' 1
		32 bits	"AS" 2
	longue	32 bits	'AS' 2
		32 bits	"TOTO" 4

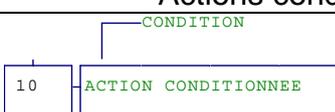
NDC : nombre de caractères maxi par mot / long

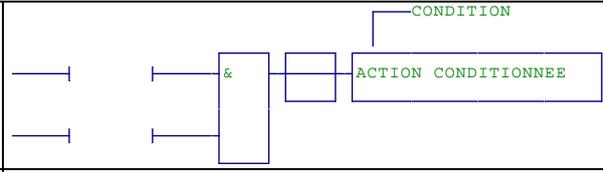
7 - LES ACTIONS

Les actions sont utilisées dans les rectangles d'action des langages grafcet et logigramme et dans les bobines du langage ladder.

Langage	Commande de l'élément	Elément	Nom de l'élément
GRAF CET			Rectangle d'action
Logigramme			Rectangle d'action
Ladder			Bobine

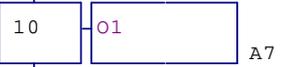
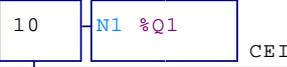
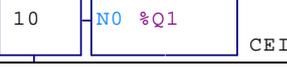
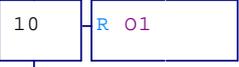
Langage	Actions multiples		
GRAF CET			
Logigramme			
Ladder			

Langage	Actions conditionnées
GRAF CET	

Logigramme	
Ladder	Pas de graphisme particulier

Les actions sur une variable booléenne

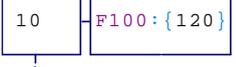
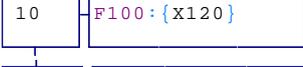
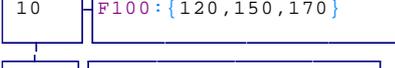
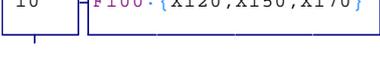
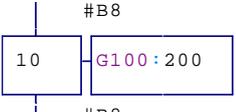
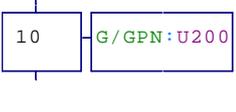
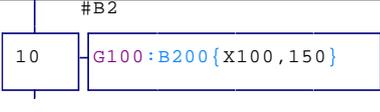
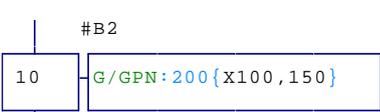
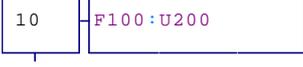
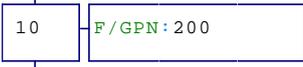
État de la Variable Booléenne VB (In, On, Un) en fonction de l'état de la commande CDE

Action	CDE	VB	Exemple (langage Grafset)	Commentaire
Affectation VB	0	0		Si l'étape 10 est active alors O1 prend la valeur 1, sinon O1 prend la valeur 0
	1	1		
Affectation complémentée	0	1		Si l'étape 10 est active alors O1 prend la valeur 0, sinon O1 prend la valeur 1
	1	0		
Mise à un	0	inchangé		Si l'étape 10 est active alors O1 prend la valeur 1, sinon O1 reste dans le même état
	1	1		
Mise à zéro	0	inchangé		Si l'étape 10 est active alors O1 prend la valeur 0, sinon O1 reste dans le même état
	1	0		
Inversion	0	inchangé		Si l'étape 10 est active alors l'état de O1 est inversé, sinon O1 reste dans le même état
	1	inverse		
Impulsion sur front montant	0, 1	0		A l'activation de l'étape 10 O1 passe fugitivement à l'état 1
	^1	impulsion		
Impulsion sur front descendant	^0	impulsion		A la désactivation de l'étape 10 O1 passe fugitivement à l'état 1
	0, 1	0		

Automgen ne différencie pas les minuscules des majuscules.

Les actions sur le GRAFCET

Pour désigner un GRAFCET il suffit de faire référence à une étape quelconque de ce GRAFCET
 Pour désigner le ou les GRAFCETs contenus dans un folio il suffit de faire référence au nom de ce folio

Action	Syntaxe	Exemple	Commentaire
Figeage	F<n° d'étape>		Le grafcet contenant l'étape n°100 (grafcet 100) est figé
	F/<nom du folio>		Les grafcets contenus dans le folio GPN sont figés
Forçage	F<n° d'étape>:{étapes actives}		Force l'étape 120 du grafcet 100 à 1 et les autres à 0
			
	F/<nom du folio>:{étapes actives}		Force les étapes 120, 150 et 170 du grafcet 100 à 1 (et les autres à 0)
			Force toutes les étapes des grafcets du folio GPN à 0
Mémorisation	G<n° d'étape>:<n° de bit>		Mémorise l'état des 8 étapes du grafcet 100 dans les bits U200 à U207
	G/<nom du folio>:<n° de bit>		Mémorise l'état des 8 étapes du folio GPN dans les bits U200 à U207
	G<n° d'étape>:<n° de bit>{étapes actives}		Mémorise l'état du grafcet 100 ou du folio GPN dans les bits U200 et U201 lorsque seules les étapes 100 et 150 sont actives (état particulier du ou des grafcets)
	G/<nom du folio>:<n° de bit>{étapes actives}		
Forçage sur mémorisation	F<n° d'étape>:<n° de bit>		Restauration du grafcet 100 ou du folio GPN suivant l'état mémorisé à partir du bit U200
	F/<nom du folio>:<n° de bit>		

Ces ordres supportent les instructions S, R, N et I

Les actions sur un compteur, un mot, un long

Valeur d'une Variable Numérique VN (Cn, Mn, Ln) en fonction de l'état de la commande CDE

Action	CDE	VN	Exemple (langage Grafset)	Commentaire
Mise à moins un	0	inchangé	10 S M200	Si l'étape 10 est active alors M200 prend la valeur 1, sinon M200 conserve sa valeur
	1	-1		
Mise à zéro	0	inchangé	10 R M200	Si l'étape 10 est active alors M200 prend la valeur 0, sinon M200 conserve sa valeur
	1	0		
Incrémentatation	0	inchangé	10 +C0	Si l'étape 10 est active alors le compteur C0 s'incrémente
	1	valeur+1		
Décrémentatation	0	inchangé	10 -L10	Si l'étape 10 est active alors L10 se décrémente
	1	valeur-1		
Incrémentatation sur front montant	0, 1	inchangé	10 P1 +C0	A l'activation de l'étape 10 le compteur C0 s'incrémente de 1
	^1	valeur+1		
Incrémentatation sur front descendant	^0	valeur+1	10 P0 +C0	A la désactivation de l'étape 10 le compteur C0 s'incrémente de 1
	0, 1	inchangé		

Actions et temporisations

Une temporisation est un type structuré composé de 2 variables booléennes (lancement et fin de tempo) et de 2 variables numériques 32 bits (consigne et compteur)

La durée maximale est de 4 294 967 295ms, soit 49d17h02m47s295ms

Unité	jour	heure	minute	seconde	dixième de seconde	milliseconde
Opérateur	d	h	m	s		ms

La durée s'exprime par défaut en dixième de seconde.

Syntaxe	GRAFSET	Commentaire
Temporisation(durée) dans l'action Variable associé dans le test		
Temporisation/variable de lancement/durée		
Durée/variable de lancement/temporisation		
Durée/variable de lancement		Le test sera à 1 5 secondes après le front montant de i1 (et son maintien à 1)

Type de temporisation	GRAFSET	Commentaire
Non mémorisée		t1 passe à 1 5 secondes après l'activation de l'étape 10 et son maintien
Mémorisée		t1 passe à 1 5 secondes après l'activation de l'étape 10 (maintenue active ou désactivée)
Mise à zéro ou initialisation		t1 est remis à 0 (réinitialisé)

Action et temporisation	GRAFSET	Commentaire
Non mémorisée et limitée dans le temps		La sortie O1 est à 1 pendant 5s maximum après l'activation de l'étape 10
Mémorisée et limitée dans le temps		La sortie O1 est à 1 pendant 5s après l'activation de l'étape 10 (maintenue active ou désactivée)
Non mémorisée et temporisée		La sortie O1 est à 1 5s après l'activation de l'étape 10 et son maintien activé
Mémorisée et temporisée		La sortie O1 est à 1 5s après l'activation de l'étape 10 (maintenue active ou désactivée)

7 - LES TESTS

Les tests sont utilisés dans les rectangles d'action, les conditions des actions conditionnées et les transitions du langage grafcet et dans les contacts des langages logigramme et ladder.

Langage	Test	Nom du test
GRAF CET		Condition Transition
Logigramme		Contact
Ladder		Contact

Un test est une équation booléenne composée de une ou de n variables séparées par des opérateurs « + »(ou) ou « . »(et).

Les différents tests sont :

- « / » : test l'état complémenté
- « u » ou « ↑ » : test le front montant
- « d » ou « ↓ » : test le front descendant
- « » (néant) ou « =1 » : test toujours vrai (cette option est modifiable : Pour obliger à mettre =1 dans une transition vraie :

dossier "Configuration"

double clic gauche sur dossier "Options du compilateur"

cocher la dernière case "refuser les transitions vides"

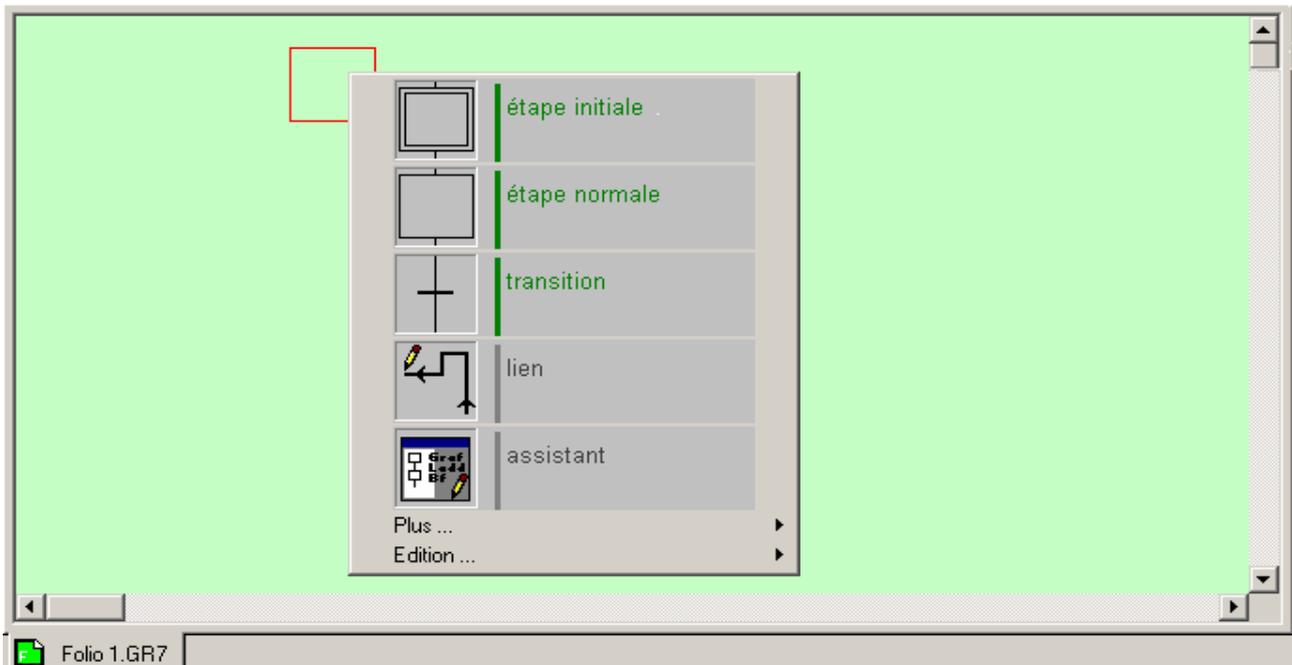
- « = » : égal
- « ! » ou « <> » : différent
- « < » : inférieur (non signé)
- « > » : supérieur (non signé)
- « << » : inférieur (signé)
- « >> » : supérieur (signé)
- « <= » : inférieur ou égal (non signé)
- « >= » : supérieur ou égal (non signé)
- « <<= » : inférieur ou égal (signé)
- « >>= » : supérieur ou égal (signé)

8 - FOLIOS

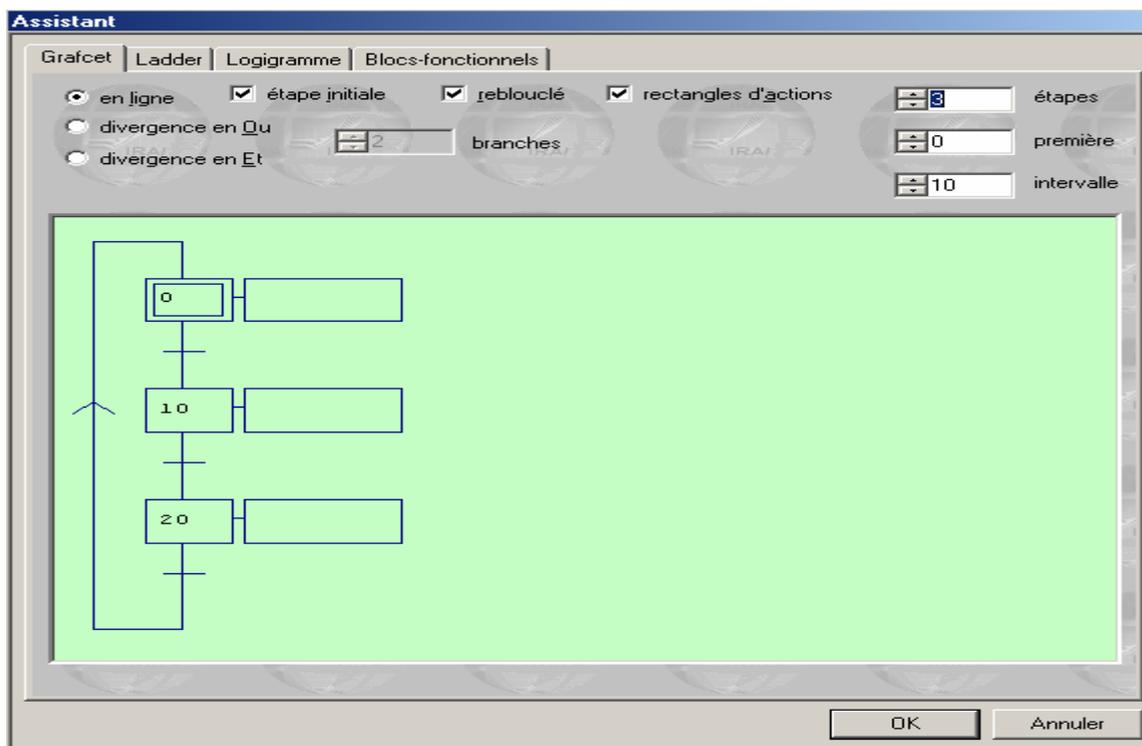
Une fois, qu'on a crée un nouveau folio, on peut le programmer.

CRÉATION DE GRAFCET

En cliquant sur le bouton droit de la souris dans la fenêtre du folio, on obtient une fenêtre de sélection d'objets graphiques :

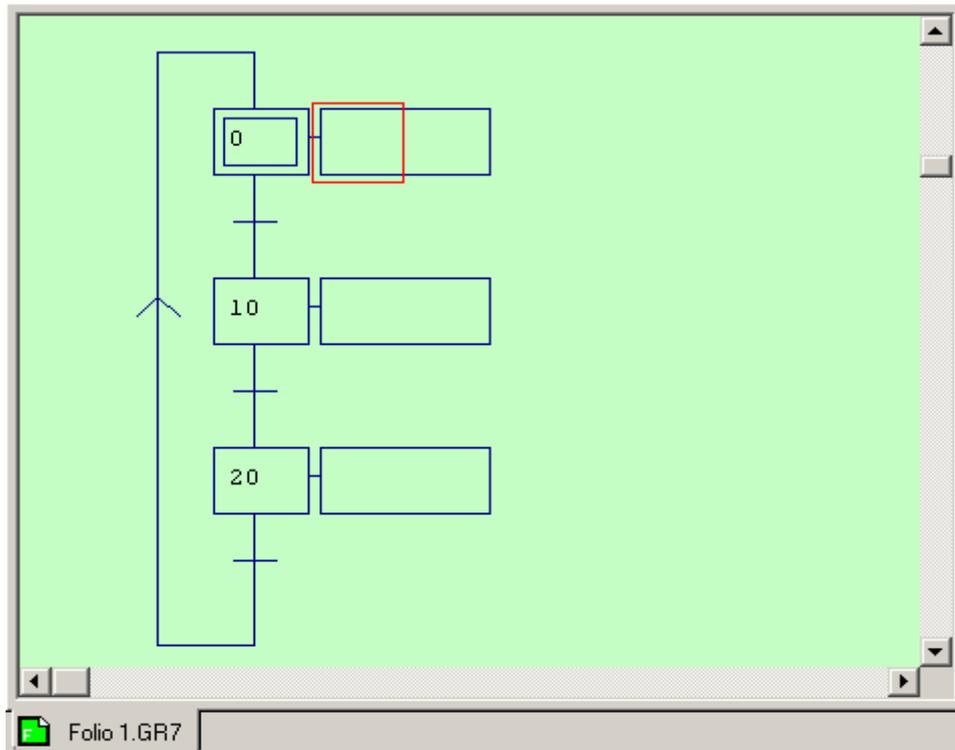


Si on sélectionne l'assistant, on obtient une aide à la création de grafcet, de ladder, de logigramme ou de blocs-fonctionnels :

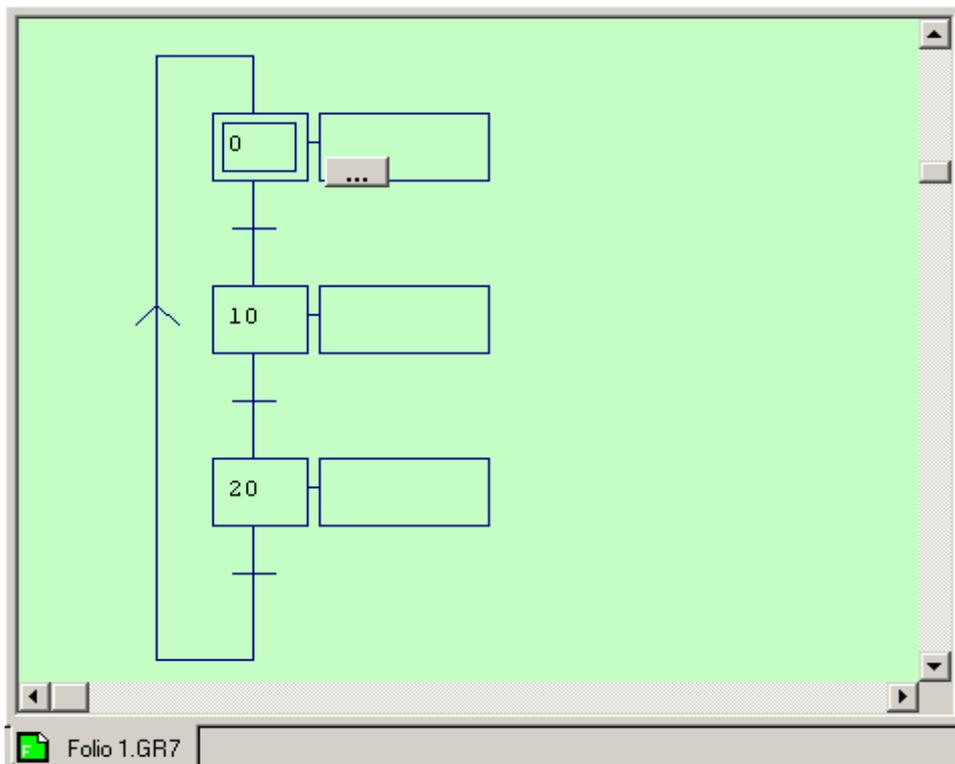


ÉTAPES

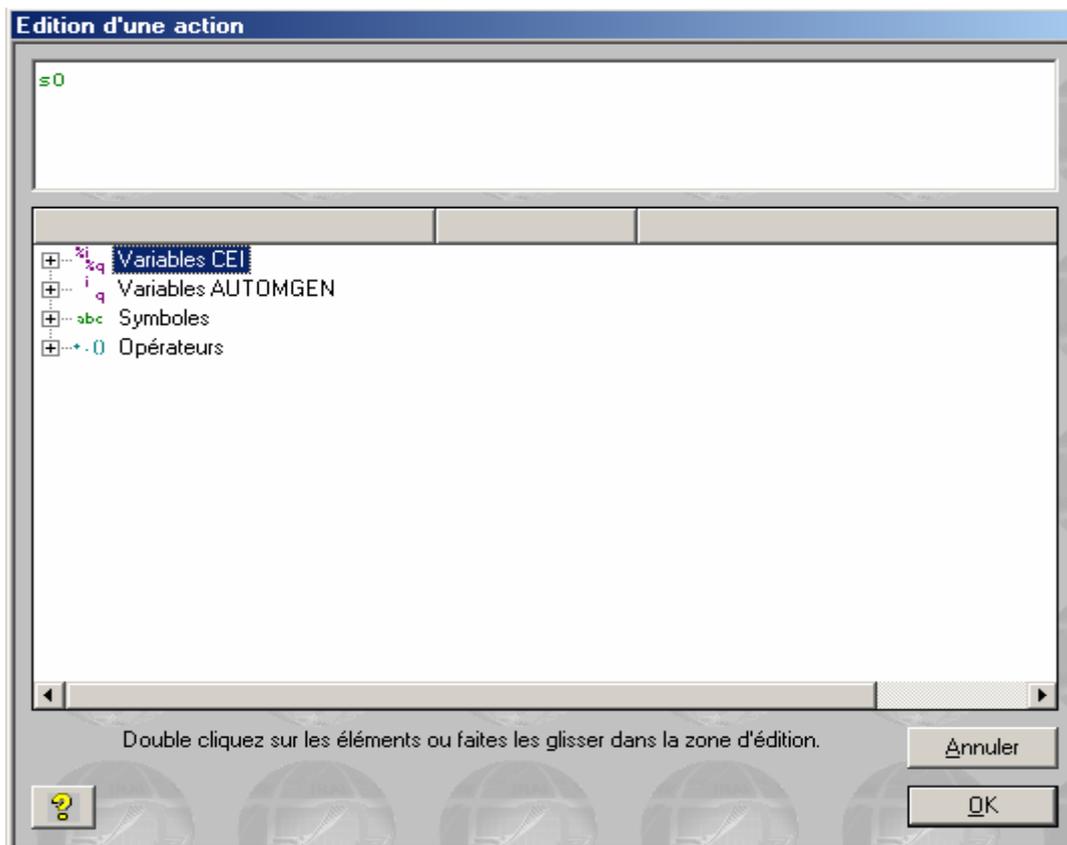
Pour insérer les actions à effectuer sur une étape, on vient cliquer sur le rectangle lié à l'étape.



Une fois qu'on a cliqué, on obtient :

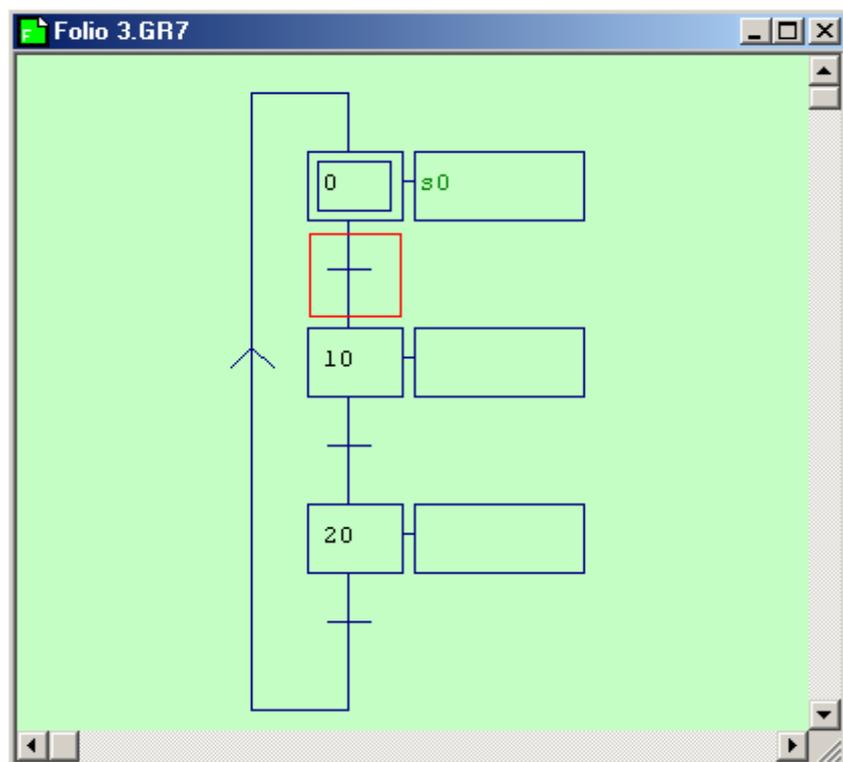


On peut écrire directement l'action et la variable ou en cliquant sur , on obtient une fenêtre contenant toutes les actions et les variables qu'on peut effectuer sur cette étape.

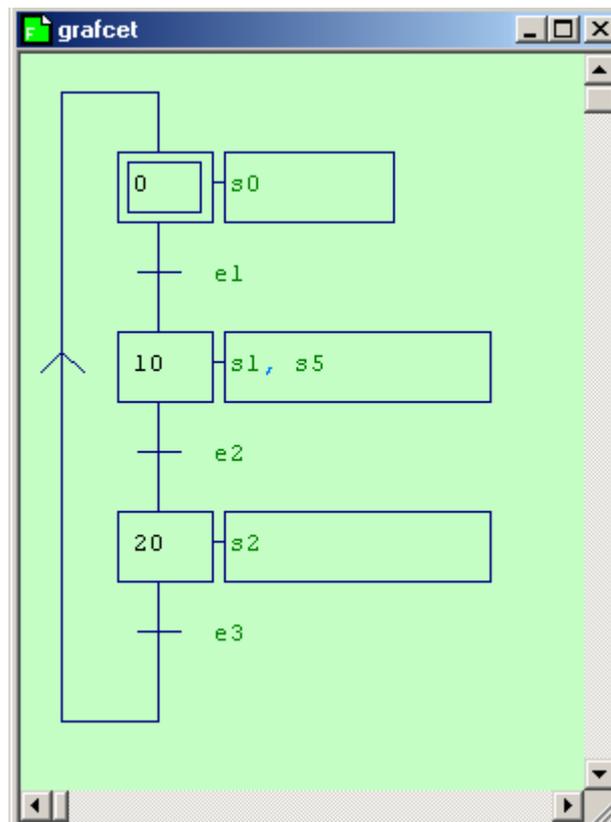


TRANSITIONS

On clique sur la transition qu'on veut renseigner :



Ensuite, on suit la même démarche que pour une étape.



Remarque :

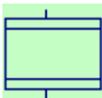
- L'onglet Palette permet de modifier le grafcet, en rajoutant d'autre étape ou des divergences et des convergences.
- Sur les étapes, on peut effectuer plusieurs actions en les séparant par « , ».
- Pour ajouter des commentaires, il ne faut pas se positionner sur le GRAFCET.

MACRO-ÉTAPES

Une macro-étape (ME) est l'unique représentation d'un ensemble unique d'étapes et de transitions nommé « expansion de ME ».

Règles d'une ME :

- Une macro-étape est un GRAFCET se trouvant dans un folio distinct.
- Il doit comporter une étape d'entrée et une étape de sortie.
- L'étape d'entrée devra porter le numéro 0 ou le repère Exxx, avec xxx = numéro quelconque.
- L'étape de sortie devra porter le numéro 9999 ou le repère Sxxx, avec xxx=numéro quelconque.

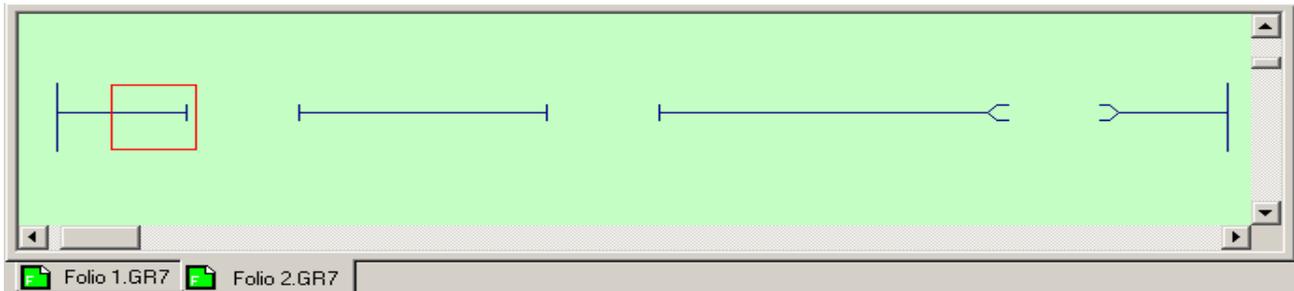


- Symbole :

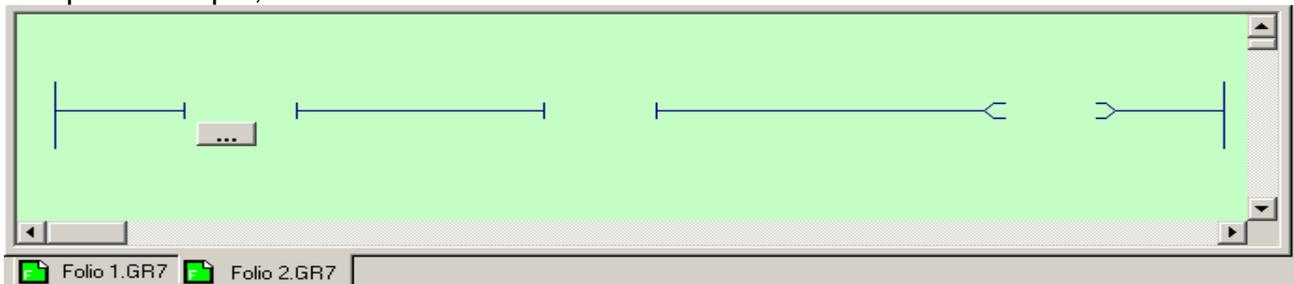
CRÉATION DE LADDER

De la même façon que le GRAFCET, on sélectionne le ladder dans l'assistant.

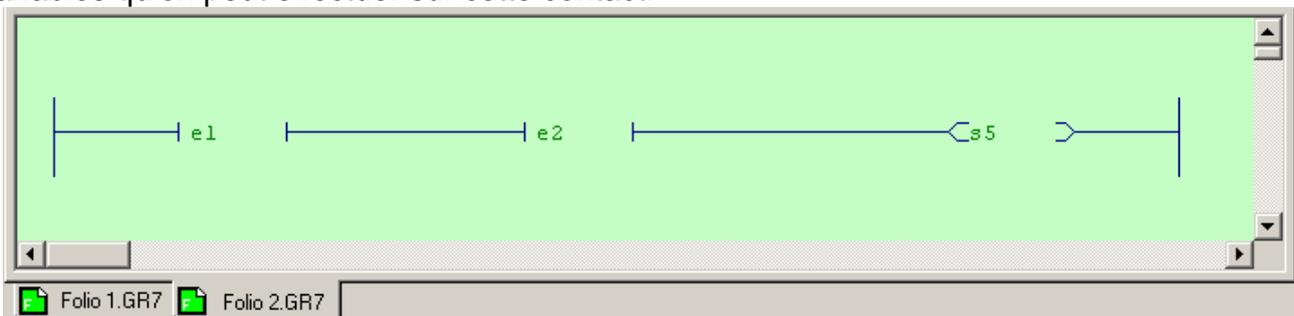
Pour renseigner le contact, on se positionne comme suit et on clique sur le bouton gauche de la souris :



Une fois qu'on a cliqué, on obtient :

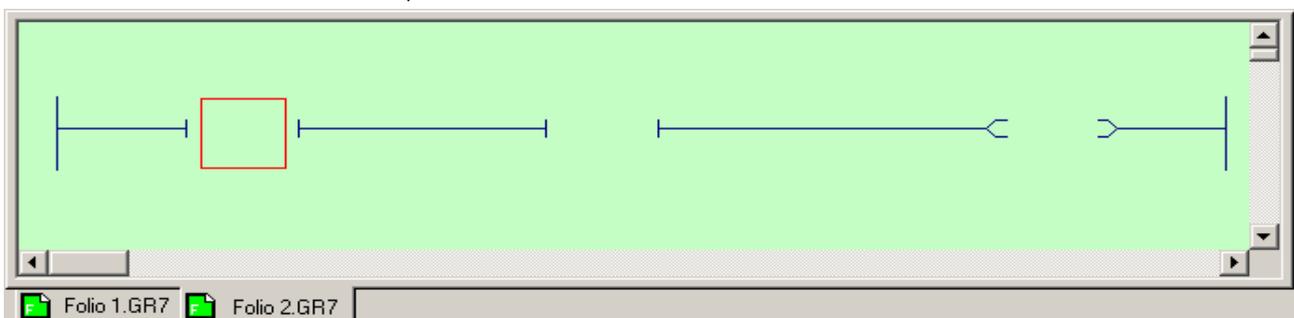


En cliquant sur les **...**, on obtient la même fenêtre que pour le grafcet contenant tous les tests et les variables qu'on peut effectuer sur cette contact.



Remarque :

- L'onglet Palette permet de modifier le ladder, en rajoutant d'autre contact ou bobines.
- Pour ajouter des commentaires, il ne faut pas se positionner sur le ladder. Par contre pour mettre le commentaire sur le contact, il faut se mettre comme suit :

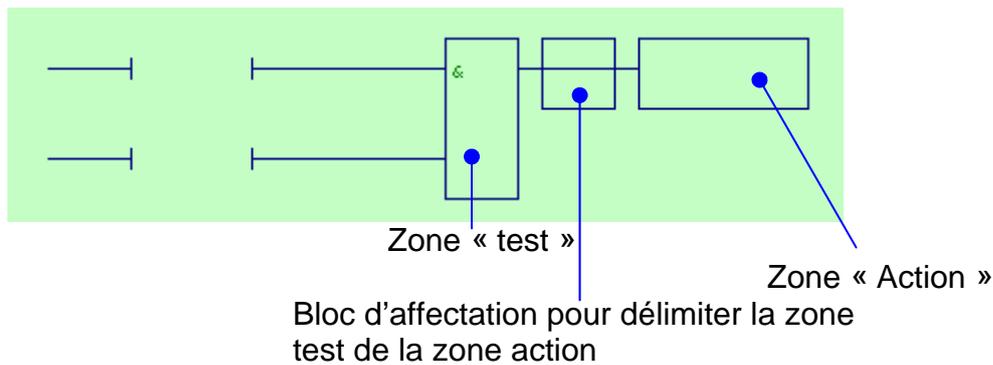


CRÉATION DES LOGIGRAMMES

De la même façon que le grafcet, on sélectionne le logigramme dans l'assistant.

Le langage logigramme permet d'écrire graphiquement des équations booléennes.

Le logigramme se présente de la forme :



Le paramétrage se fait de la même façon que le ladder et le GRAFCET.

9 - CONFIGURATION

POST-PROCESSEUR

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.

LES FICHIERS DE CONFIGURATION

Quatre éléments de configuration sont utilisés par chaque post-processeur.

SYSTÈME

Le 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.

CORRESPONDANCES DE VARIABLES

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.

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.

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.

OPTIONS DU COMPILATEUR

Cet élément permet de choisir les options de compilation de l'application.

EXEMPLE : EXECUTEUR PC

Si on choisit la cible « Exécuteur PC », on a alors un automate embarqué sur le PC. Il contient les éléments suivantes :

- Variables :

Il permet de configurer le nombre de variable pour l'application.

- Exécution :

Il permet de modifier la période d'exécution sur PC.

- Drivers E/S :

En ajoutant les drivers d'E/S, on peut piloter les entrées/sorties connecté directement à l'ordinateur.

EXEMPLE : POST-PROCESSEUR PL7 (TSX37 et TSX57)

Il contient les éléments suivants :

- Système
- Correspondances des variables
- Code constructeur démarrage
- Code constructeur fin
- Option de connexion
- Module de communication

On peut définir les quatre premiers éléments comme des éléments de configuration par défaut.

Remarque :

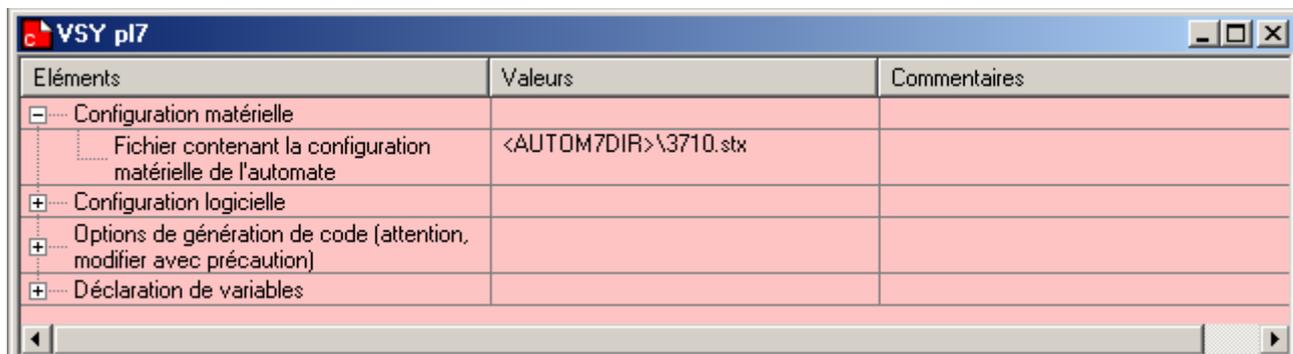
Le fait de définir un élément comme configuration par défaut, entraîne que seule la réinstallation du post-processeur permet de restaurer l'élément de configuration.

SYSTÈME

Le système est décomposé de 4 éléments :

- Configuration matérielle
- Configuration logicielle
- Options de génération de code
- Déclaration de variables

CONFIGURATION MATÉRIELLE



Eléments	Valeurs	Commentaires
[-] Configuration matérielle		
Fichier contenant la configuration matérielle de l'automate	<AUTOM7DIR>\3710.stx	
[+] Configuration logicielle		
Options de génération de code (attention, modifier avec précaution)		
[+] Déclaration de variables		

Pour changer de fichier, il faut cliquer 2 fois sur le fichier.

Remarque :

Le fichier de configuration doit être créé avec les outils de programmation SCHNEIDER. Et ensuite, on le sauvegarde dans le répertoire d'installation d'AUTOMGEN ou on l'intègre dans les ressources du projet d'AUTOMGEN.

CONFIGURATION LOGICIELLE

Eléments	Valeurs	Commentaires
+ Configuration matérielle		
- Configuration logicielle		
Générer directement le fichier binaire sans passer par le logiciel SCHNEIDER	YES	
Lancer automatiquement le logiciel SCHNEIDER (Seulement si BUILDBIN=NO)	NO	
Version du logiciel SCHNEIDER (seulement si BUILDBIN=NO et RUNPL7SOFT=YES)	TOPL7PRO.EXE	PL7 Pro
Fichier à importer dans PL7Micro ou PL7Junior ou PL7PRO après compilation (BUILDBIN=NO)		
+ Options de génération de code (attention, modifier avec précaution)		
+ Déclaration de variables		

- Générer directement le fichier binaire sans passer par le logiciel SCHNEIDER :
Si on choisit « non », on est dans le mode d'import manuel et on est obligé de choisir un nom de fichier dans l'élément « fichier à importer dans PL7Micro ou PL7Junior ou PL7PRO après compilation » pour l'export vers l'atelier logiciel SCHNEIDER.
Si on choisit « oui », on est dans le mode d'import automatique et on doit rentrer la version du logiciel utilisé dans l'élément « Version du logiciel SCHNEIDER »

OPTION DE GÉNÉRATION DE CODE

Eléments	Valeurs	Commentaires
+ Configuration matérielle		
+ Configuration logicielle		
- 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	
Utiliser la nouvelle entête pour les fichiers FEF (seulement pour PL7 PRO et TSX 57)	Non	
+ Déclaration de variables		

- 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 :
Si réglé sur « oui », on doit écrire dans l'élément « code constructeur fin » les instructions permettant la copie des états immédiats de variables booléennes vers les états passés.

L'état immédiat, c'est l'état écrit par la dernière instruction exécutée par le 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.

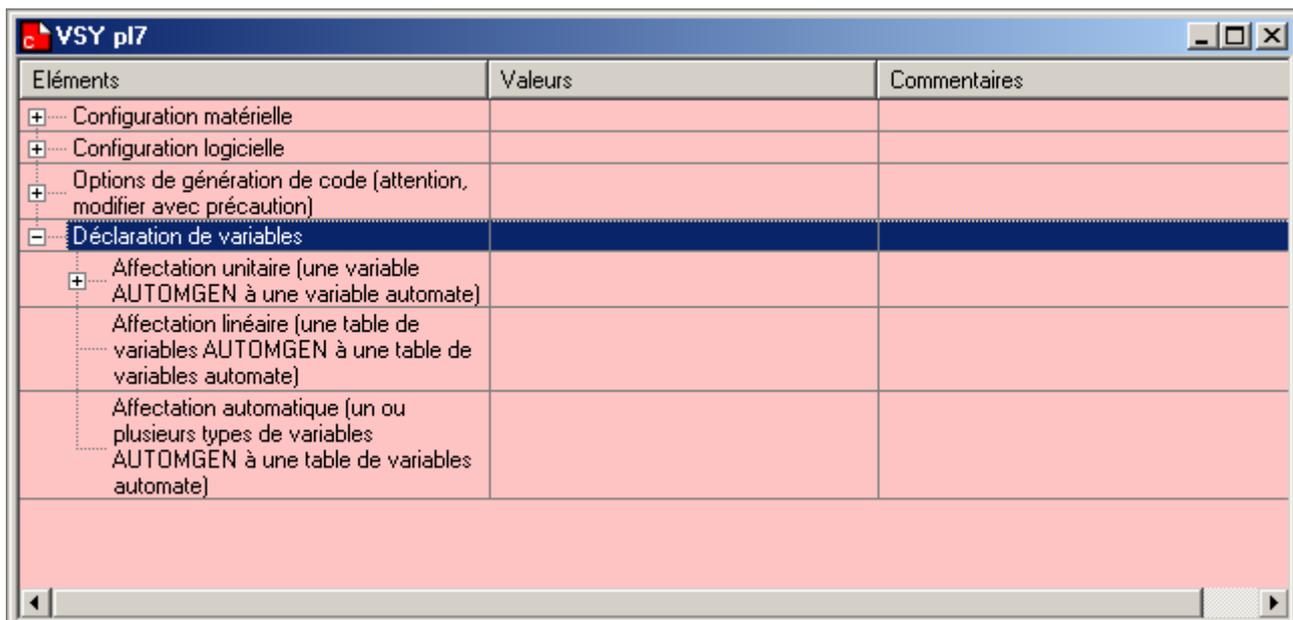
L'état passé de la variable, c'est l'état qu'avait la variable à la fin du dernier cycle d'exécution.

De ce fait, 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.
- Ne pas générer le code d'évolution des bits utilisateurs :

Identique à l'option précédente mais appliquée au bits utilisateurs.

DÉCLARATION DE VARIABLES SYSTÈME



- Affectation unitaire :

Elle permet d'associer une variable AUTOMGEN à une variable de la cible. 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.

- Affectation linéaire :

Elle permet d'associer une série de variables consécutives 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

Cette déclaration utilise 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 variable.

- Affectation automatique :

Elle 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 de la déclaration.

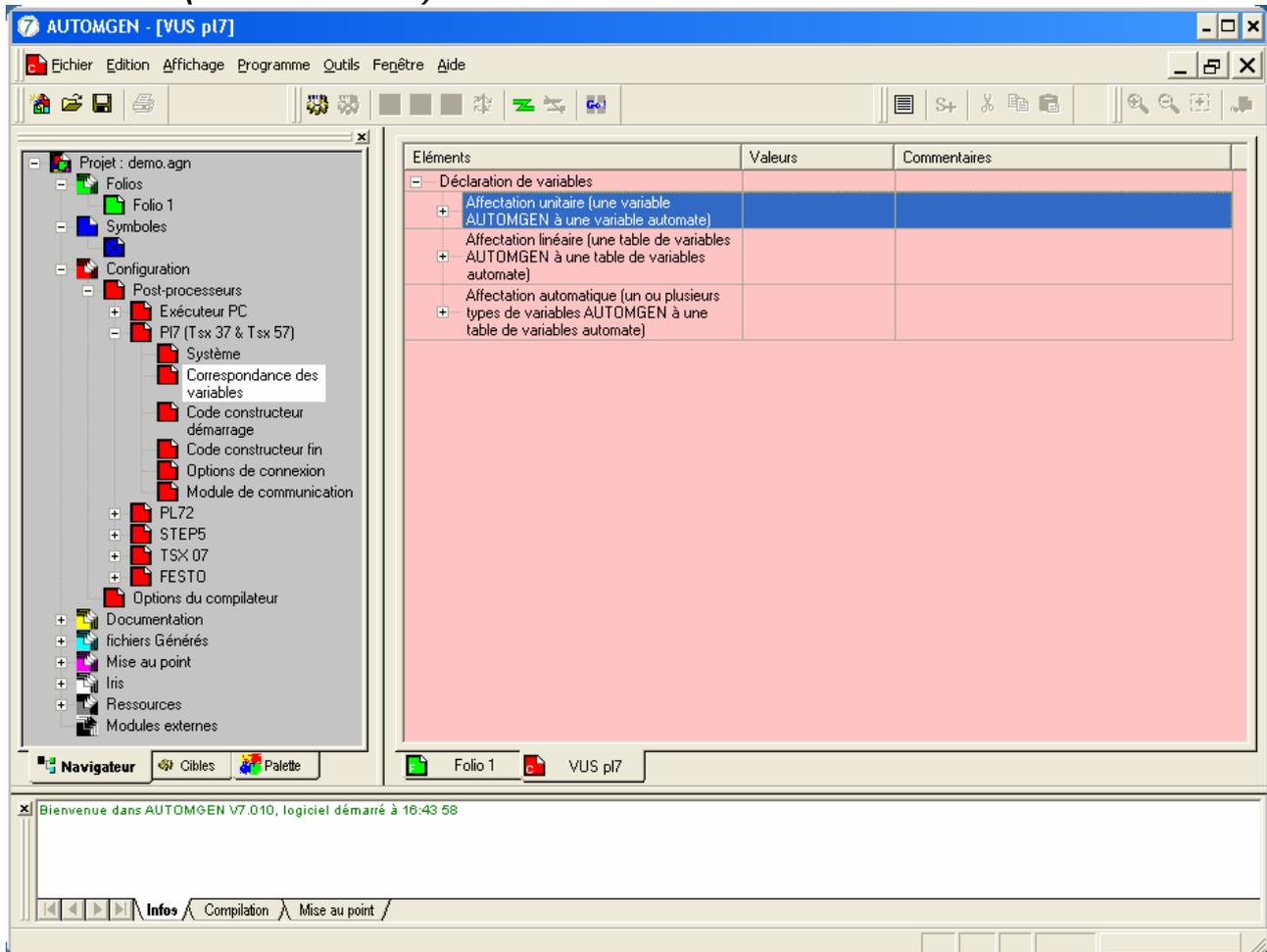
Cette déclaration utilise trois informations : le type de la variable AUTOMGEN, le nom de la première variable de la cible et le numéro de la dernière variable incluse de la plage de la cible.

CORRESPONDANCES DE VARIABLES

La correspondance de variables permet d'attribuer des variables AUTOMGEN à des variables de la cible.

La déclaration se fait de la même façon que dans l'élément « système ».

Sélectionner le document **<Correspondance des variables>** dans le dossier **<Configuration/Post-processeurs/PI7 (Tsx37 & Tsx 57)>**

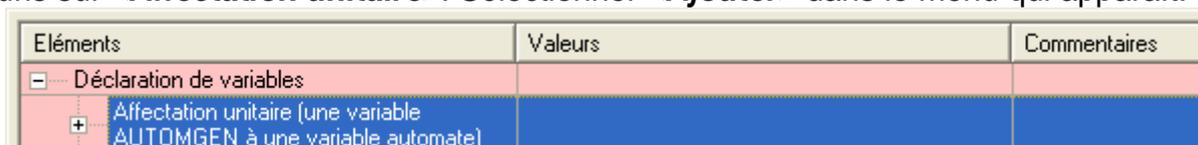


AUTOMGEN propose 2 types d'adressage :

- l'**Affectation unitaire** : une variable AUTOMGEN est affectée à une adresse de l'automate,
- l'**Affectation linéaire** : les variables AUTOMGEN sont affectées automatiquement et de façon linéaire aux adresses de l'automate.

l'Affectation unitaire :

Pour affecter unitairement les variables AUTOMGEN aux adresses API, cliquer avec le **bouton droit** de la souris sur **<Affectation unitaire>**. Sélectionner **<Ajouter>** dans le menu qui apparaît.

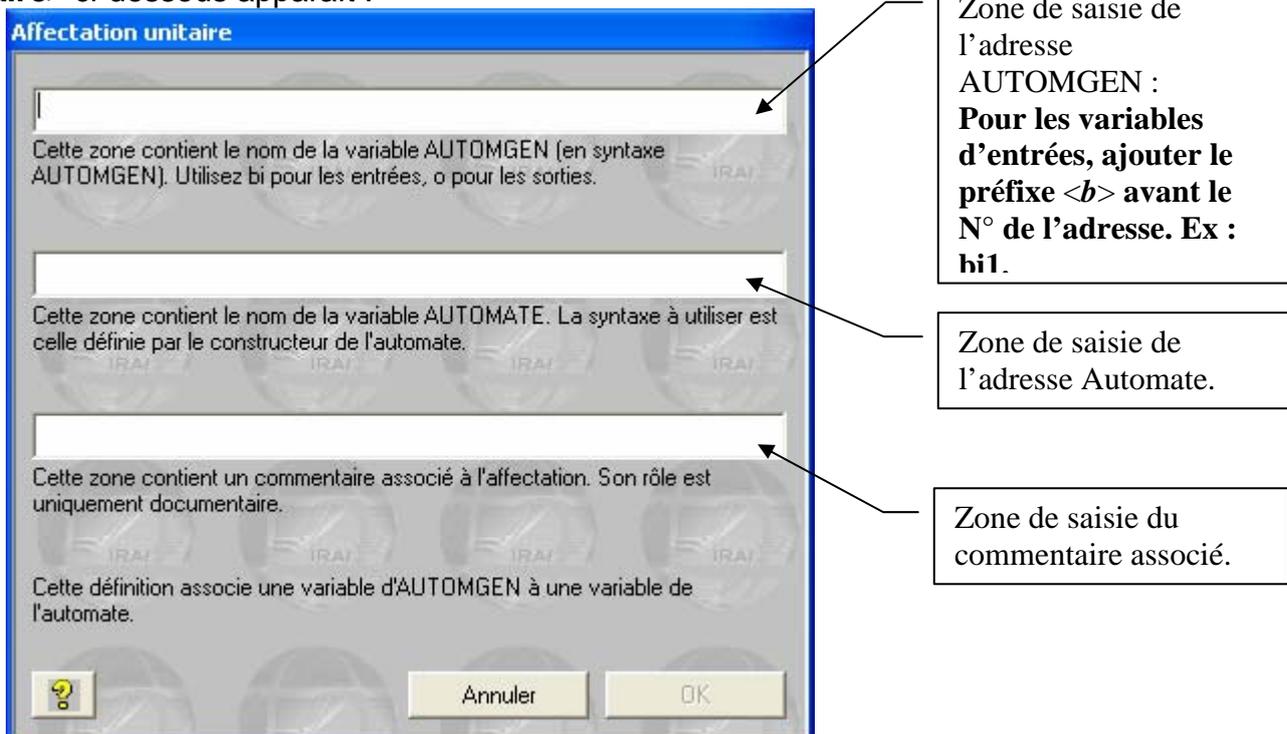


Sélectionner **<Ajouter>** dans le menu qui apparaît.

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



Sélectionner le <Domaine de validité de l'affectation> puis valider, la fenêtre <Affectation unitaire> ci-dessous apparaît :



Après avoir complété les zones de saisie, valider. L'adressage apparaît dans la liste ci-dessous.

Eléments	Valeurs	Commentaires
[-] Déclaration de variables		
[-] Affectation unitaire (une variable AUTOMGEN à une variable automate)		
m62	%SW30	temps de cycle en ms
b8	%S11	débordement chien de garde
bi1	%i0.5	BP départ cycle
bi2	%i0.8	
o1	%q1.2	
o11	%q1.14	
[+] 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)		

Une fois l'adressage unitaire terminé, il faut supprimer les zones d'affectation définies dans l'affectation linéaire.

9.4.2. l'Affectation linéaire :

L'affectation linéaire permet d'affecter automatiquement une table de variables AUTOMGEN à une table d'adresses API. Pour cela l'utilisateur doit définir des zones d'affectation.

Pour ajouter une zone d'affectation, cliquer avec le **bouton droit** de la souris sur **<Affectation unitaire>**. Sélectionner **<Ajouter>** dans le menu qui apparaît.

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)		
... <-32> bi0	%I1.0	
... <-32> o0	%Q2.0	
... <-64> tempo	0	
+ Affectation automatique (un ou plusieurs types de variables AUTOMGEN à une table de variables automate)		

Compléter la boîte de dialogue suivante :

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.

Zone de saisie de la 1^{ère} adresse

Zone de saisie de la 1^{ère} adresse

Zone de saisie de la Longueur de la table.

Zone de saisie d'un commentaire.

OPTION DE CONNEXION

Cet élément permet de choisir le mode de connexion et de définir les paramètres de communication.

Options de connexion

Connecter et télécharger

Ne plus ouvrir cette boîte de dialogue, toujours utiliser l'option choisie ci-dessus

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 ».

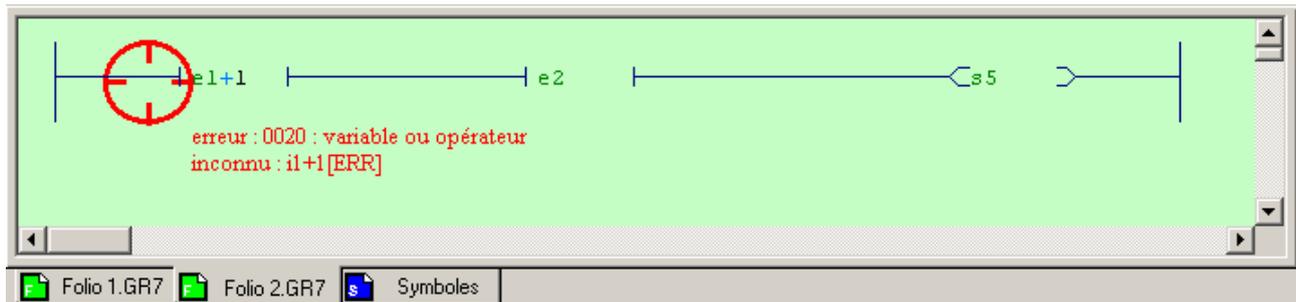
MODULE DE COMMUNICATION

Cet élément permet de définir le port de communication et le paramétrage du port.

10 - EXÉCUTER UNE APPLICATION

COMPILATEUR

Le compilateur  traduit les folios en un ensemble d'équations de langage pivot. Pour localiser une erreur, il faut faire un double clic sur le message d'erreur et le logiciel renvoie sur l'erreur.



EXÉCUTION

Le bouton GO représente la méthode la plus rapide pour observer le résultat de l'exécution d'une application.

Il active les mécanismes suivants :

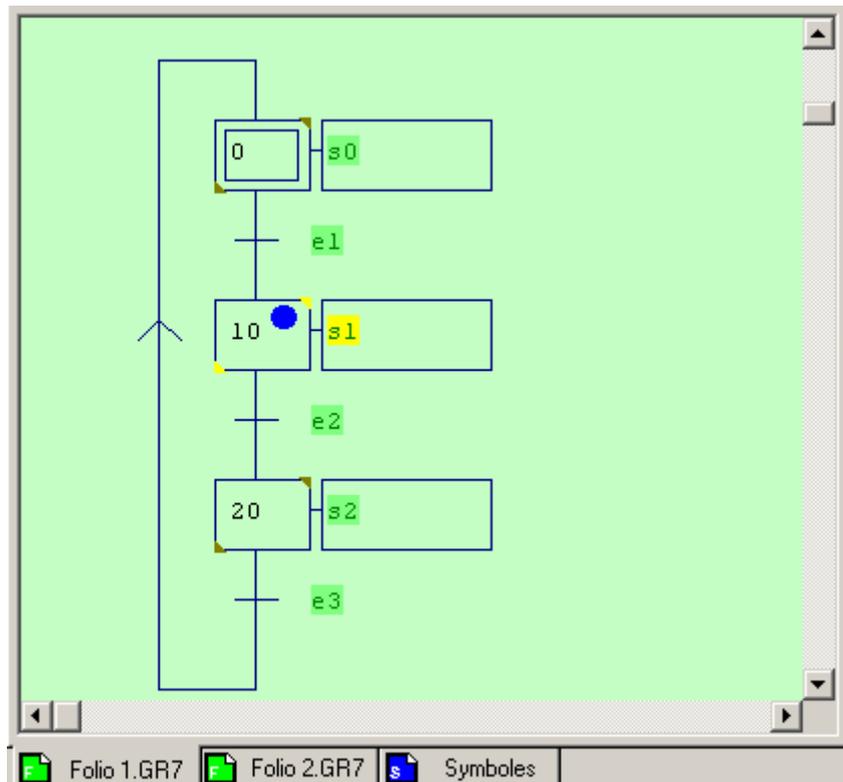
- Compilation de l'application si elle n'est pas à jour
- Installation du module d'exécution
- Passage de la cible en RUN
- Activation de la visualisation dynamique

Sinon on peut lancer par le bouton RUN, une fois qu'on a compilé le projet sans erreurs.

Remarque :

L'activation de la visualisation dynamique se fait par le bouton . Au niveau de la visualisation en dynamique sur les folios :

- Vert correspond à l'état 0
- Jaune correspond à l'état 1



SUR PC

Dans la fenêtre « projet », on sélectionne l'onglet « cibles ».

Ensuite on sélectionne l'option « Exécuteur PC » pour exécuter l'application sur PC.

SUR AUTOMATE

Dans la fenêtre « projet », on sélectionne l'onglet « cibles ».

Ensuite on sélectionne l'automate correspondant.

On compile l'application et on lance la connexion avec laquelle on doit définir le type de liaison avec l'automate. Ceci permet de transférer le programme dans l'automate.

Programmation du Run/Stop et du Chien de Garde pour TSX 17

AUTOMGEN - [VSY PL72]

Fichier Edition Affichage Programme Outils Fenêtre Aide

Projet : (sans nom)

- Folios
- Symboles
- Configuration
 - Post-processeurs
 - PC
 - PI7 (Tsx 37 & Tsx 57)
 - PL72
 - Systeme
 - Correspondance des variables
 - Code constructeur démarrage
 - Code constructeur fin
 - Options de connexion
 - Module de communication
 - TSX 07
 - PL71
 - ZELIO
 - FESTO
 - TWIDO
 - Options du compilateur
 - Documentation
 - fichiers Générés
 - Mise au point
 - Iris
 - Autosim
 - Ressources
 - Modules externes

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)	0	aucune
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 I0,0	RUN/STOP	entrée RUN/STOP
Sortie O0,0	SECURITY	sortie de sécurité
Entrée I0,24	NORMAL	entrée normale
Entrée I0,25	NORMAL	entrée normale
Configuration logicielle		
Mots constants		
Blocs textes		
Options de génération de code (attention, modifier avec précaution)		
Déclaration de variables		

Messages

Bienvenue dans AUTOMGEN V7.101, logiciel démarré à 17:17 55

Infos Compilation Mise au point

NUM N LYCCM 210249