

Fiche de programmation SoMachine 4.1

Automate SCHNEIDER M241



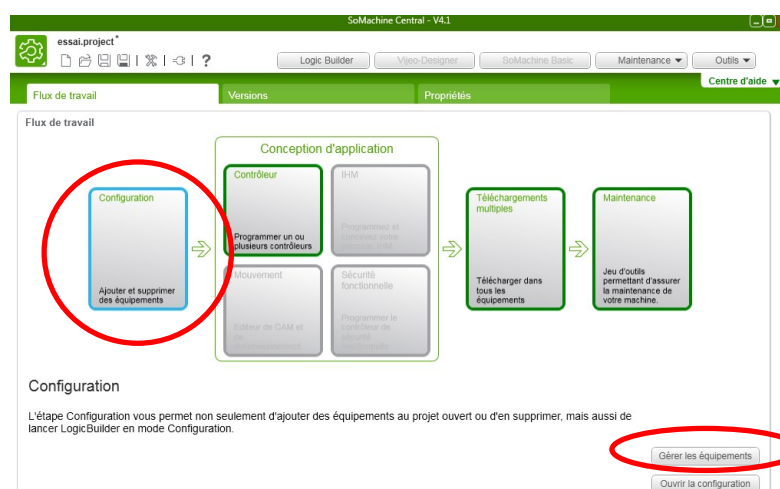
Lancez le programme SoMachine sous Windows en cliquant 2 fois sur l'icône. Passez au paragraphe 1 ou 2.

1. Ouverture d'un fichier existant :

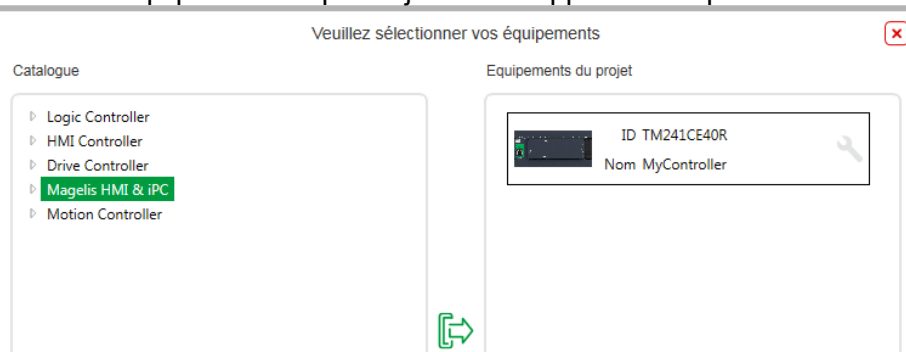
Dans l'écran d'accueil, cliquez sur « Ouvrir le projet » et sélectionnez votre projet.

2. Création d'une programmation :

Dans l'écran d'accueil, cliquez sur "Nouveau projet", puis sur "Projet vide". Enregistrer votre projet.



Cliquez sur « Gérer les équipements » pour ajouter les appareils. Déposer un contrôleur et un HMI.



Remarque :

Lorsque le mode simulation est coché dans Logic Builder, l'automate se transforme en

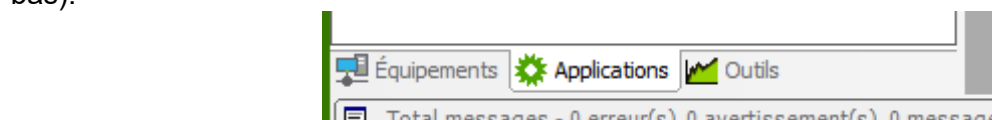


Cliquez sur « Ouvrir la configuration » pour configurer l'automate.

Pour ajouter une carte d'extension, choisir un module à droite et le faire glisser à gauche sur l'automate.

3. Ecriture des mnémoniques :

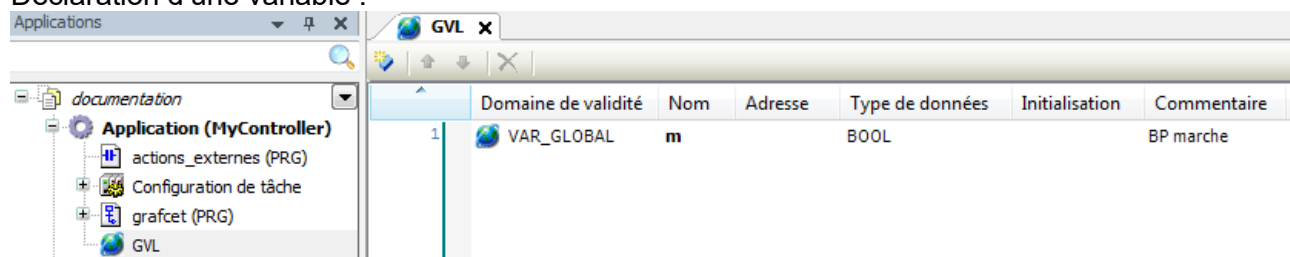
Cliquez sur Logic Builder pour entrer le programme automate et allez dans l'onglet Applications (en bas).



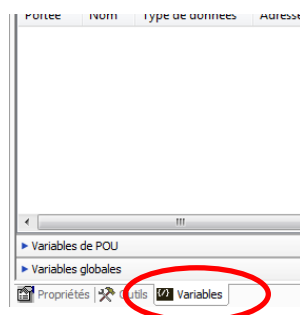
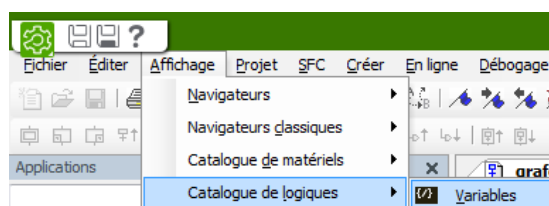
Il n'est pas nécessaire, à ce stade, de déclarer toutes les variables utilisées dans le programme. Cela se fera au fur et à mesure de la création de chaque variable nouvelle.

Les variables mémoires ou internes se déclarent en GVL (Global Variable).

Déclaration d'une variable :



Accès aux variables :



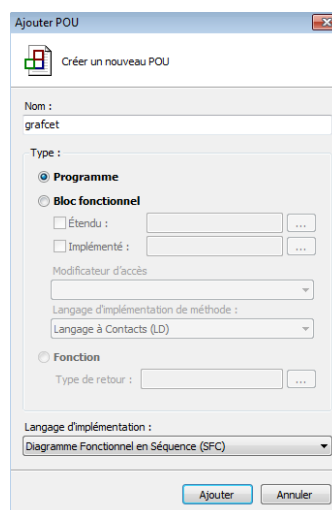
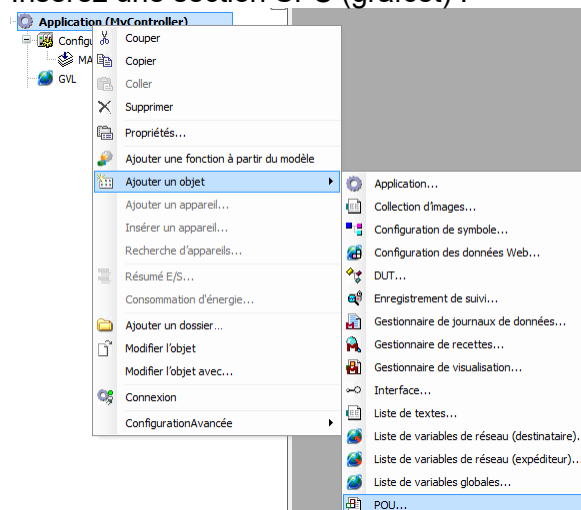
Accès aux variables :
Un onglet « variables »
est ajouté à droite.

4. Ecriture du programme :

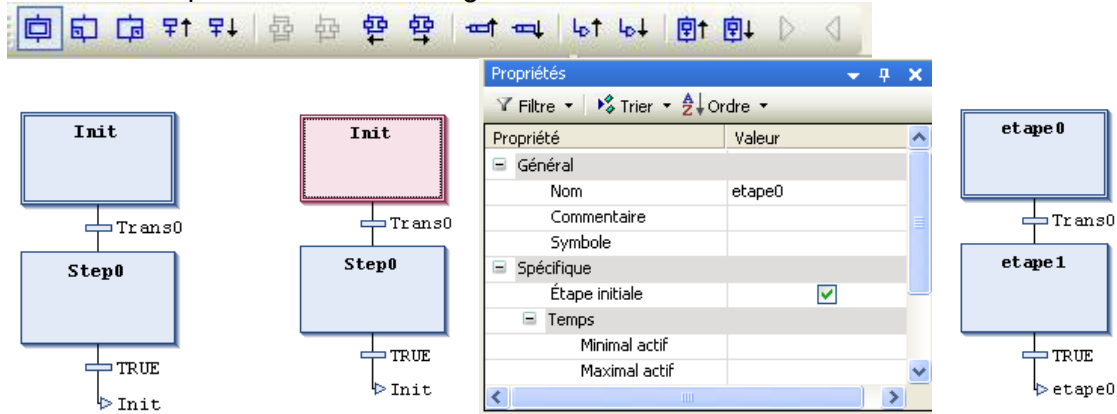
Choix en fonction du langage de programmation utilisé :

- ➔ Programmation en langage SFC (grafcet) :
- ➔ Programmation en langage LD d'un grafcet : voir annexe 5 page 16.

Insérez une section SFC (grafcet) :



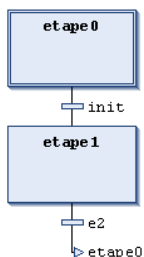
Barre d'outils pour la création d'un grafcet :



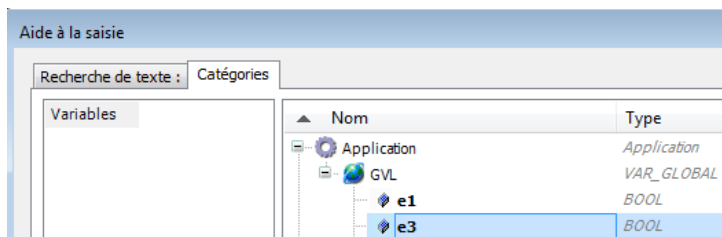
L'étape initiale se définit dans la propriété de l'étape (à droite).

Transitions :

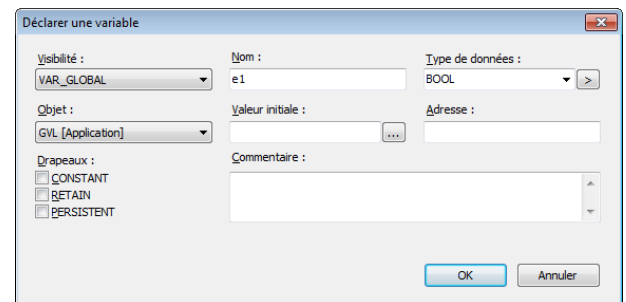
Cas d'une seule variable :



Il suffit d'indiquer le nom de cette variable dans la boîte de dialogue.

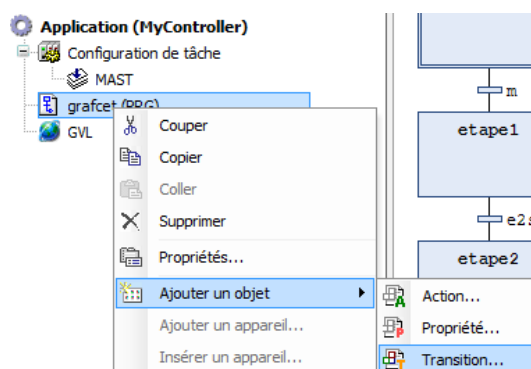


Variable existante

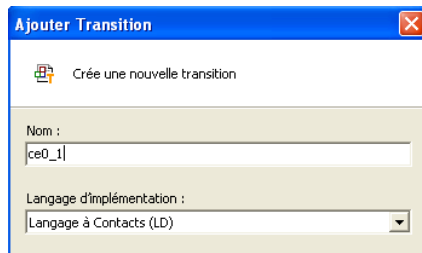


ou création d'une nouvelle variable

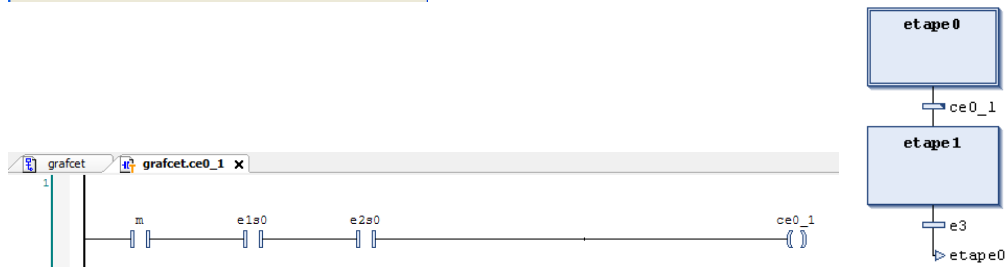
Cas d'une combinaison de variables ou d'une transition inversée (équation booléenne) :



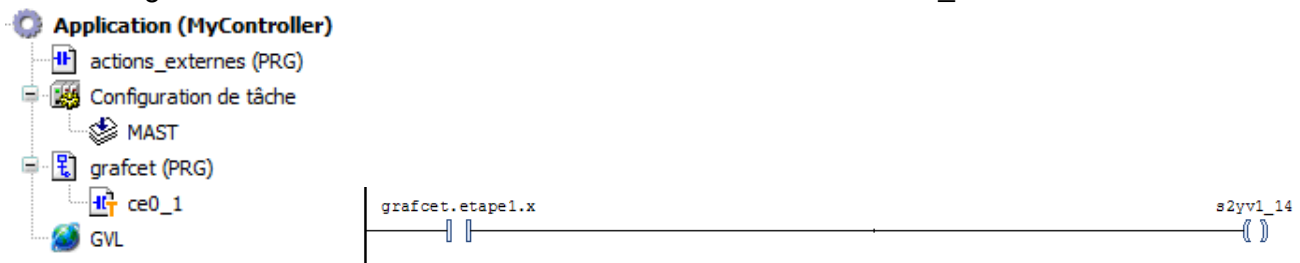
Il faut ajouter une section TRANSITION, lui donner un nom et cliquer sur Editer.



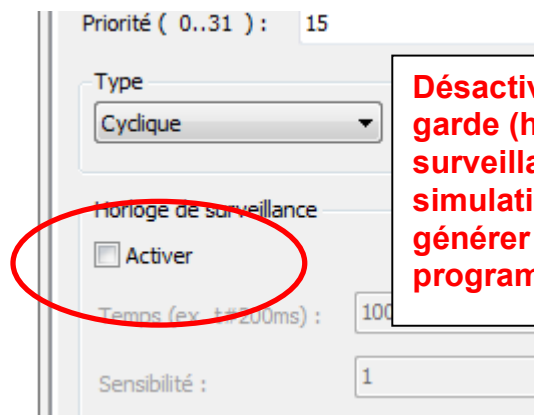
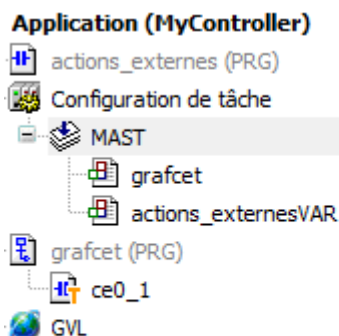
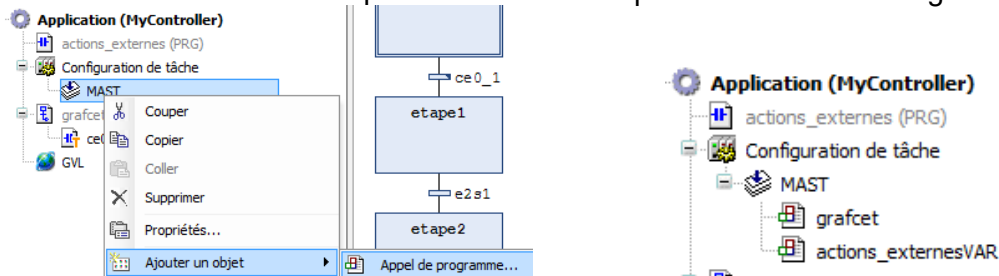
Il faut choisir le langage et compléter la section TRANSITION.



Programmation des sorties dans une section LADDER « Actions_externes » :

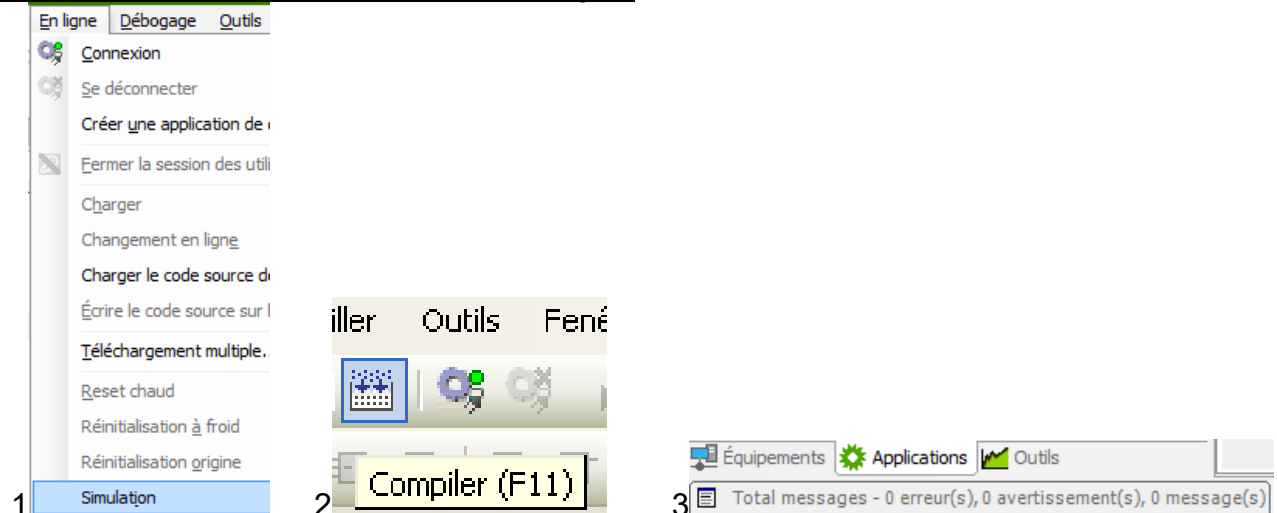


Définir les sections qui seront exécutées par le contrôleur "config tâche → MAST".

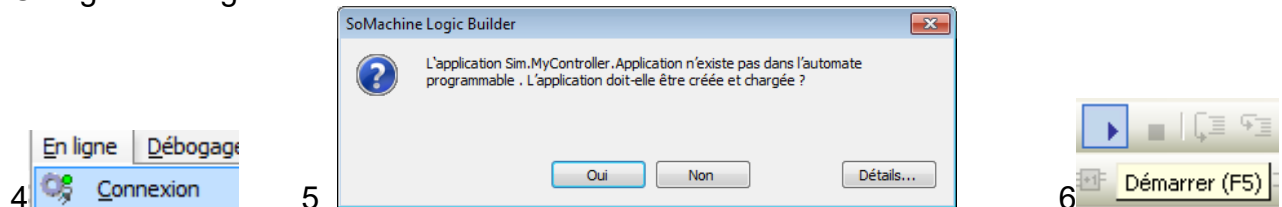


Désactiver le chien de garde (horloge de surveillance) pour la simulation, cela peut générer une exception programme

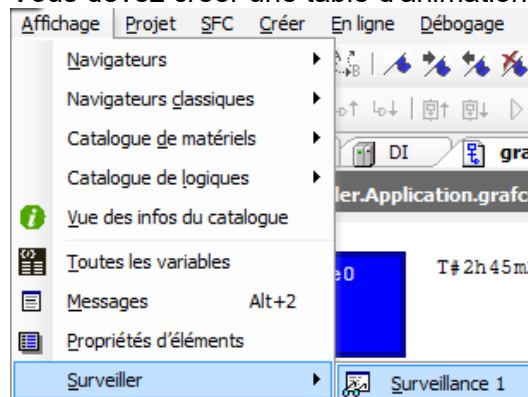
5. Test du programme avec le simulateur intégré :



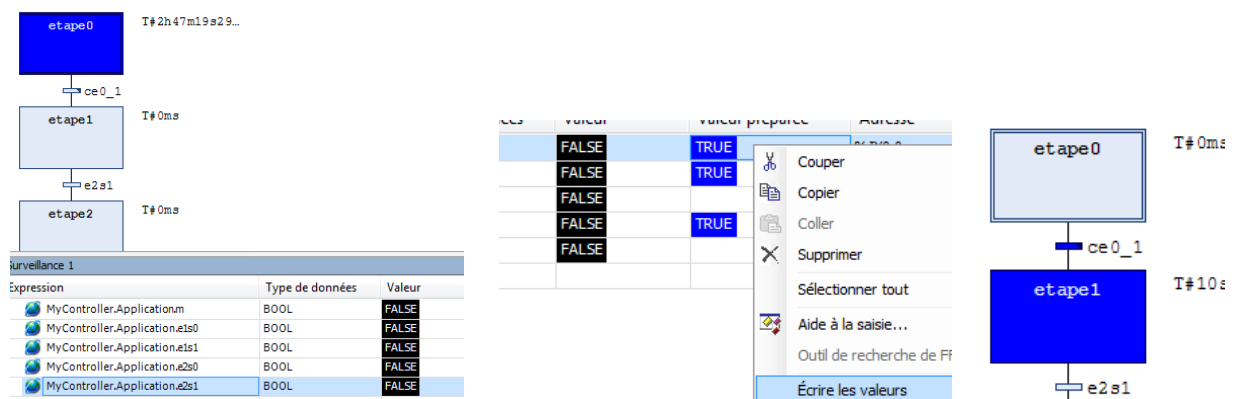
Corriger les bugs éventuels.



Vous devez créer une table d'animation (menu surveiller).



Ajoutez les variables.

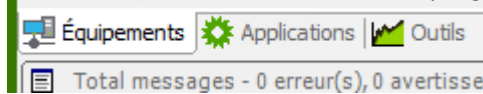


Visualisez vos variables grâce à cette table.

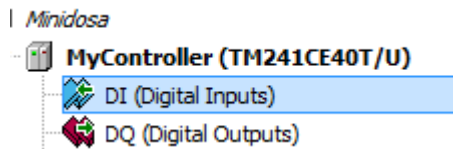
Modifiez les valeurs des variables d'entrées dans "valeur préparées" et écrivez les nouvelles valeurs. Testez le programme et corrigez les erreurs.

6. Test réel :

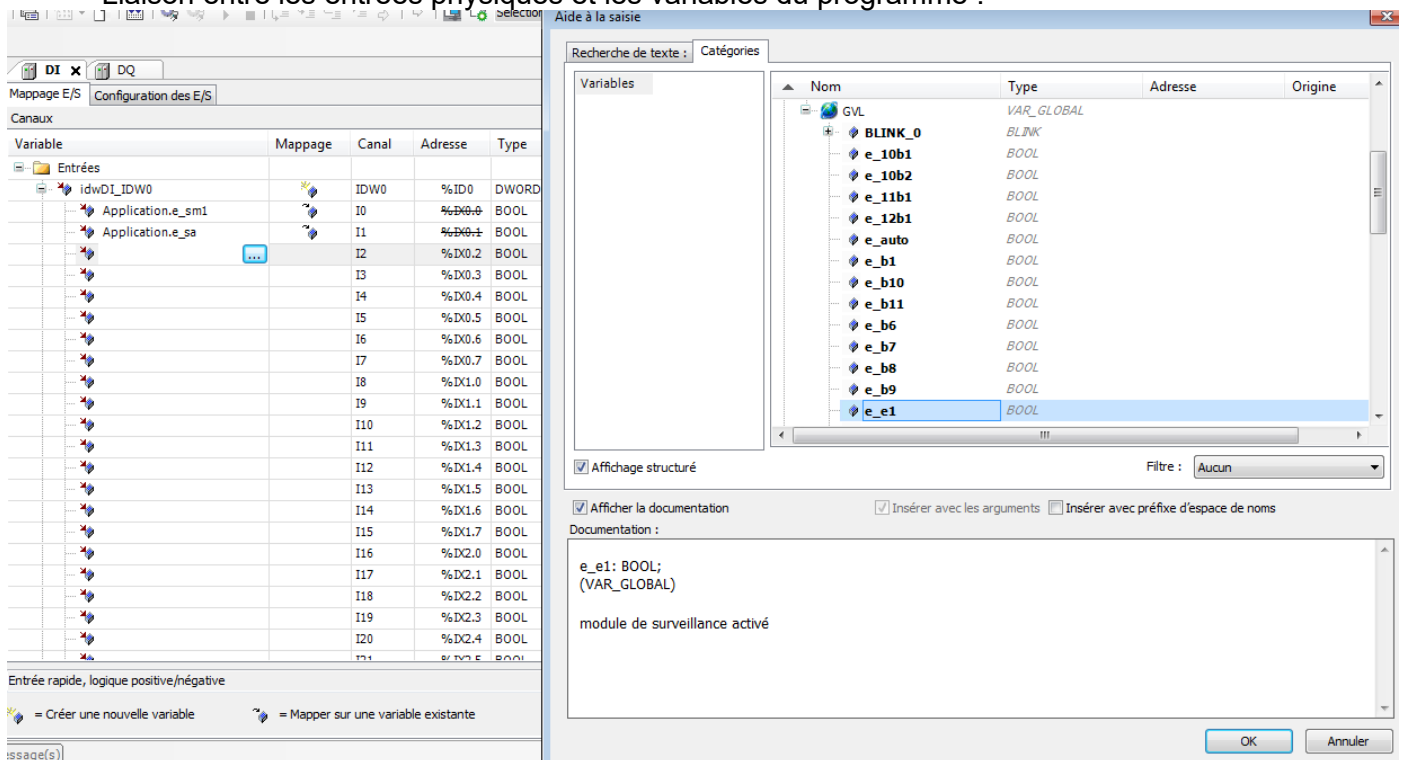
Mnémoniques d'entrées sorties (onglet équipement) :



Les entrées se déclarent dans "DI" :



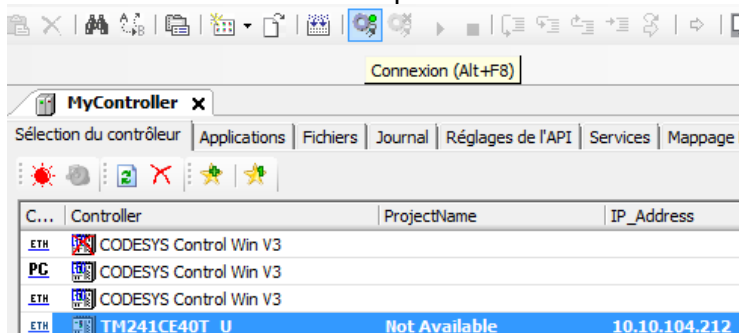
Liaison entre les entrées physiques et les variables du programme :



Les sorties se déclarent dans "DQ" :

Connexion et transfert

Connecter un câble réseau entre l'automate et le PC
Sélectionner l'automate et cliquez sur connexion

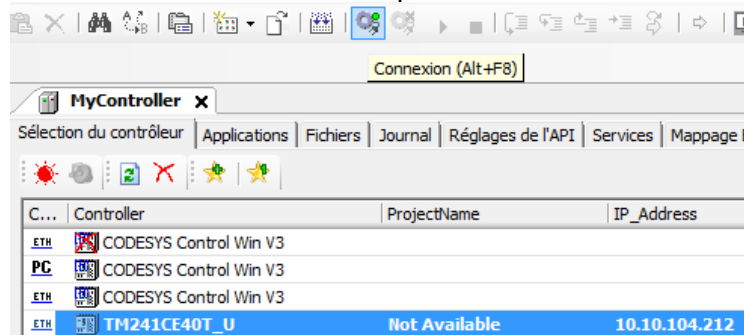


Testez votre programme.

Connexion et transfert (cas d'un AUTOMATE NEUF)

Connecter un câble réseau entre l'automate et le PC

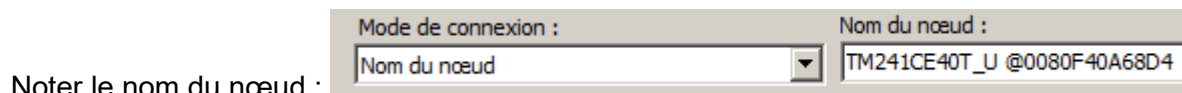
Sélectionner l'automate et cliquez sur connexion



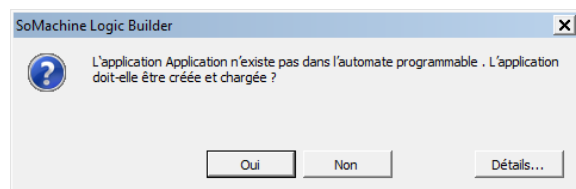
Placer le PC dans le réseau du contrôleur

Adresse IPv4 10.10.104.1

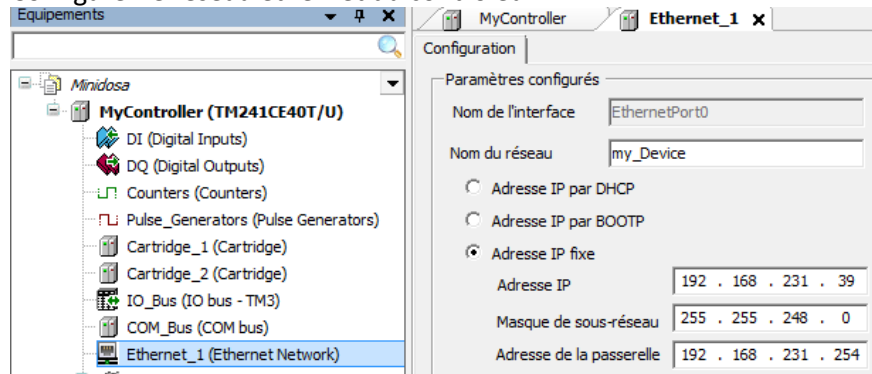
Masque de sous-réseau ... 255.0.0.0



Noter le nom du nœud :



Configurer le réseau ethernet du contrôleur



Transférer les modifs et replacer le PC dans le nouveau réseau de l'automate.

Reconnecter le PC au contrôleur.

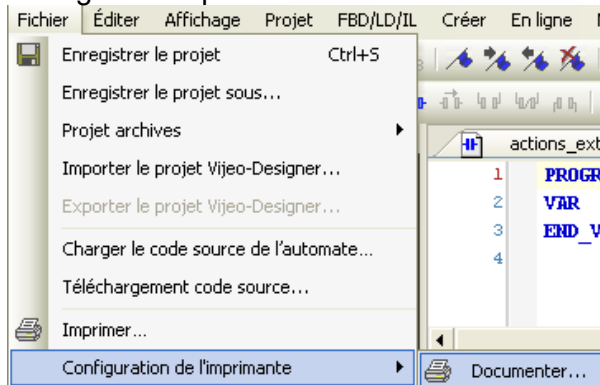
Sauvegarde de l'application dans l'EEPROM de l'automate :



Faire ensuite un téléchargement complet.

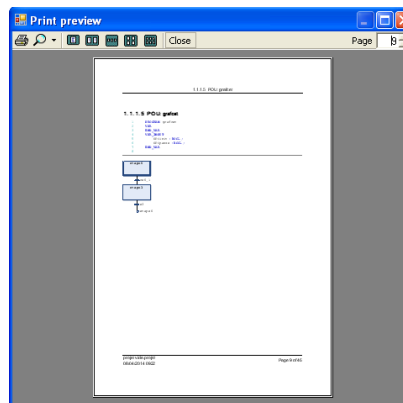
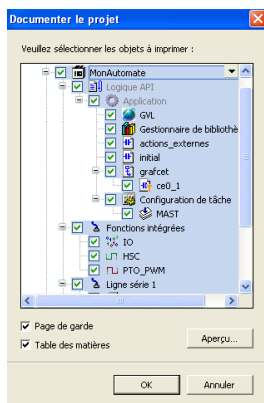
7. Impression :

Configurez l'impression.



Seule la rubrique programme contient votre travail, donc ne sélectionnez qu'elle.

Le reste des rubriques (pour un TP) est superflue à imprimer.



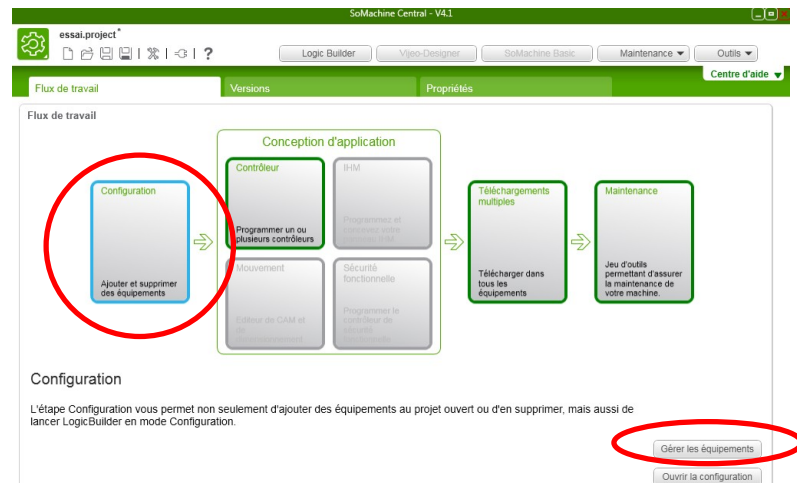
Voir aussi "onglet rapport".

8. Sauvegarde et sortie :

Cliquez sur « Enregistrer sous », sélectionnez le répertoire de votre classe, donnez un nom « xxxxx.project » (xxxxx correspondant à votre nom) et cliquez sur OK.

Cliquez sur « Fichier » puis sur « Quitter » pour quitter le programme.

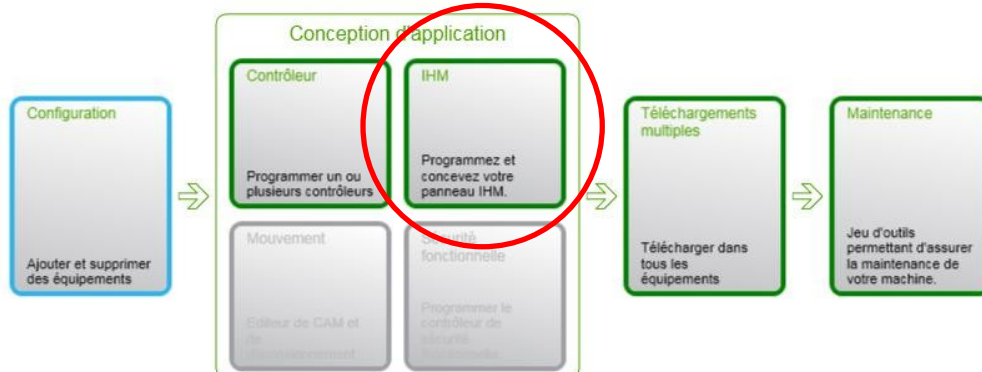
Partie HMI : Vijeo Designer



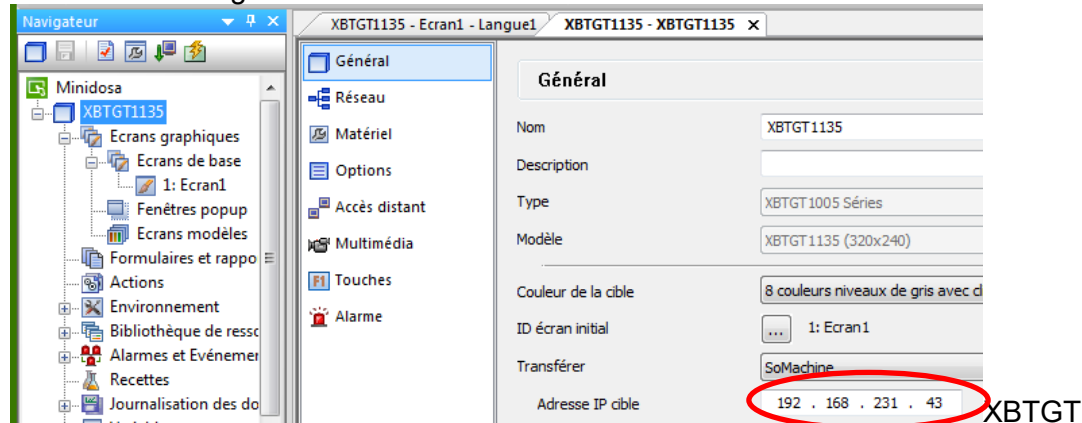
Ajouter un HMI.



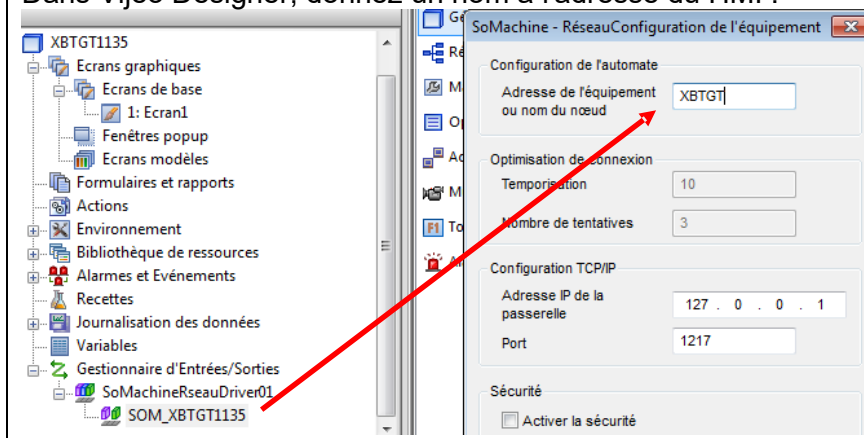
Accès à Vijeo Designer (conception de la partie HMI)



Effectuer la configuration réseau : adresse IP du HMI.

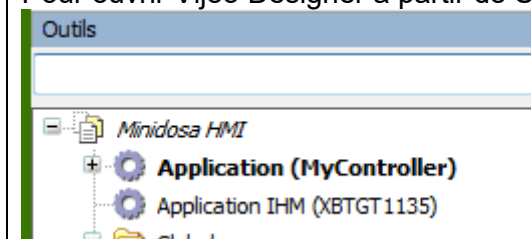


Dans Vijeo Designer, donnez un nom à l'adresse du HMI :



????????????????

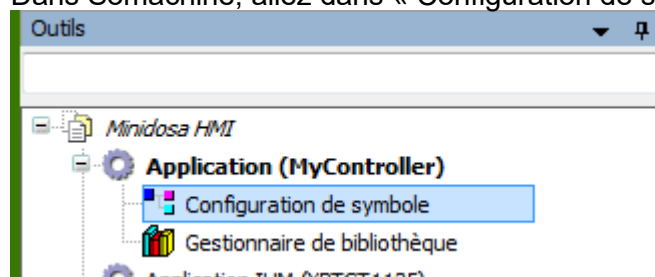
Pour ouvrir Vijeo Designer à partir de Somachine :

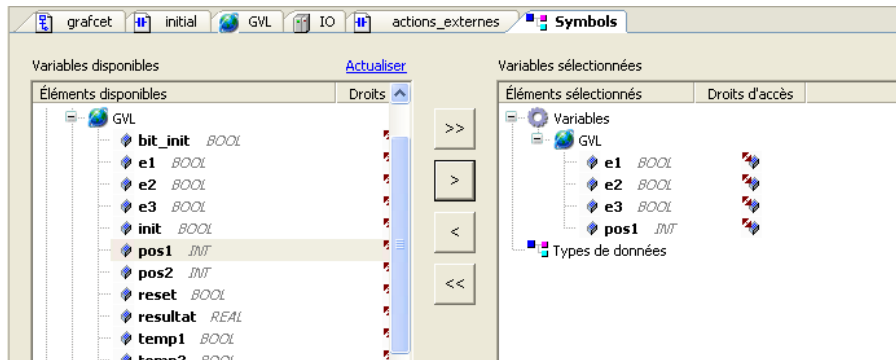


Dans l'onglet "programmation", le HMI apparaît. Double cliquez sur "HMI Application" pour ouvrir VijeoDesigner.

Variables partagées entre Somachine et VijeoDesigner

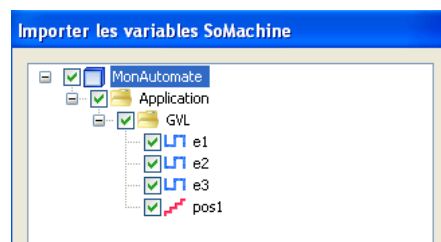
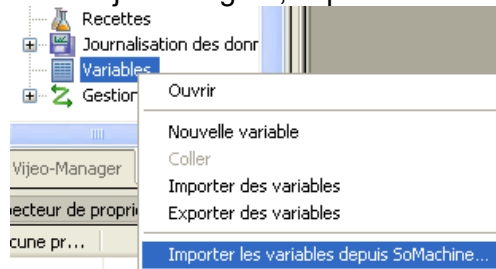
Dans Somachine, allez dans « Configuration de symbole » :





Cochez dans la colonne « droits d'accès » pour partager les variables (nécessite une compilation du programme).

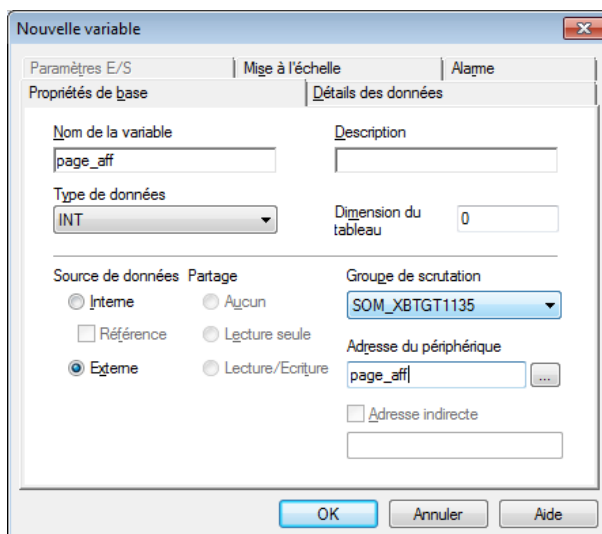
Dans Vijeo Designer, importation des variables Somachine

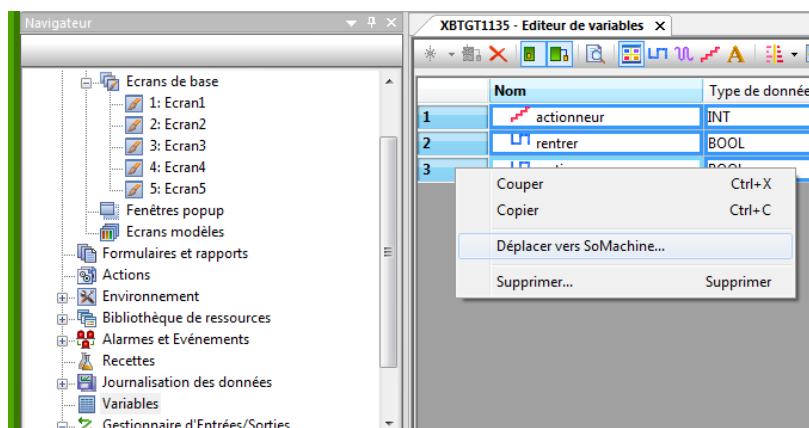


Sélectionnez les variables à importer.




HMISTU655 - Editeur de variables							
	Nom	Type de données	Source de donn...	Groupe de scrut...	Adresse du péri...	Groupe d'alarmes	Groupe de journ...
1	MonAutomate						
	Application						
	GVL						
	e1	BOOL	Externe	SOM_MonAutom...	Application.GVL.e1	Désactivé	Aucun
	e2	BOOL	Externe	SOM_MonAutom...	Application.GVL.e2	Désactivé	Aucun
	e3	BOOL	Externe	SOM_MonAutom...	Application.GVL.e3	Désactivé	Aucun
	pos1	INT	Externe	SOM_MonAutom...	Application.GVL.pos	Désactivé	Aucun

Et le contraire, création de variables dans Vijeo que l'on veut partager dans Somachine :



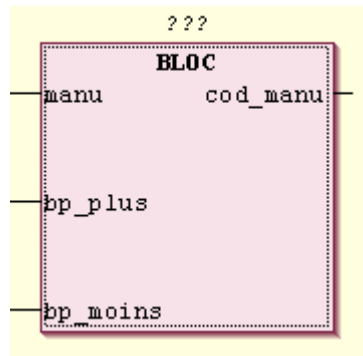
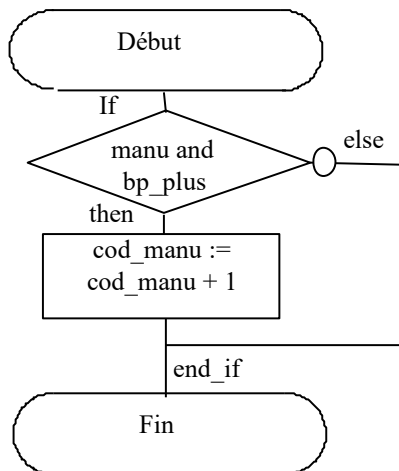


Sélectionnez les variables, clic droit et Déplacer vers SoMachine GVL et OK

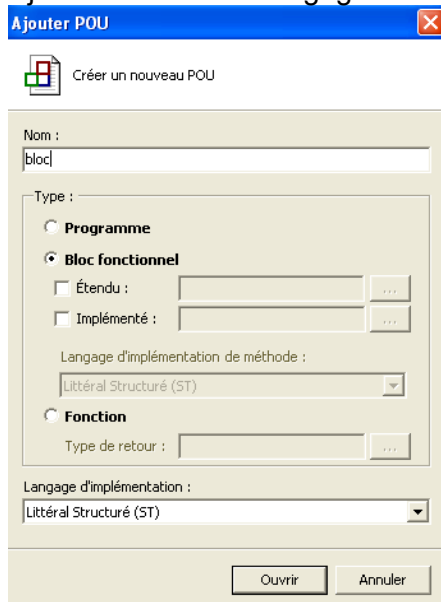
43		VAR_GLOBAL	rentrer	BOOL
44		VAR_GLOBAL	actionneur	INT
45		VAR_GLOBAL	sortir	BOOL

Annexe 2 : programmation d'un bloc fonctionnel en langage structuré

Exemple pour l'algorithme suivant :



Ajouter un POU en langage ST et définissez les entrées sorties.



```

1  FUNCTION_BLOCK bloc
2  VAR_INPUT
3      manu:BOOL;
4      bp_plus:BOOL;
5      bp_moins:BOOL;
6  END_VAR
7  VAR_OUTPUT
8      cod_manu:INT;
9  END_VAR
10 VAR
11 END_VAR
12

```

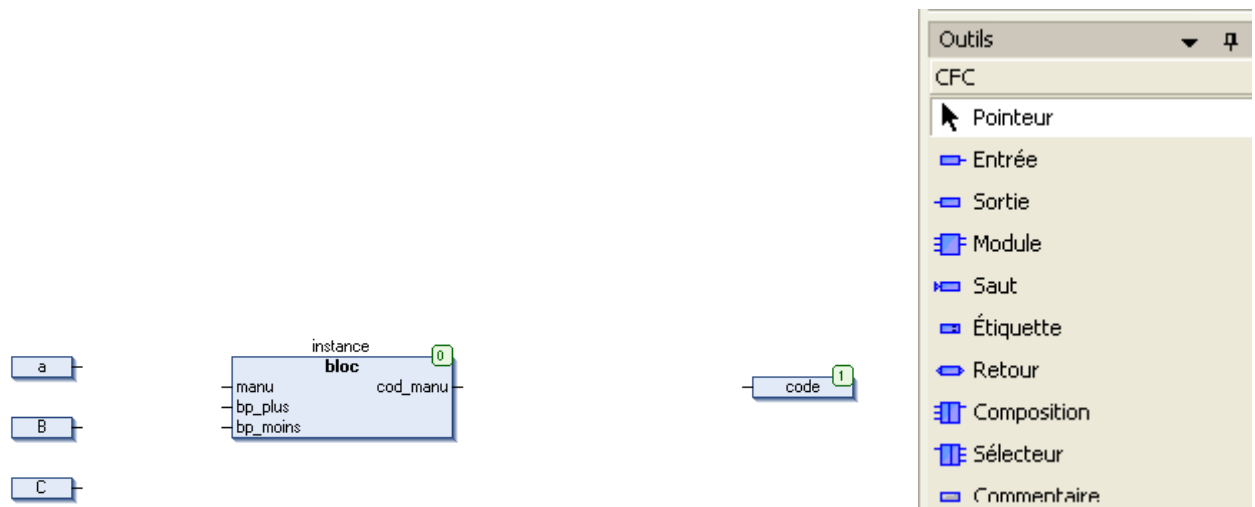
Code du programme :

```

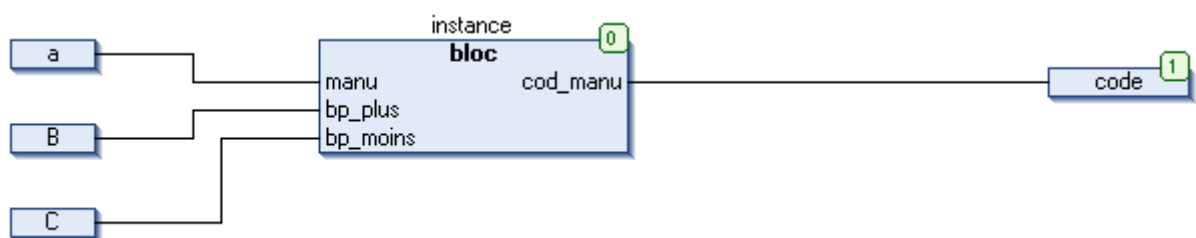
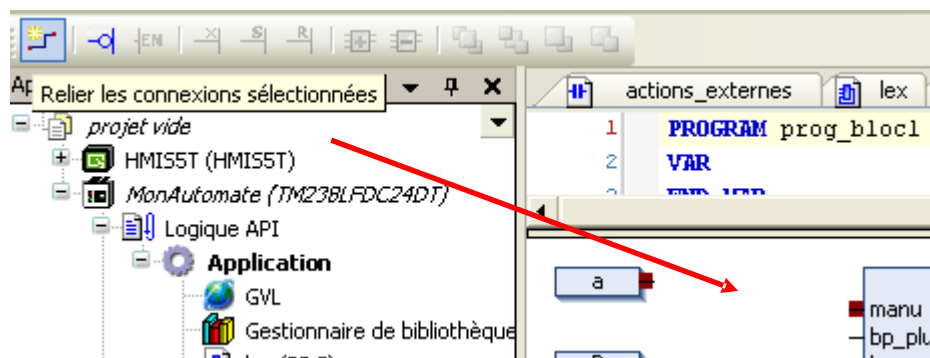
1  IF manu AND bp_plus THEN
2      cod_manu:=cod_manu+1;
3  END_IF

```

Créez un POU en CFC dans l'application principale :



Pour faire apparaître le bloc fonctionnel, utiliser les outils à droite et réalisez les connexions.

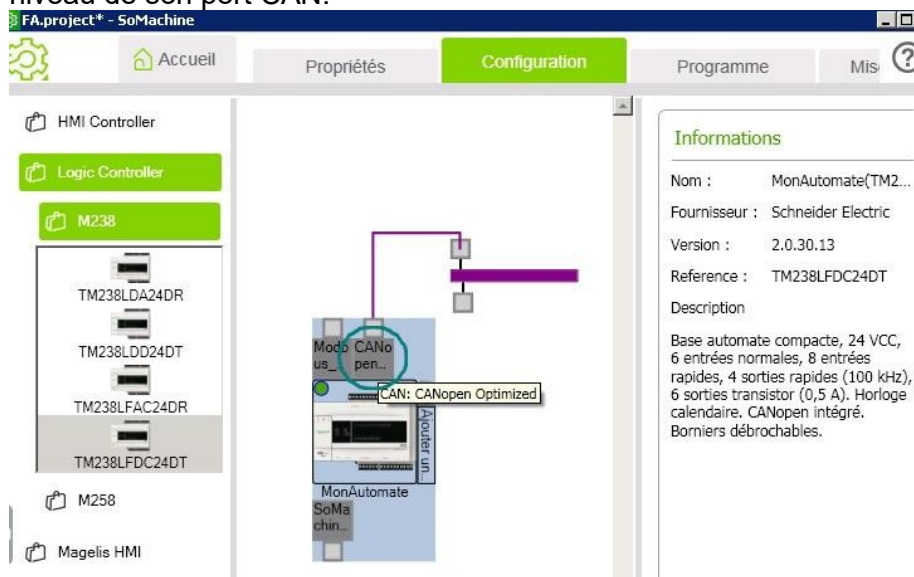


Annexe 4 : Configuration d'un axe lexium32M sur bus CAN

(Configuration de la communication :

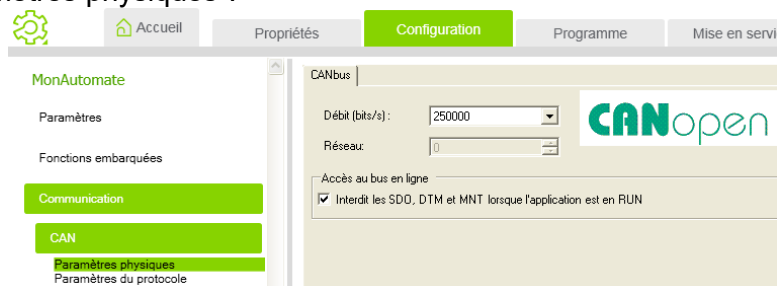
Dans le menu COM du lexium, configurez ADCO (Adresse CANopen) (ici 2) et BDCO (Baud CANopen)(ici 250).

Dans Somachine, ajouter l'équipement **CANopen Optimized** en cliquant sur l'illustration du contrôleur au niveau de son port CAN.



configurez la vitesse du bus CAN de l'automate (ici 250 kBaud) :

Onglet "configuration", double cliquez sur l'automate, cliquez sur "Communication", sur "CAN" et sur "Paramètres physiques".

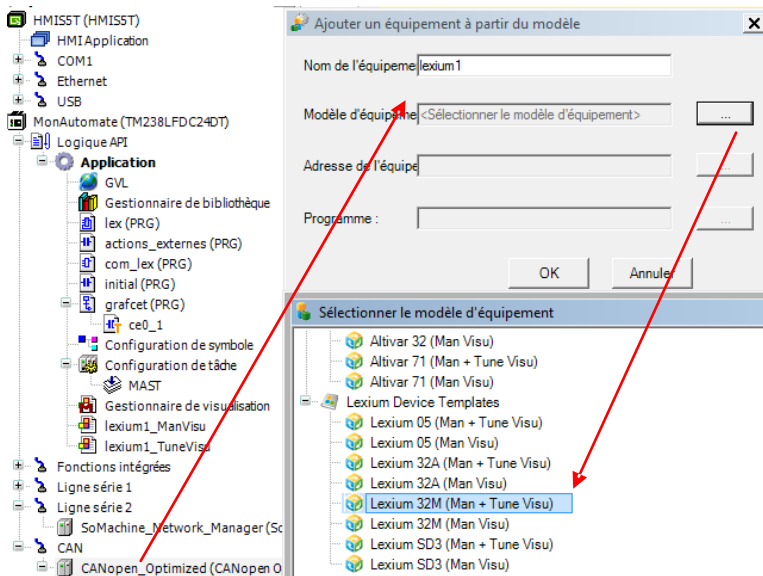


Depuis l'onglet Programme, cliquer droit sur Application et choisir Ajouter un objet > POU... . Nommer le POU et choisir le langage d'implémentation comme par exemple Language CFC. Ajouter dans la tâche MAST l'appel à ce POU en double-cliquant sur MAST.

Ajouter un équipement à partir d'un modèle en cliquant droit sur CANopen_Optimized (CANopen optimized).

Nommer l'équipement et sélectionner le modèle d'équipement Lexium 32 (Man + Tune Visu).

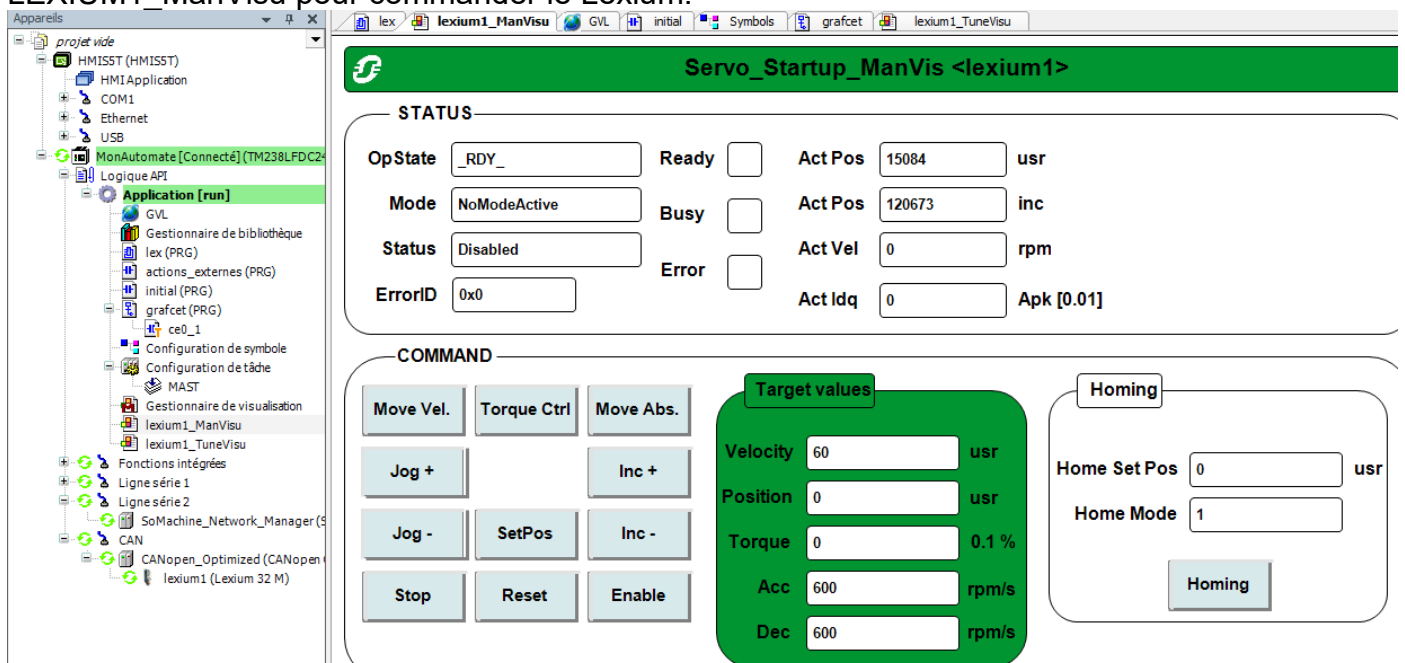
Indiquer l'adresse CAN de l'équipement et préciser le POU recevant le Bloc fonction pour l'équipement d'E/S 'LEXIUM32' (c'est à dire le POU créé précédemment).



-cdt"

La configuration est terminée.

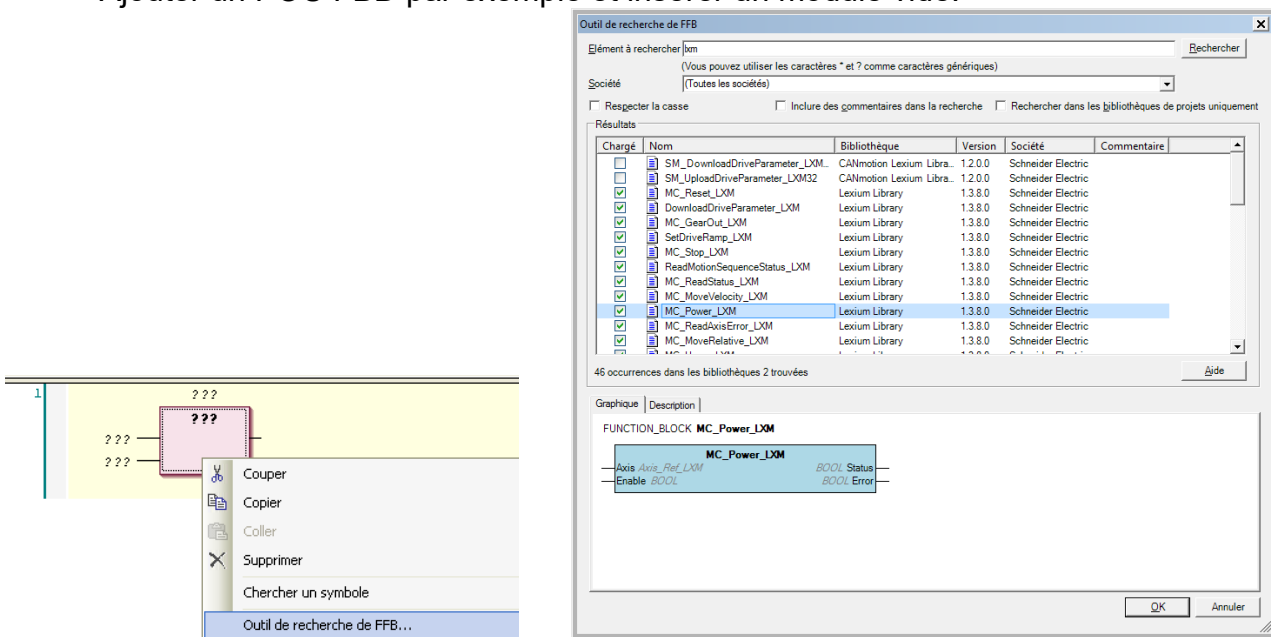
Il suffit de transférer l'application dans le contrôleur et une fois en ligne de double-cliquer sur LEXIUM1_ManVisu pour commander le Lexium.



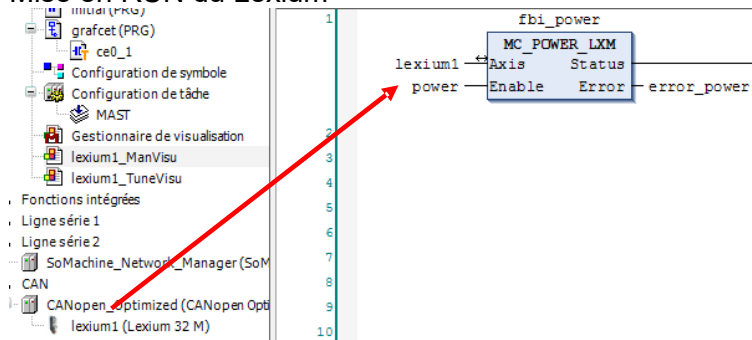
Cliquez d'abord sur Enable, puis sur la commande voulue.

Programmation :

Ajouter un POU FBD par exemple et insérer un module vide.

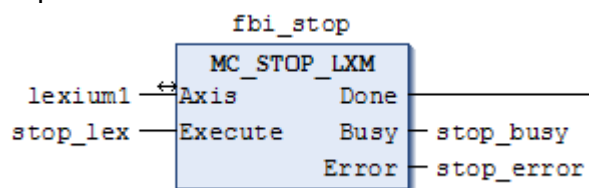


Mise en RUN du Lexium

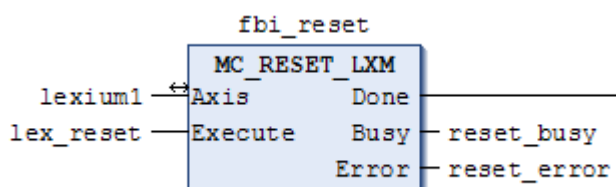


Le nom de l'axe est celui déclaré en configuration.

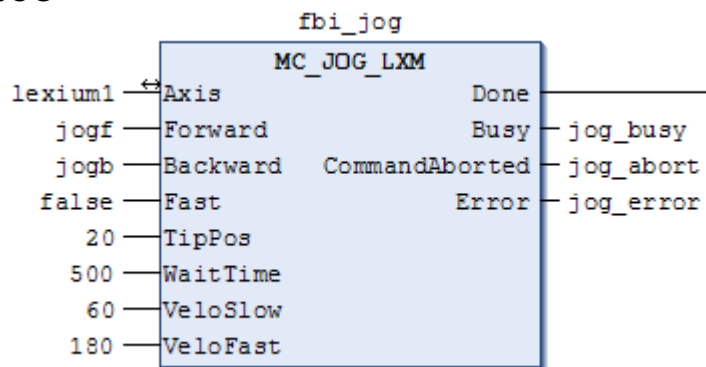
Stop mouvement



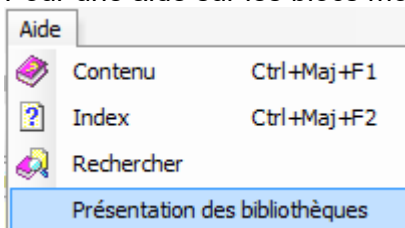
Reset Lexium



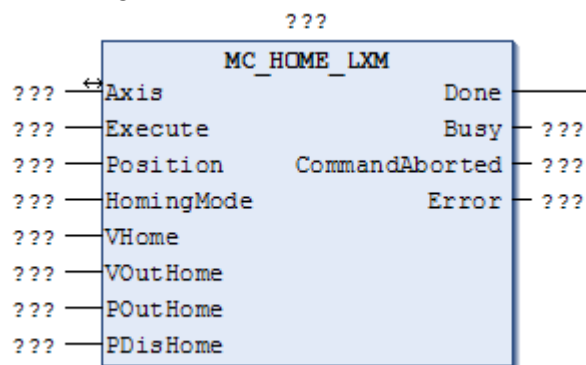
JOG



Pour une aide sur les blocs motion :



Prise d'origine



Type de prise d'origine (extrait doc Lexium)

HomingMode	UINT	<p>Value range:</p> <p>Initial value: 1</p> <p>1: LIMN with index pulse</p> <p>2: LIMP with index pulse</p> <p>7: REF+ with index pulse, inverted, outside</p> <p>8: REF+ with index pulse, inv., inside</p> <p>9: REF+ with index pulse, not inv., inside</p> <p>10: REF+ with index pulse, not inv., outside</p> <p>11: REF- with index pulse, inv., outside</p> <p>12: REF- with index pulse, inv., inside</p> <p>13: REF- with index pulse, not inv., inside</p> <p>14: REF- with index pulse, not inv., outside</p> <p>17: LIMN</p> <p>18: LIMP</p>
------------	------	--

		23: REF+, inv., outside 24: REF+, inv., inside 25: REF+, not inv., inside 26: REF+, not inverted, outside 27: REF-, inv., outside 28: REF-, inv., inside 29: REF-, not inv., inside 30: REF-, not inverted, outside 33: Index pulse in negative direction 34: Index pulse in positive direction 35: Position setting
--	--	--

Voir doc Lexium pour plus de précision sur toutes les méthodes d'indexation.

Fonctionnement :

« power » doit restée à 1 durant tout le fonctionnement : affichage de RUN sur le Lexium. Lorsqu'elle est à 0, affichage de RDY sur le LEXIUM.

« stop_lexium » permet de stopper le mouvement en cours sur le lexium.

« reset_lexium » permet d'acquitter les défauts sur le lexium.

Fiche de programmation SoMachine 4.1	1
1.Ouverture d'un fichier existant :	1
2.Création d'une programmation :	1
3.Ecriture des mnémoniques :	2
4.Ecriture du programme :	2
5.Test du programme avec le simulateur intégré :	5
6.Test réel :	6
7.Impression :	8
8.Sauvegarde et sortie :	8
Partie HMI : Vijeo Designer	9
Annexe 2 : programmation d'un bloc fonctionnel en langage structuré	13
Annexe 4 : Configuration d'un axe lexium32M sur bus CAN	15