FICHE D'APPLICATION GRAFCET-LADDER-ST	PROGRAMMATION AVANCÉE DU P400XI

RÉPERTOIRE DES ÉVOLUTIONS

Version	Date	Auteur	Nature des modifications	Pages
E				
D				
C				
B				
A	**12/08/09**	Thierry Caballé	Création du document	Toutes

Le P400Xi dispose, en standard, d'un ensemble de fonctions logiques et mathématiques qui vous permet d'associer une programmation spécifique à la gestion de vos entrées/sorties.

D'une simple rotation de pompes à la gestion complexe d'un poste assainissement, les fonctionnalités du P400Xi vous amènent loin … mais si vous souhaitez conserver vos acquits de programmation ou développer dans un langage standardisé dans le milieu de l'automatisme, l'atelier Straton se révèle être l'outil qu'il vous faut.

Perax a donc ajouté, aux fonctionnalités du P400Xi, les différents outils de programmation API :

- Grafcet
- Ladder
- Langage ST

Nous allons détailler la mise en route de ces trois modes de programmation.

Ce n'est, en aucun cas, un cours sur ces trois langages … juste une première approche pour vous montrer la facilité d'intégration de ces langages dans notre automate P400Xi.

La création de ces programmes, quel que soit le langage utilisé, se réalise par l'intermédiaire d'un logiciel de développement spécifique : l'atelier Straton.

Commençons donc par la présentation de ce logiciel.

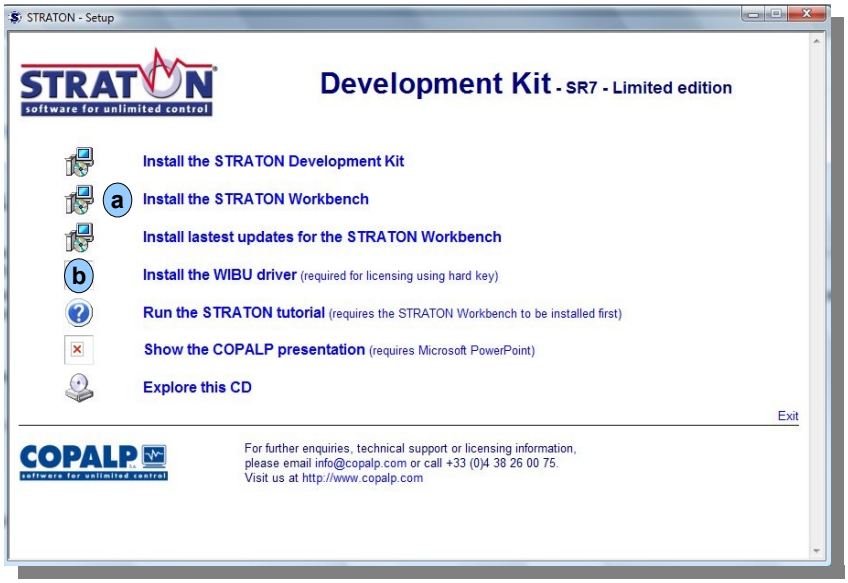
1 L'ATELIER DE DÉVELOPPEMENT STRATON

Vous devez posséder l'atelier complet de développement pour programmer en Grafcet et/ou Ladder; Perax propose, en effet, un atelier limité au langage ST.

En résumé, l'atelier Straton limité Perax vous autorise la programmation en langage ST; l'atelier complet Straton permet la programmation en Grafcet, Ladder et langage ST.

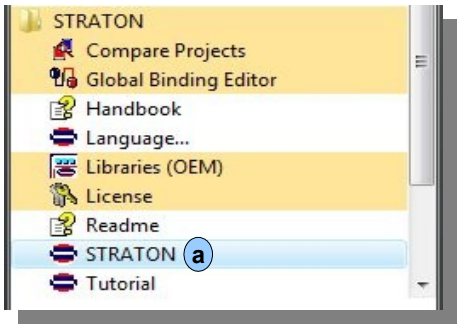
1.1 INSTALLATION DE L'ATELIER STRATON

Insérer le cédérom STRATON ... une fenêtre d'installation apparaît.



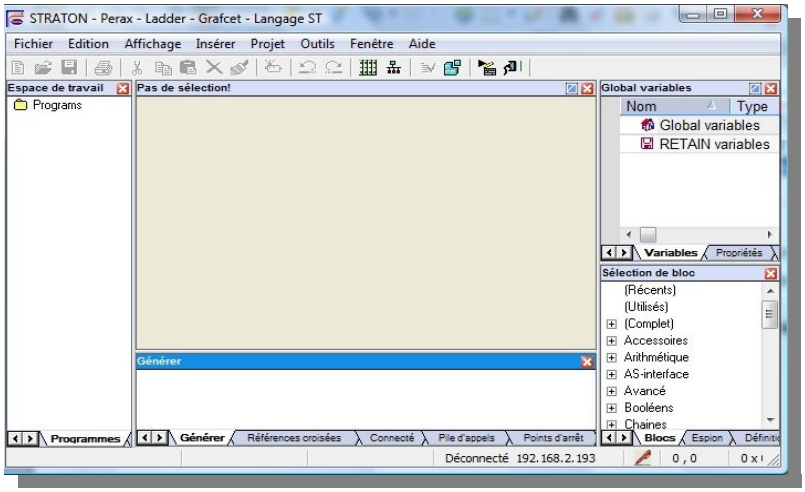
Installez le programme « Straton Workbench » (a) ; si vous possédez une clef de contrôle, installez le driver de la clef « Wibu » (b) puis connectez votre clef.

Après avoir installé l'atelier , lancez l'outil de programmation (a).



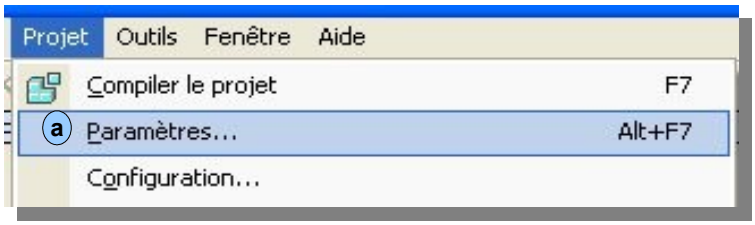
Créez un nouveau projet.

Vous vous retrouvez dans la fenêtre d'édition de programme.

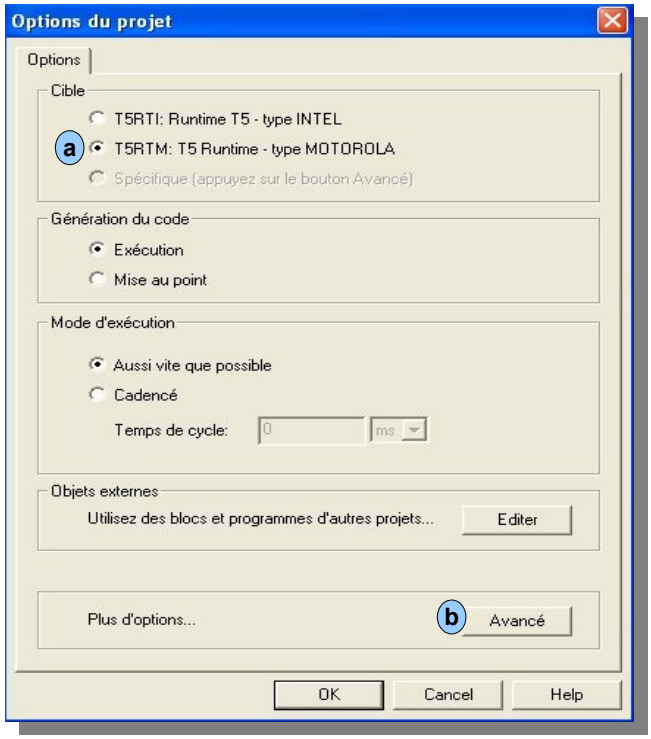


1.2 Paramétrage de l'atelier en vis à vis du P400Xi

Dans le menu « Projet » , choisissez « Paramètres (a).

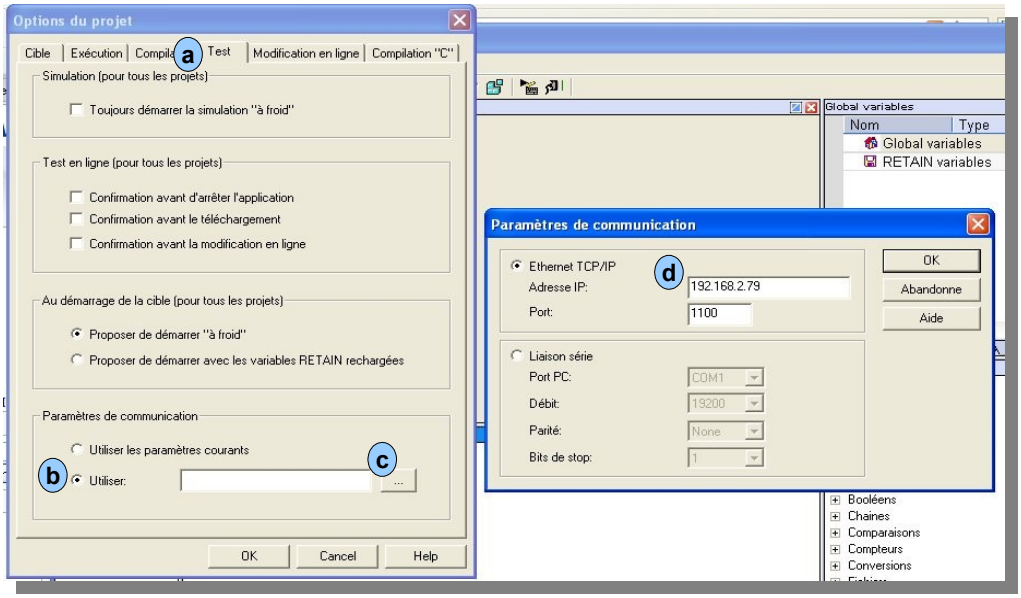


Cochez, dans la partie « Cible » , T5RTM pour une PxiBASE et INTEL pour une PxiCPU (a).

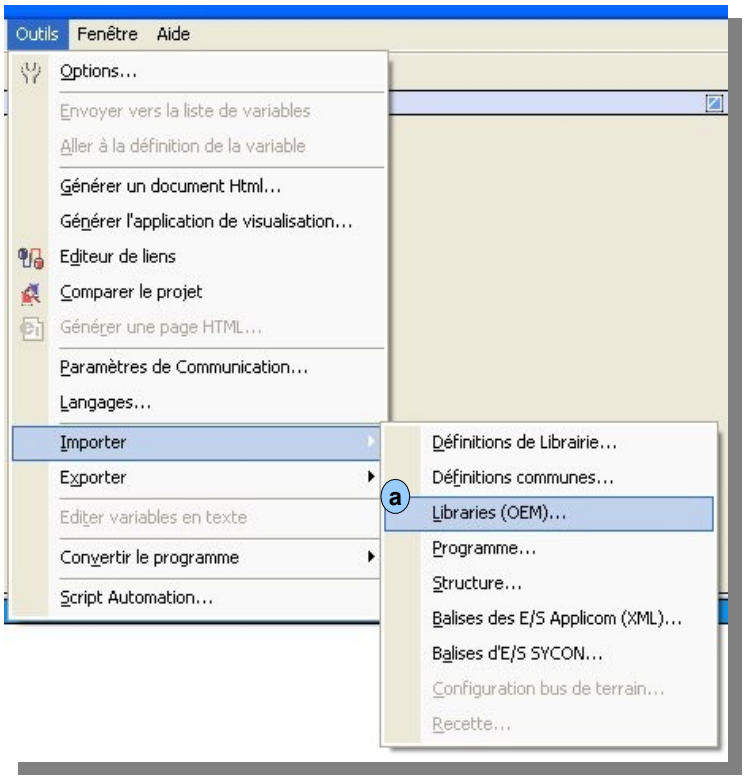


Cliquez sur le bouton « Avancé » . (b)

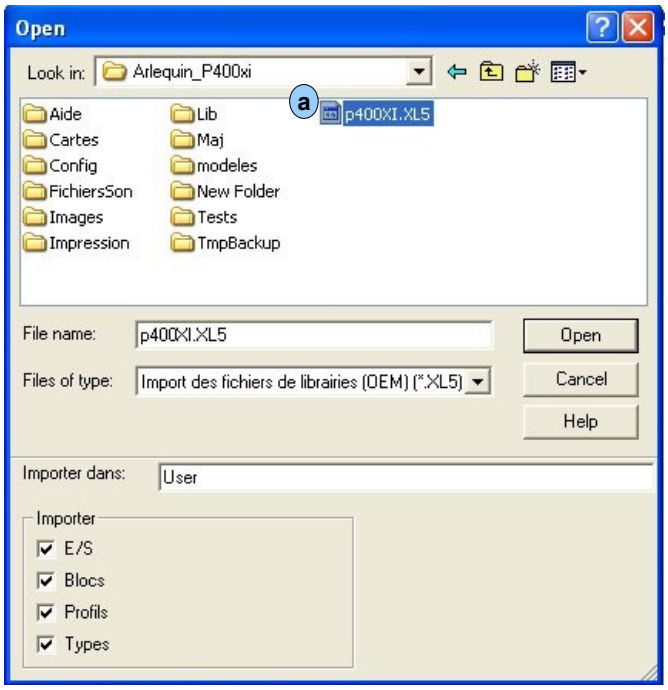
Dans l'onglet « Test » (a), cliquez sur « Utiliser » (b) puis sur le bouton « ... » (c) .
Saisissez l'adresse TCP/IP du P400Xi (d) et cliquez sur les différents boutons « OK » afin de valider votre paramétrage.



Il nous faut maintenant importer la librairie propre au P400XI .
Dans le menu « Outils », choisissez « Importer » puis « Librairies (OEM) » (a) .



Choisissez la librairie fournie par la société Perax. (a)
Validez en cliquant sur le bouton « Open ».



L'initialisation de l'atelier Straton est terminée.

2 LE PROJET COMMUN

Nous allons réaliser, dans les 3 langages (St, Ladder & Grafcet) la gestion d'une intrusion.

C'est une programmation très simple qui prend en compte la lecture d'un badge (le lecteur de badges se trouve à l'extérieur du bâtiment) et le contact d'une porte.

2.1 Déroulement du programme

Dès que l'on détecte le passage du badge, on attend l'ouverture de la porte (pendant une quinzaine de secondes); si la porte s'ouvre, nous sommes en intervention, sinon on revient en attente du badge.

Une porte ouverte sans présence de badge provoque une intrusion.

L 'information « badge passé » reste naturellement active pendant une dizaine de secondes dans le lecteur de badges; nous n'aurons pas à la temporiser dans notre programmation.

Le but de cet exercice est de vous montrer la faisabilité de programmation en ces 3 langages et non la réalisation d'une gestion d'intrusion complète.

2.2 Les informations lues dans le P400Xi

Le badge : entrée Tor
Le contact de porte : entrée Tor

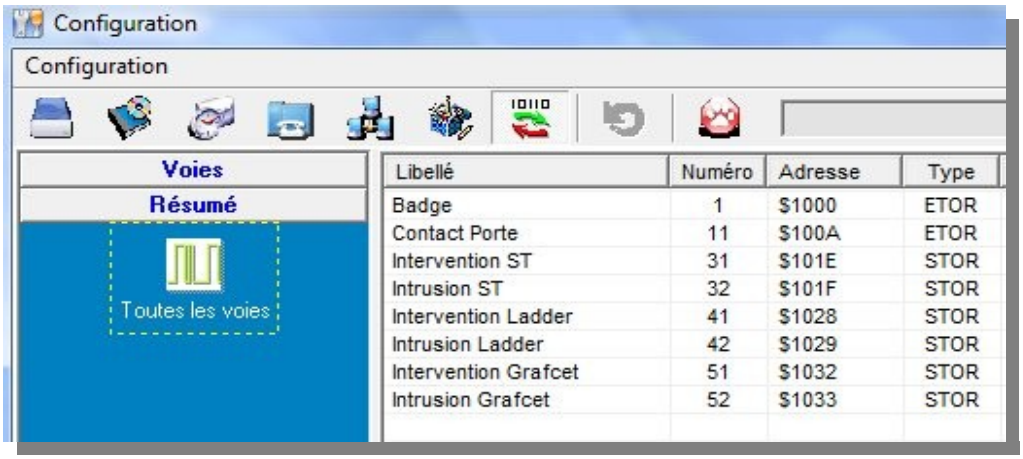
Ces 2 entrées sont communes aux 3 langages de programmation.

2.3 Les informations écrites dans le P400Xi

Intervention : sortie Tor
Intrusion : sortie Tor

Nous avons 2 sorties par type de langage de programmation (donc 6 voies).

Nous allons considérer que le P400Xi est déjà paramétré.



3 LA PROGRAMMATION

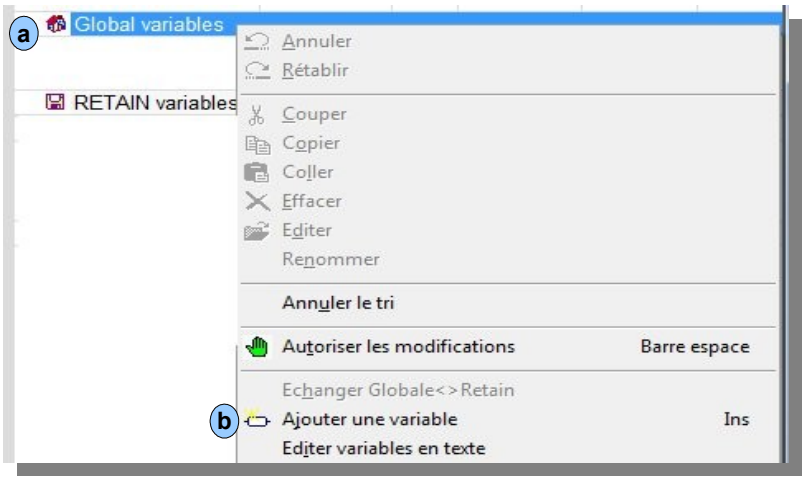
3.1 DÉCLARATION DES VARIABLES COMMUNES AUX 3 LANGAGES

Dans tout langage de programmation, nous trouvons 2 types de variables :

- variables locales : utilisées uniquement dans un programme.
- variables globales : utilisées dans l'ensemble des programmes (cette variable est accessible par l'ensemble des programmes; les modifications apportées par un programme sont répercutées sur l'ensemble des programmes).

Dans la zone de déclaration des variables, nous allons rajouter le contact de porte et la lecture du badge.

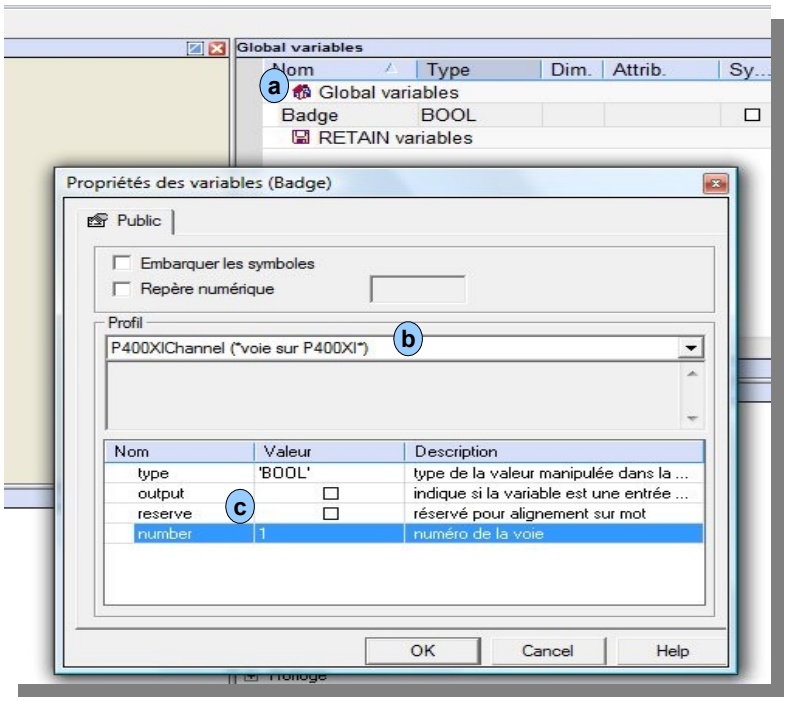
Faîtes un click droit sur « Global variables » (a) et cliquez sur « Ajouter une variable » (b).



La variable NewVar, de type BOOL a été créée.

Pour donner un nom à la variable, faire un double clic sur NewVar; saisissez le nom et VALIDEZ EN APPUYANT SUR LA TOUCHE ENTREE DE L'ORDINATEUR .

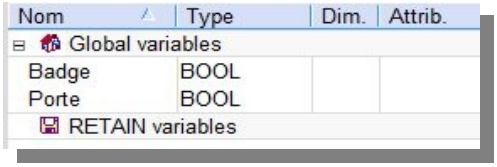
Cliquez sur cette nouvelle variable; maintenez la touche « Alt » de votre ordinateur enfoncée et appuyez sur la touche « Entrée ». Renseignez les différents champs proposés.



Voici donc la déclaration du badge :

- emplacement : Global Variables (a)
- type : Bool → E/S Tor
- Profil : P400XiChannel (liaison avec le P400Xi) (b)
- output : non coché car c'est une variable lue (c)
- number : 1 (le numéro de la voie dans le P400Xi)

On réalise la même opération avec le contact de porte (voie 11 dans le P400Xi).

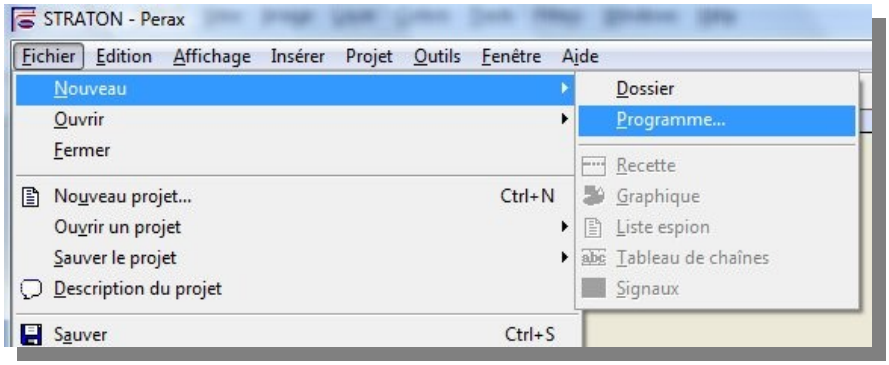


Nous disposons maintenant, pour nos 3 programmes, du contact de porte et de la lecture du badge d'entrée.

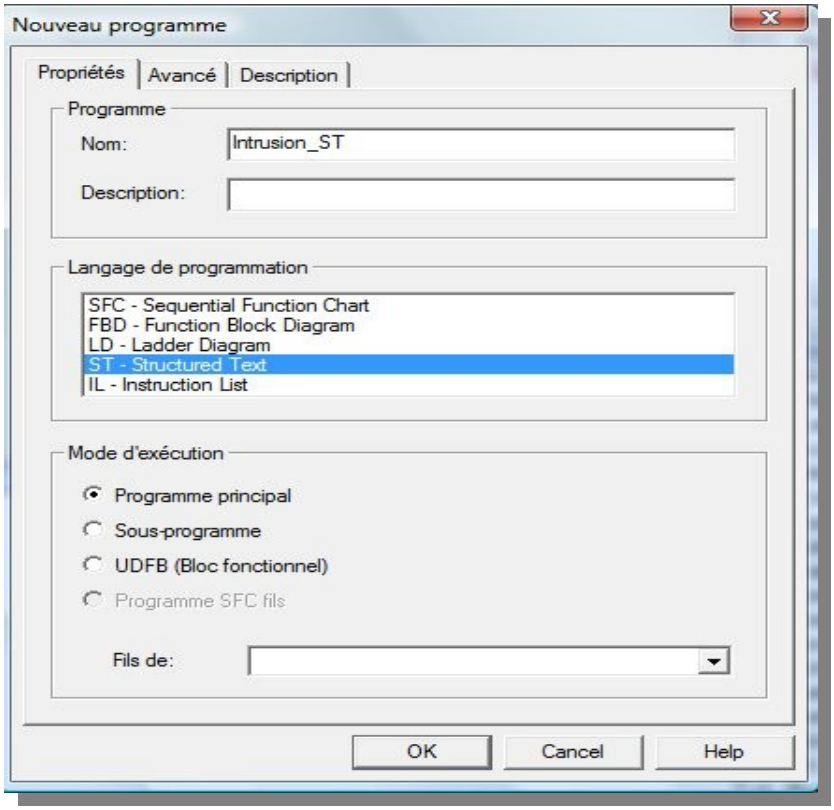
Si vous voulez ajouter des informations à ces variables (valeurs initiales, commentaires, ...) faîtes un double clic sur le champ souhaité, saisissez votre texte et VALIDEZ EN APPUYANT SUR LA TOUCHE ENTREE DE L'ORDINATEUR.

3.2 Le langage ST

Dans le menu «Fichier », nous allons déclarer un nouveau programme ...

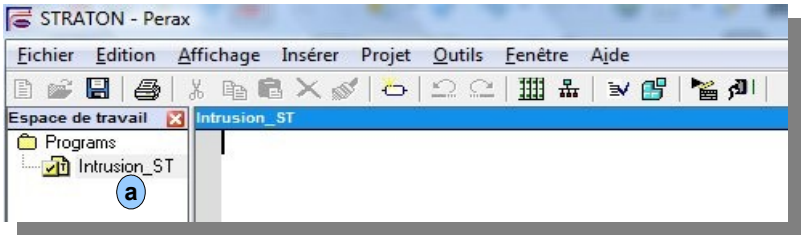


… le nommer et choisir le type de langage.

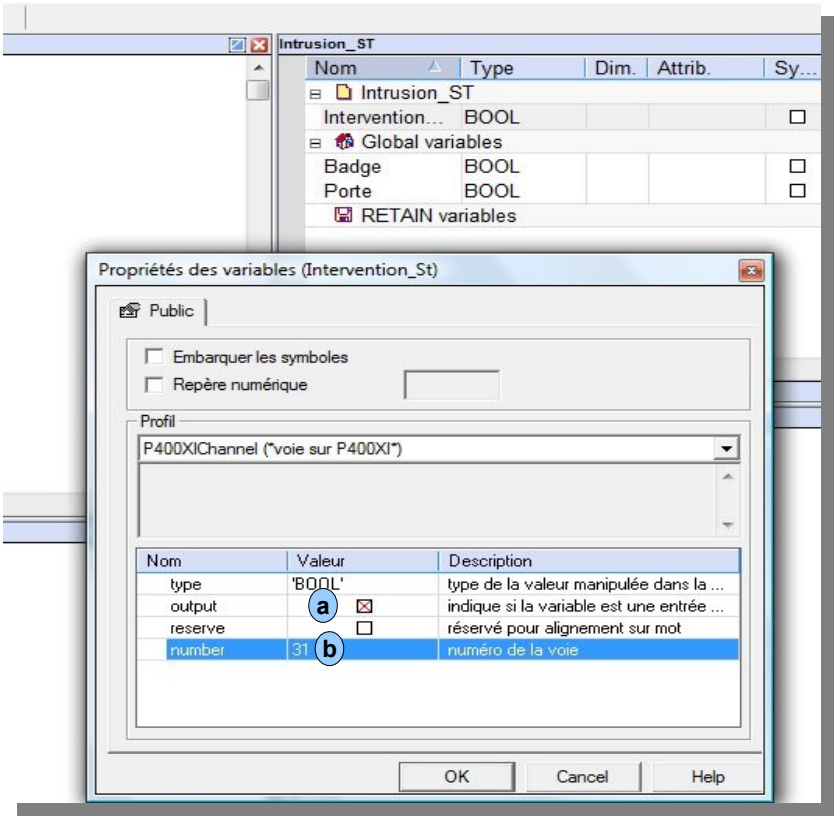


Je valide mon choix en cliquant sur le bouton « OK » .

Il ne me reste plus qu'à double cliquer sur le nom de mon programme (a) pour le mettre en mode édition.

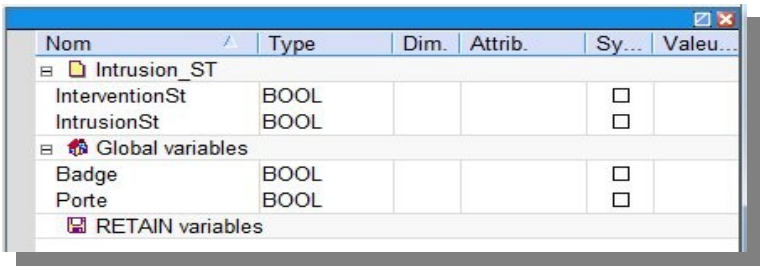


Commençons pas la déclaration des variables Intrusion et Intervention.
Dans la partie liée au programme ST, ajoutons ces 2 variables.



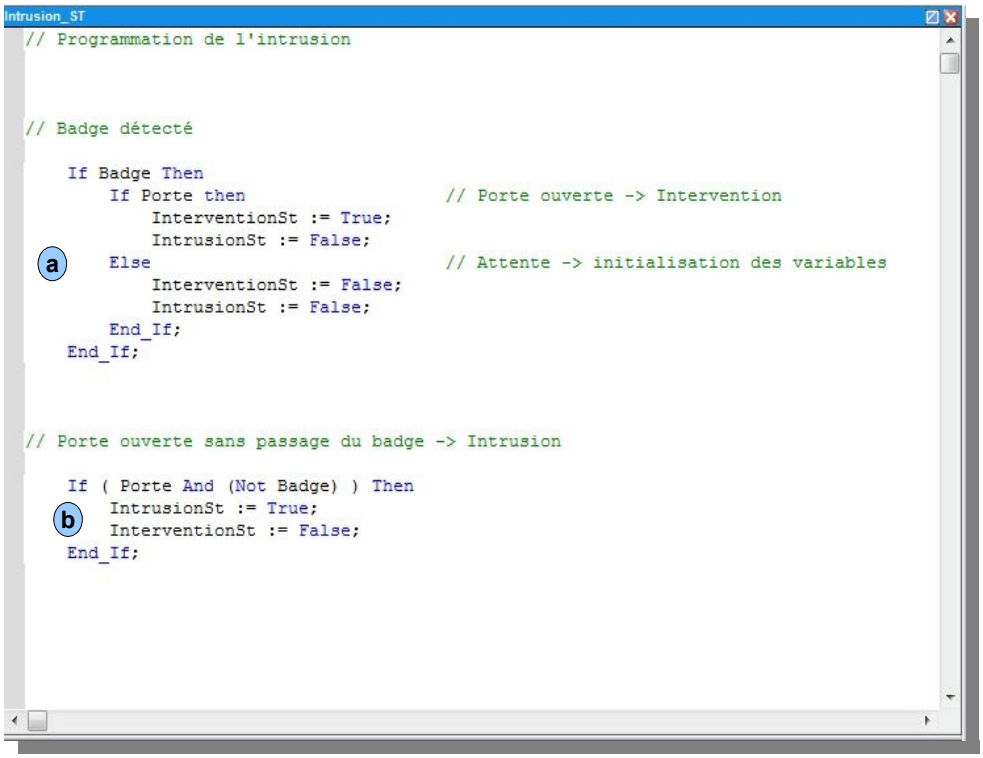
Ce sont des variables de profil P400XIChannel avec la case output cochée (écriture dans le P400Xi) (a) ; on spécifie toujours le numéro de voie (b) .

Voici nos 2 variables :



Voilà; nous avons là tout ce dont il nous faut pour programmer cette intrusion.

Détail du programme :



Les mots – clés apparaissent en bleu.

Les lignes de commentaires commencent par « // » et sont affichées en vert.

Si le badge est détecté et que la porte s'ouvre, nous sommes en intervention; si la porte ne s'ouvre pas, nous repassons en attente. (a)

Si la porte est ouverte et que le badge n'a pas été actionné, nous sommes en intrusion. (b)

Syntaxe :

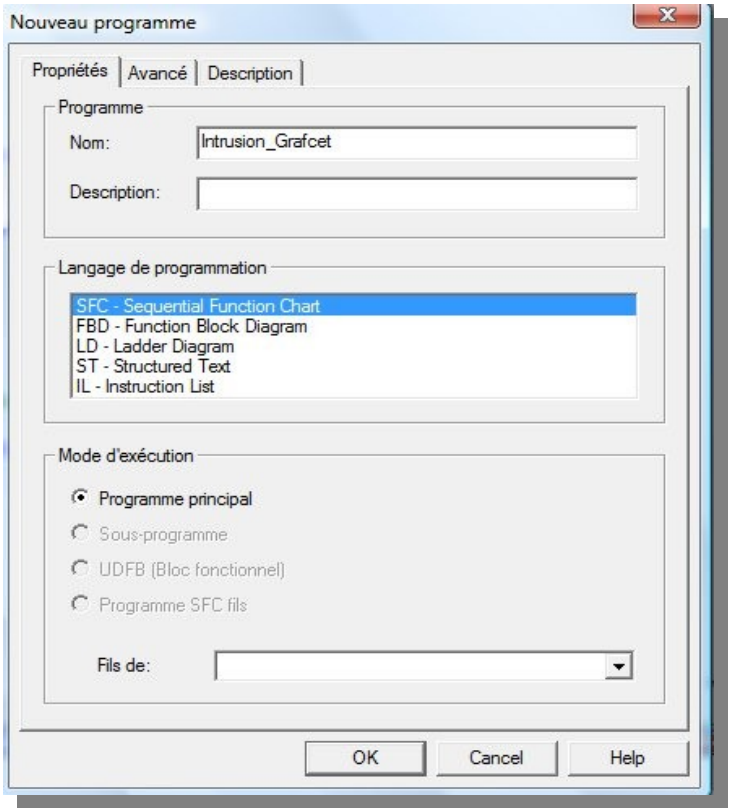
Pour cette programmation, nous n'avons utilisé que des conditions ***If*** (si) ... ***Then*** (alors) … ***Else*** (sinon).

And correspond à et; ***Not*** est utilisé pour inverser l'état d'une variable .

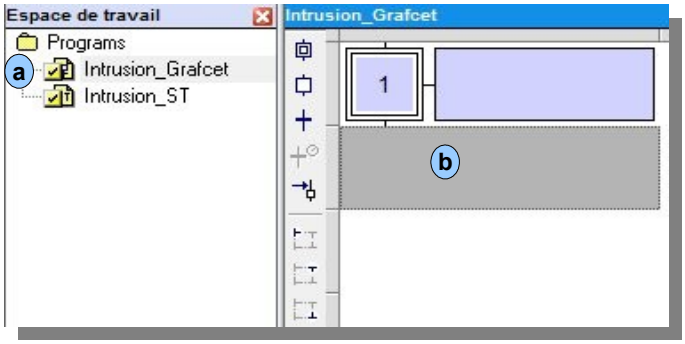
True et ***False*** correspondent à vrai et faux.

3.3 Le Grafcet

Comme pour le langage ST, nous allons créer un nouveau programme en choisissant SFC comme type de langage.



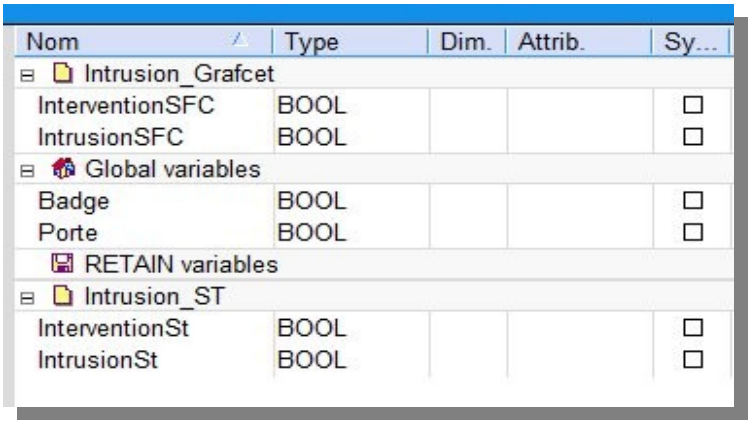
Un double-click sur le nom du programme (a) nous permet de passer en mode édition (b).



Nous pouvons remarquer que l'atelier Straton accepte différents types de langage dans le même projet.

Nous allons maintenant rajouter les variables dont nous avons besoin; à savoir, IntrusionSFC et InterventionSFC.
Le déclaration est à l'identique de la déclaration des variables pour le langage ST.

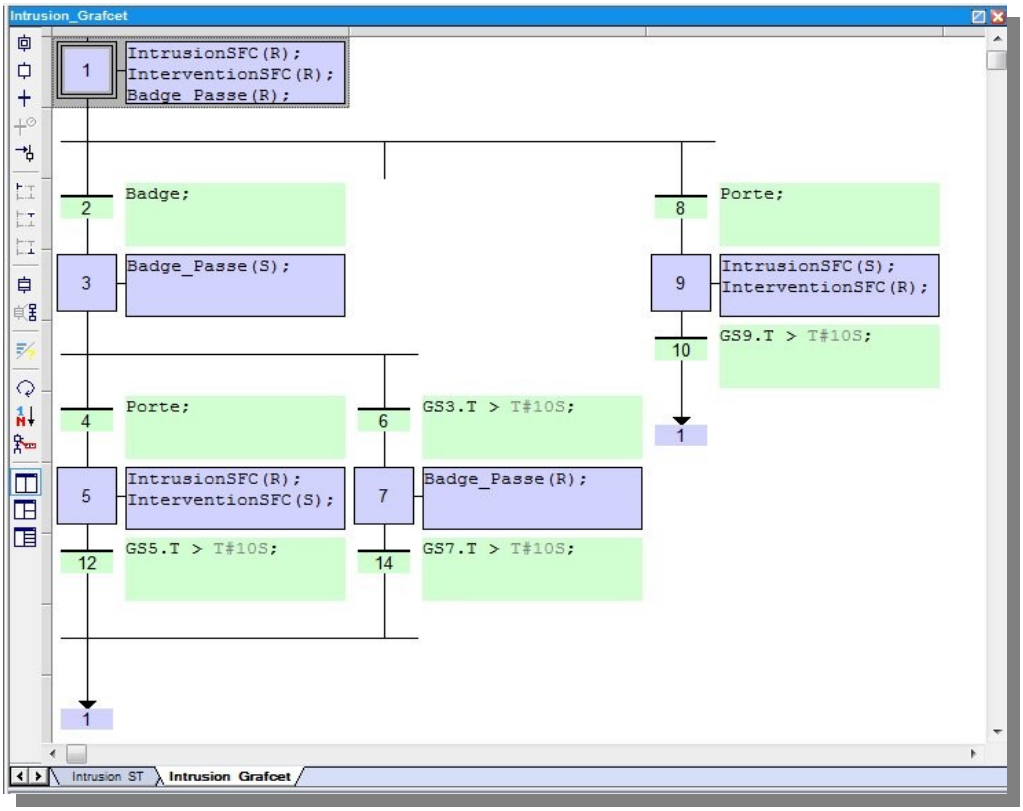
Nous voici donc avec les variables déclarées.



Nom	Type	Dim.	Attrib.	Sy...
⊟ Intrusion_Grafcet				
InterventionSFC	BOOL			☐
IntrusionSFC	BOOL			☐
⊟ Global variables				
Badge	BOOL			☐
Porte	BOOL			☐
RETAIN variables				
⊟ Intrusion_ST				
InterventionSt	BOOL			☐
IntrusionSt	BOOL			☐

Il est temps de réaliser le programme.

Détail du programme :



L'étape 1 (étape de départ) prend en compte l'initialisation des variables; on voit là une nouvelle variable (Badge_Passe) que l'on définit en variable locale, de type BOOL.

Ensuite les transitions 2 & 8 scrutent si on passe un badge ou si on ouvre la porte (avant d'avoir passé le badge).

L'étape 9 provoque immédiatement une intrusion (porte ouverte et pas de badge); nous nous retrouvons ensuite en attente sur l'étape 1.

Si un badge est détecté, transition 2 → étape 3, le programme attend l'ouverture de la porte (transition 4) pour détecter une intervention (étape 5).
Par contre, si la porte ne s'ouvre pas, après 10 secondes d'attente (transition 6), on en déduit que l'exploitant est parti (étape 7) et on se remet en attente sur l'étape 1.

Voici donc réalisée en Grafcet la programmation de cette intrusion; une programmation Grafcet est une suite d'étapes (fond mauve) et de transitions (fond vert)

Syntaxe :

GSa.T > T#bS veut dire que l'on attend b secondes après le passage de l'étape a.

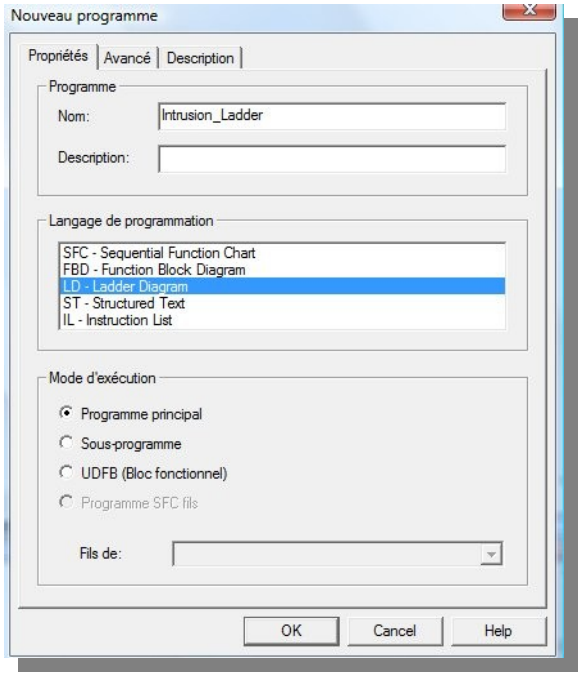
Variable***(R)*** = mise à faux de la variable; Variable***(S)*** = mise à vrai de la variable.

Badge; → on attend que le Badge soit actif (de la même manière pour la porte).

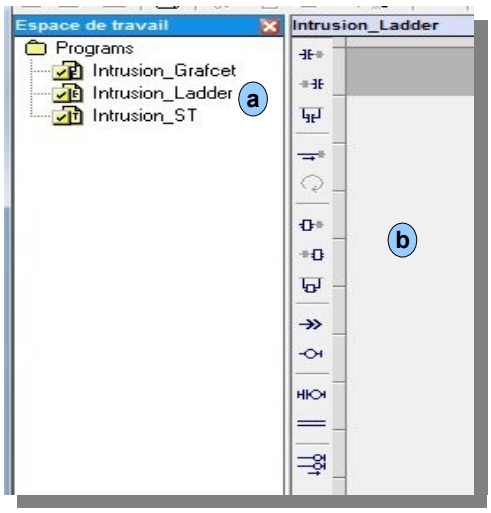
Remarque : il nous faut créer une variable locale au programme Grafcet « Badge_Passe » de type BOOL.

3.4 Le Ladder

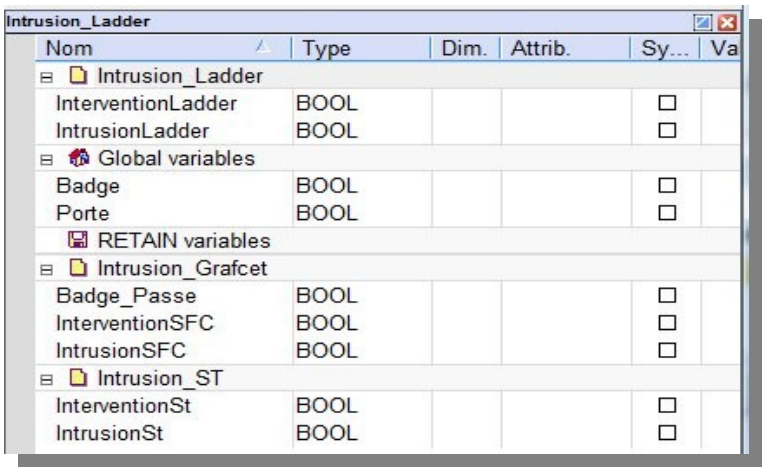
Déclarons un nouveau programme en Ladder (menu « Fichier » → « Nouveau » → « Programme »).



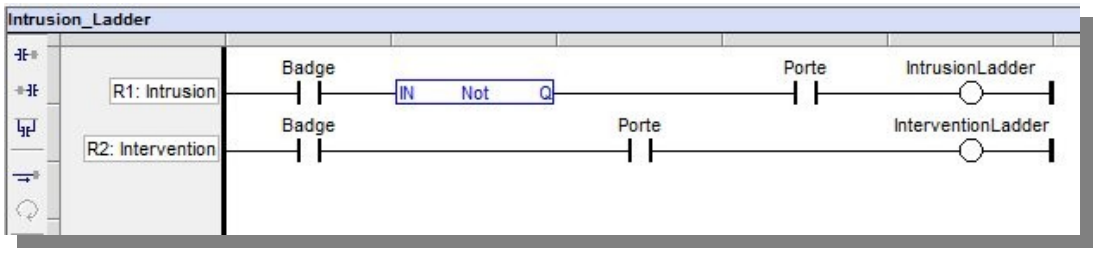
Double-cliquons sur son nom (a), et nous voici en mode édition (b).



Déclarons nos 2 variables (InterventionLadder et IntrusionLadder) …



… et programmons l'intrusion.



La ligne R1 décrit l'intrusion : l'absence de badge (contact badge inversé par le bloc fonctionnel « Not ») et le contact de porte provoque l'activation de l'intrusion.

La ligne R2, elle, nous signale l'intervention : le badge et le contact de porte.

Syntaxe :

Nous ne nous servons que de l'inverseur ***Not*** pour l'intrusion.

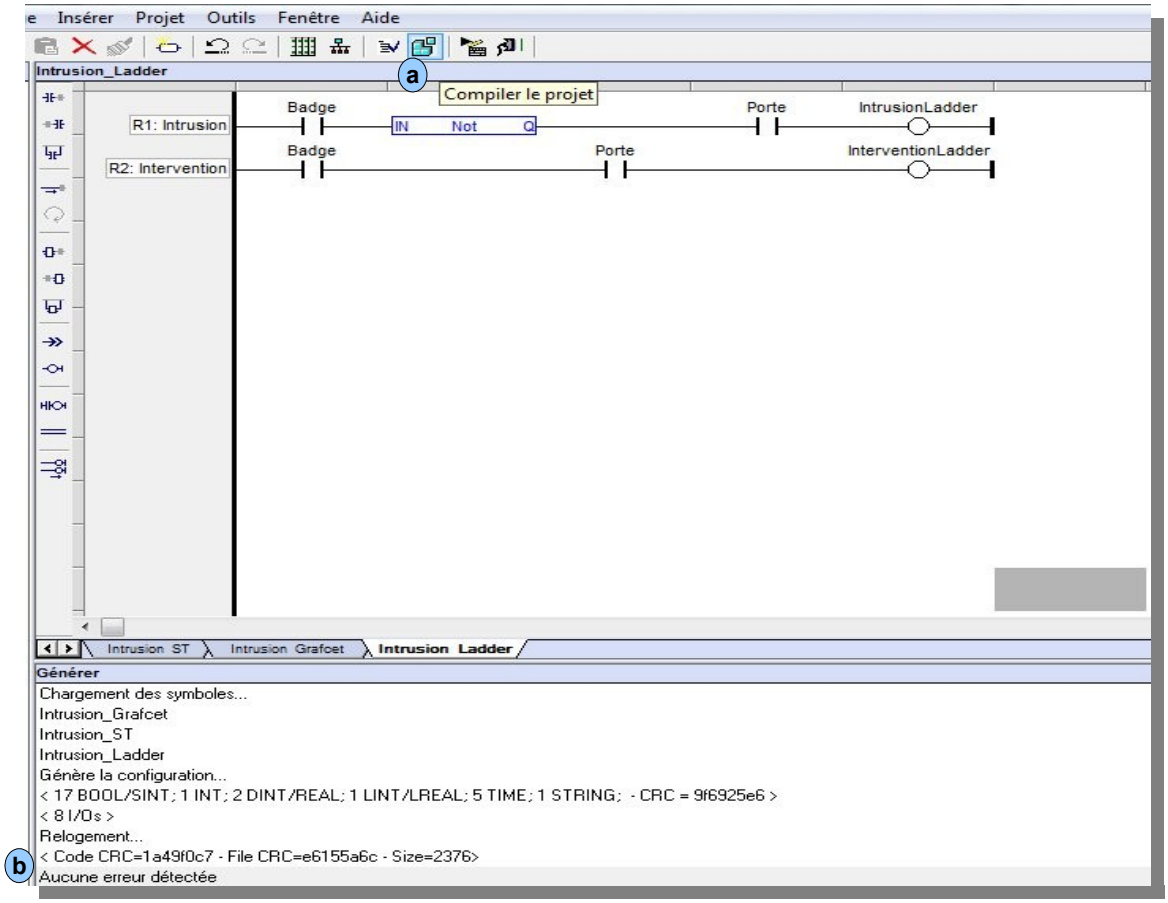
Badge et ***Porte*** passent actifs quand les contacts respectifs se font.

4 MISE AU POINT ET EXPORTATION DANS LE P400XI

Votre programmation est maintenant terminée, il est temps de la valider.
Soit vous l'injectez dans le P400Xi pour la tester en situation réelle, soit vous utilisez le simulateur intégré dans l'atelier Straton.

Mais avant de tester réellement votre programmation, vous devez contrôler que vous n'avez pas fait d'erreur de syntaxe dans le programme.

Cliquez sur le bouton « Compiler le projet » (a) pour valider votre programmation. (b)



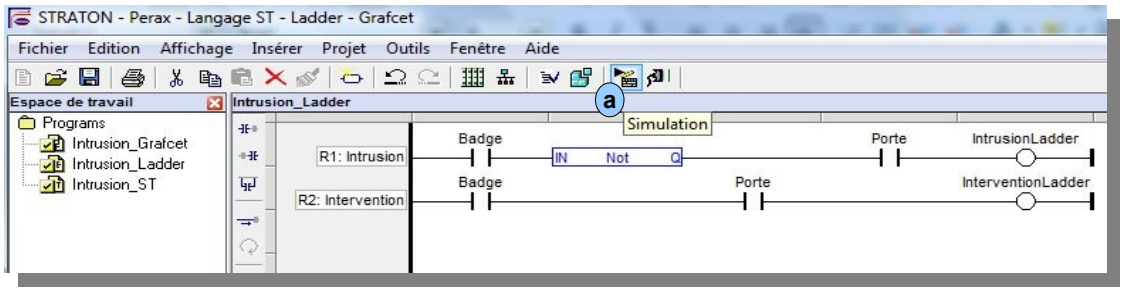
Maintenant que les (éventuelles) erreurs sont corrigées, il est temps de passer au mode exécution.

Note : la compilation s'effectue sur tous les programmes déclarés et non pas seulement sur le programme visible.

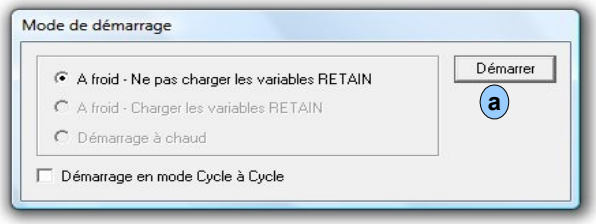
4.1 Le simulateur Straton

Il permet de changer les états de toutes les variables déclarées, en suivant, en temps réel, le comportement de votre programmation.

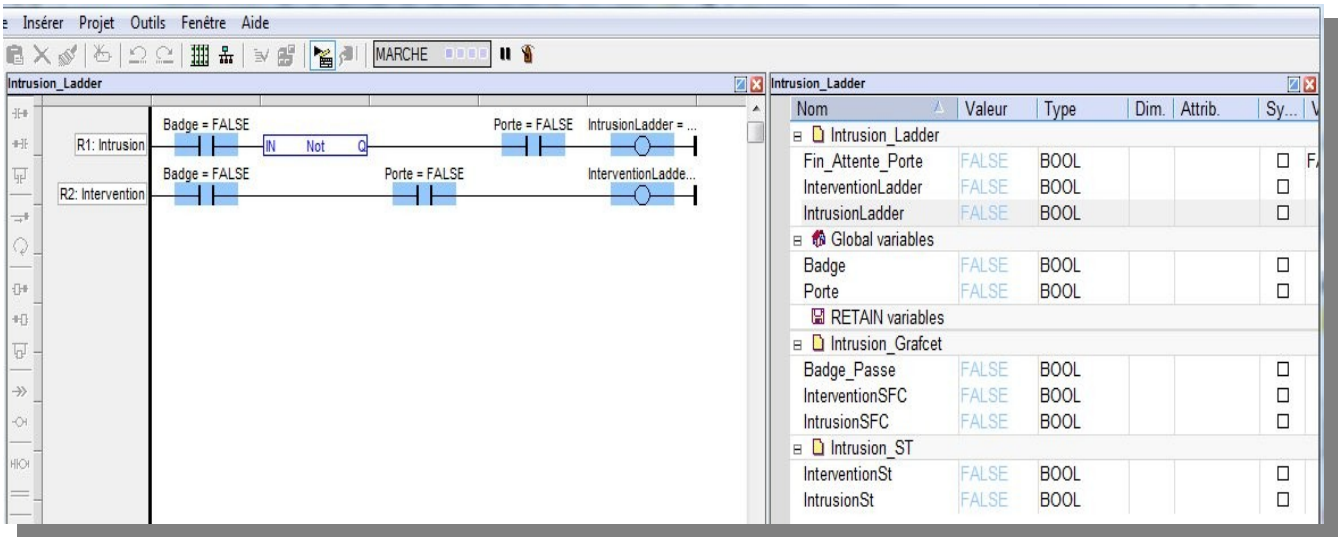
Cliquez sur le bouton « Simulation » (a) .



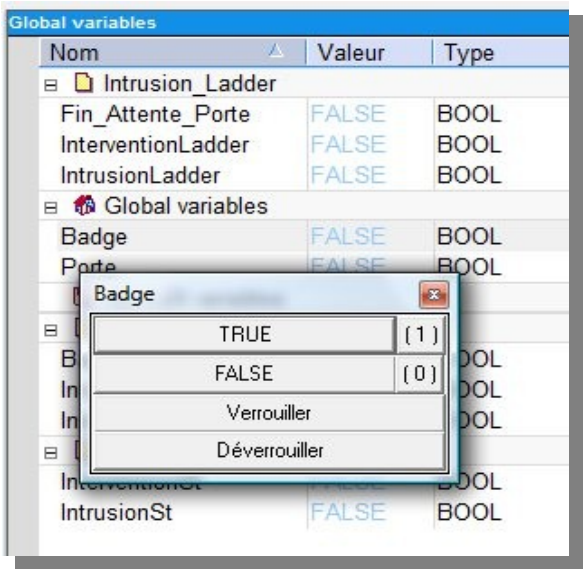
Cliquez sur « Démarrer » (a) .



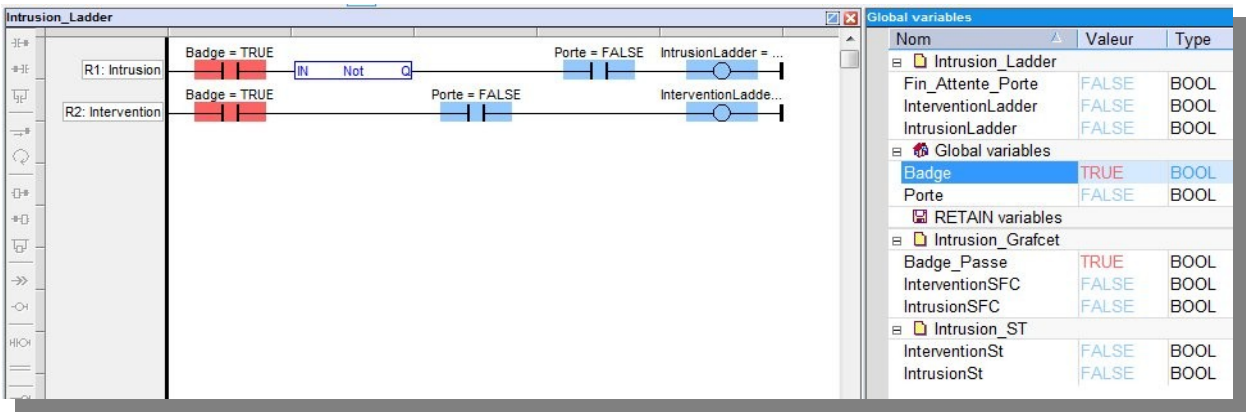
Et vous voici en mode simulation.



En faisant un double click sur une variable, on peut la faire changer d'état ...



… et contrôler immédiatement l'évolution de votre programme.



Si vous cliquez sur les différents onglets de la fenêtre de programmation, vous pouvez contrôler les états de vos différents programmes (dans notre cas, l'évolution des programmes réalisés en Langage St, en Grafcet et en Ladder).

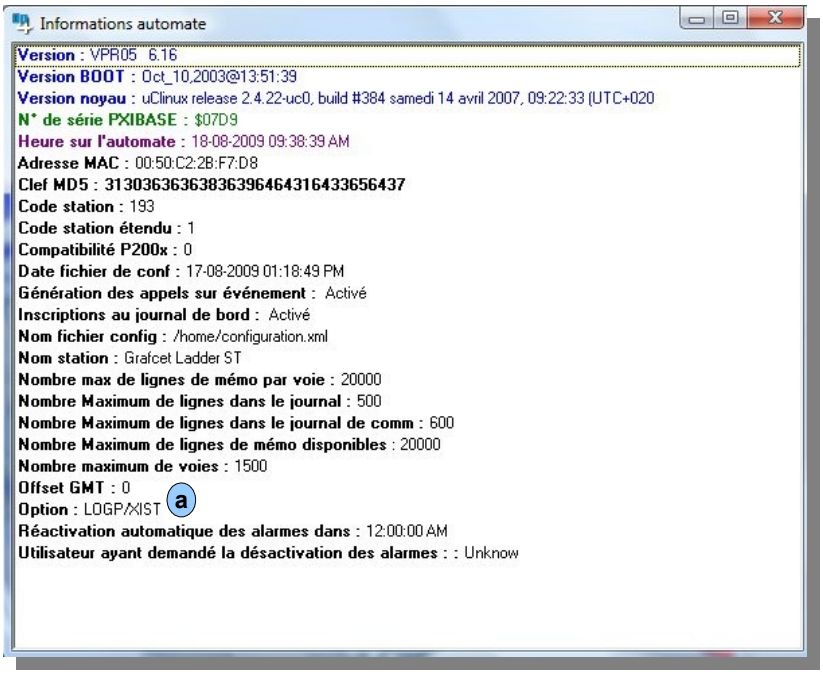
Il ne vous reste plus qu'à simuler tous les cas qui peuvent se présenter durant l'exploitation future du P400Xi.

Pour terminer le mode « Simulation », cliquez de nouveau sur son bouton.

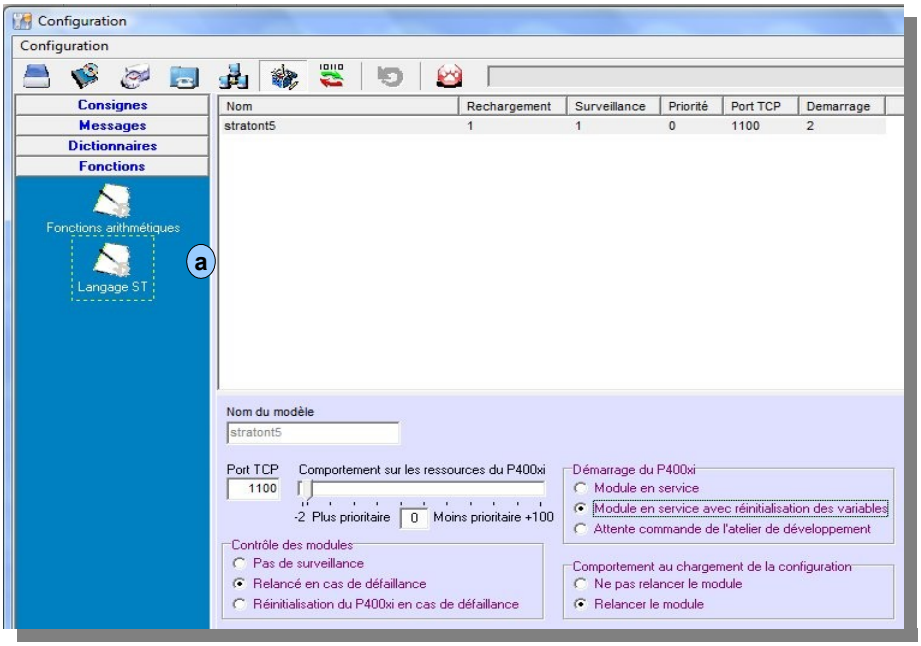
4.2 EXPORTATION DANS LE P400XI

En standard, le P400Xi n'intègre pas le lien avec l'atelier Straton. Il vous faut acquérir cette fonctionnalité et effectuer une mise à jour du produit.

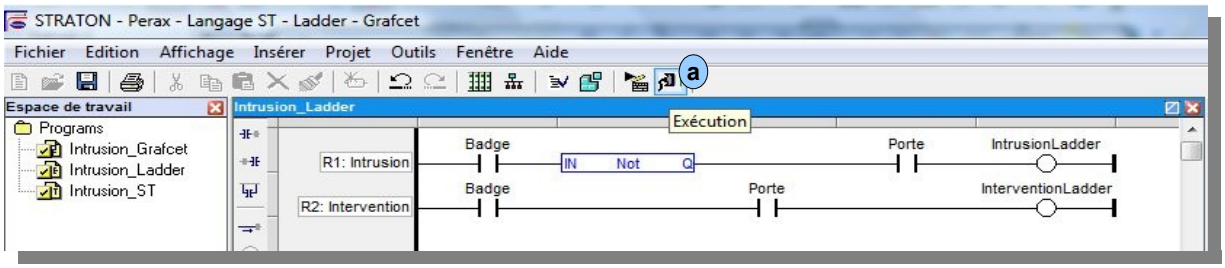
Contrôlez les « Informations automate » du P400Xi; vous devez avoir l'option « LOGP/XIST » chargée (a) .



Après avoir ajouté le « Langage ST » dans la configuration du P400Xi (a) , celui-ci est maintenant prêt à gérer les programmations Ladder, Grafcet et ST que vous allez lui télécharger.



A partir de l'atelier Straton, cliquez sur le bouton « Exécution »(a).



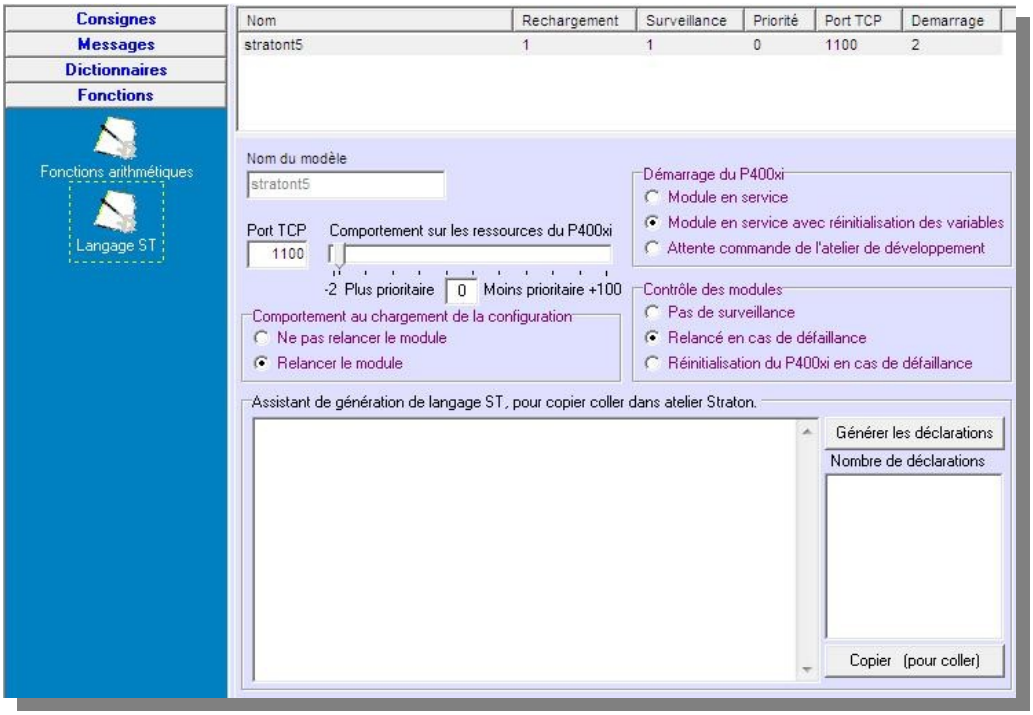
Le programme est téléchargé dans le P400Xi.

Dans ce mode « Exécution », vous contrôlez le déroulement de la programmation gérée par le P400Xi.

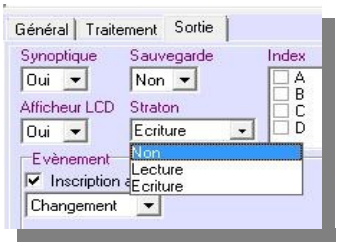
En re - cliquant sur ce bouton, vous quittez le mode « Exécution » (en repassant en mode configuration) mais votre programmation est opérationnelle dans le P400Xi.

5 EXPORTATIONS DES VARIABLES D'ARLEQUIN_P400XI VERS L'ATELIER STRATON

Dans le logiciel Arlequin_P400Xi, lorsque vous déclarez l'atelier Straton …



… un nouveau champ apparaît dans l'onglet « Sortie » de la programmation de chaque type de voie.

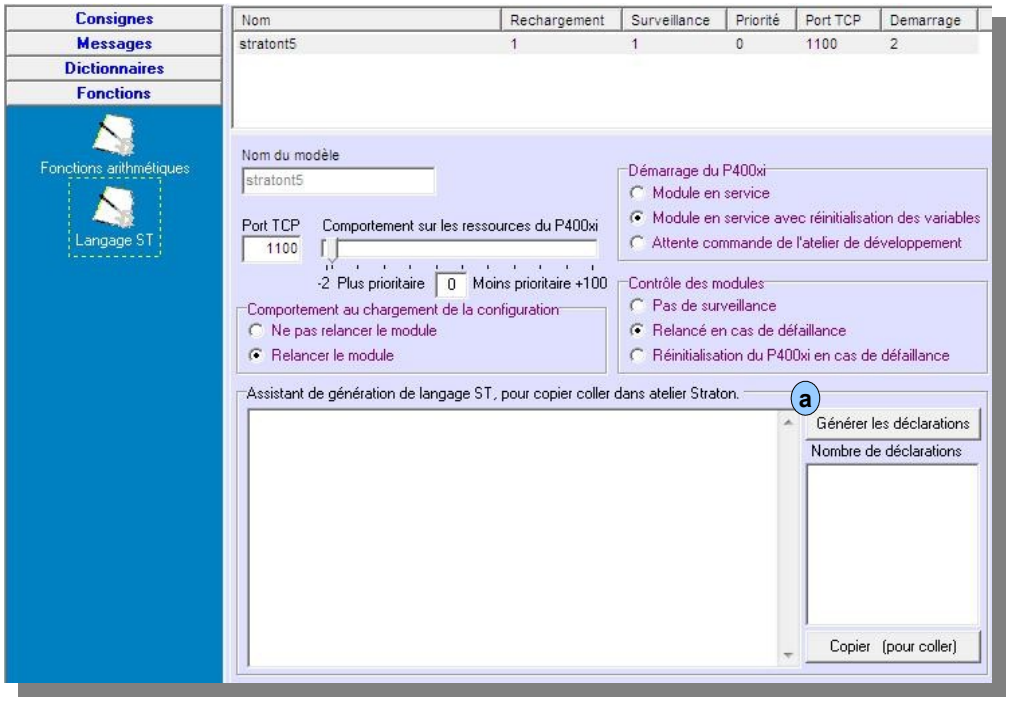


Ce champ, « Straton », permet de définir les voies qui sont concernées par votre programmation Straton (aussi bien en lecture qu'en écriture).

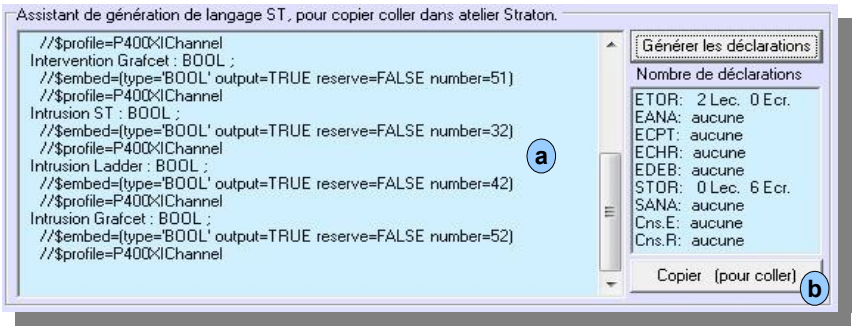
Pour chaque voie concernée, faîtes votre choix.

5.1 Copier les voies

Retournez dans la déclaration de l'atelier Straton et cliquez sur le bouton « Générer les déclarations » (a) .

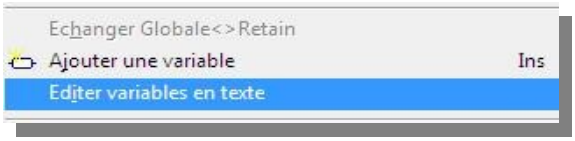


Les variables générées apparaissent (a) ; il ne vous reste plus qu'à les « Copier » (b) .

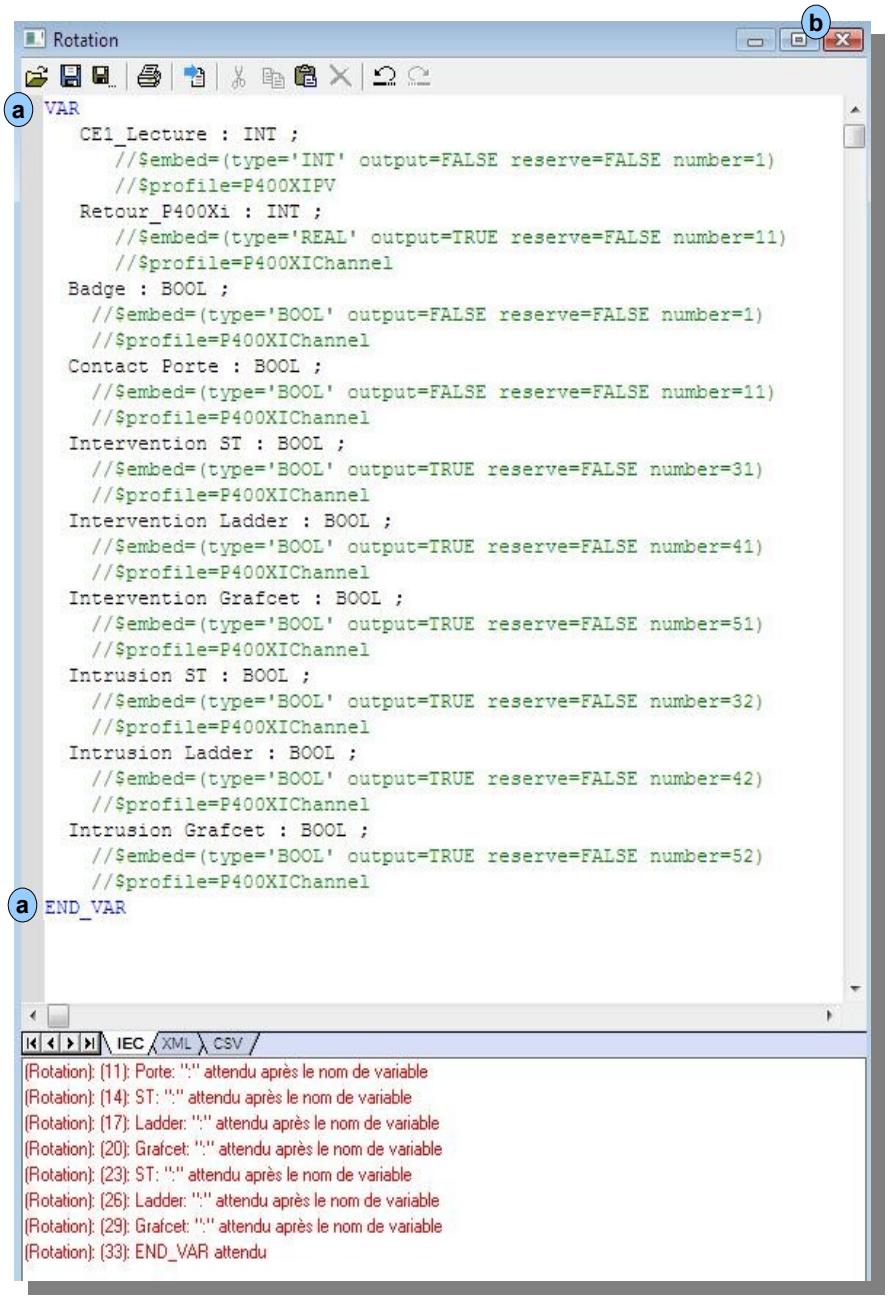


5.2 Coller les variables

Dans l'atelier de programmation Straton, dans la zone de déclaration des variables, faîtes un click droit là ou vous souhaitez insérer les voies préalablement copiées …. et choisissez « Editer variables en texte ».



Une fenêtre s'ouvre, contenant les variables déjà déclarées; collez (Edition → Coller) les nouvelles variables.



« VAR » et « END_VAR » (a) doivent encadrer toutes les variables. Corrigez la syntaxe des variables collées (pas d'espace, d'accents, …); l'atelier Straton n'accepte pas ces caractères.

Fermez cette fenêtre (b) pour valider votre programmation. Ces variables sont maintenant disponibles.

6 ANNEXES

6.1 Configurations minimales demandées

Matériel : un P400Xi avec une version logicielle 6.14 comportant l'option Langage ST

Logiciel : le logiciel Arlequin_P400Xi de version 1.3.9.10 ou ultérieure

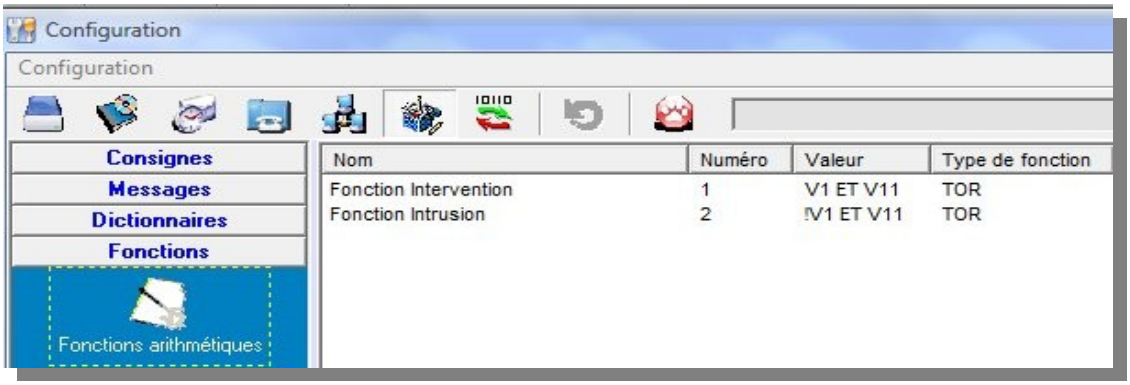
6.2 Nombre de variables faisant référence au P400Xi

Bien que le logiciel embarqué P400XI n'ait aucune limitation, c'est l'atelier STRATON que nous proposons en standard qui est limité à la gestion de 256 entrées/sorties (lecture & écriture de voies / lecture de consignes) parce que nous pensons que c'est suffisant et qu'au delà le surcoût serait inutile.

Il existe bien sûr des versions de l'atelier pour 512 variables et des versions illimitées que nous pouvons fournir à la demande.

6.3 Fonctions « Perax » intégrées au P400Xi

Si on désire effectuer la même programmation avec les fonctions intégrées au P400Xi, voici ce que cela donne.



Rappel : V1 est la voie « Lecture du badge » / V11 est la voie « Porte ouverte » .

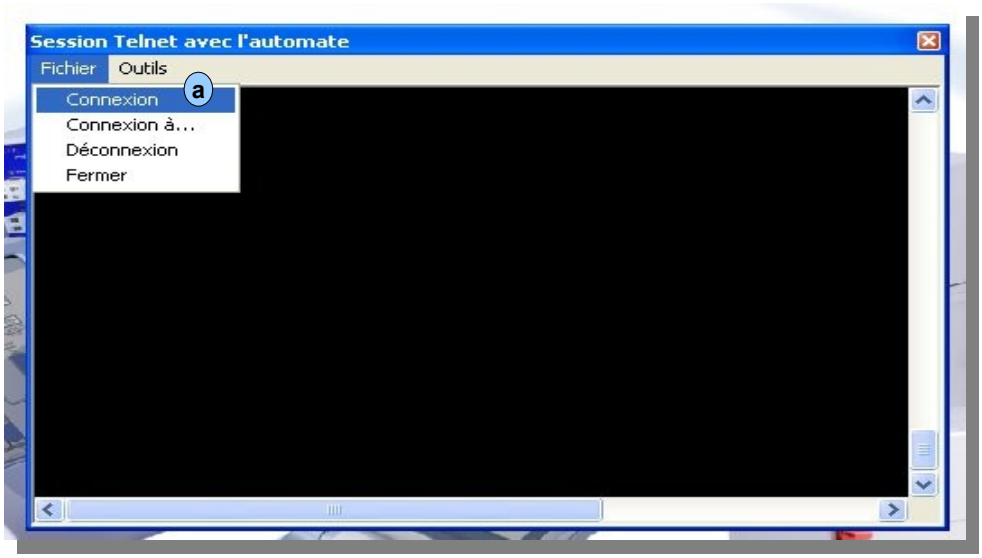
6.4 Problème de liaison Straton <-> P400Xi

Lorsque vous exportez votre programme dans le P400Xi, l'atelier Straton peut vous signaler un problème de communication avec l'automate. Cela arrive si, par exemple, votre programmation comporte des instructions qui provoquent une boucle sans fin.

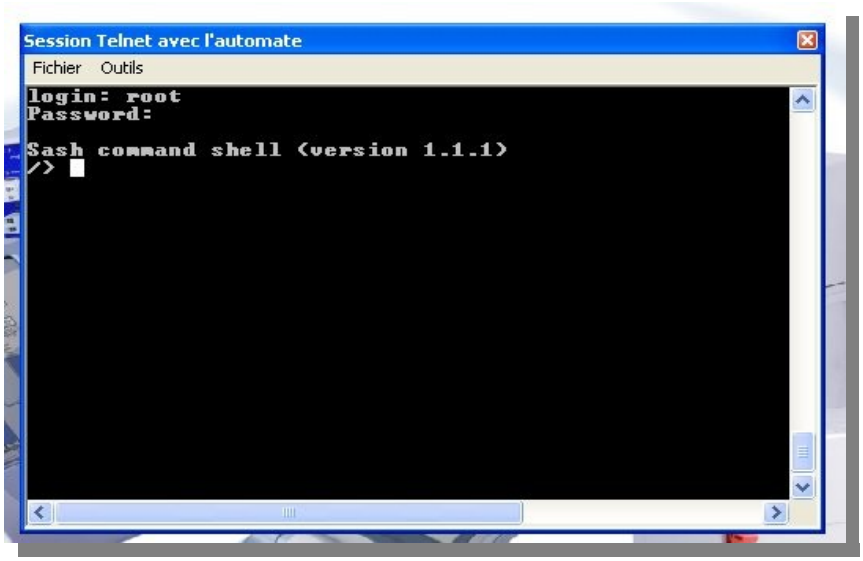
Pour remédier à cela, il vous faut d'abord corriger votre programme, et ensuite supprimer manuellement la programmation Langage ST dans le P400Xi.

Pour supprimer la programmation ST dans le P400Xi (à partir d'Arlequin_P400Xi), assurez-vous d'être en connexion avec l'automate et cliquez dans « Outils » -> « Terminal distant » . (a)

Une fenêtre 'texte' apparaît. En maintenant la touche 'Shift' enfoncée, cliquez sur « Fichier » -> « Connexion » .... (a)



... jusqu'à voir apparaître la fenêtre suivante.



Relâchez la touche « Shift ».
Au niveau du curseur, tapez :

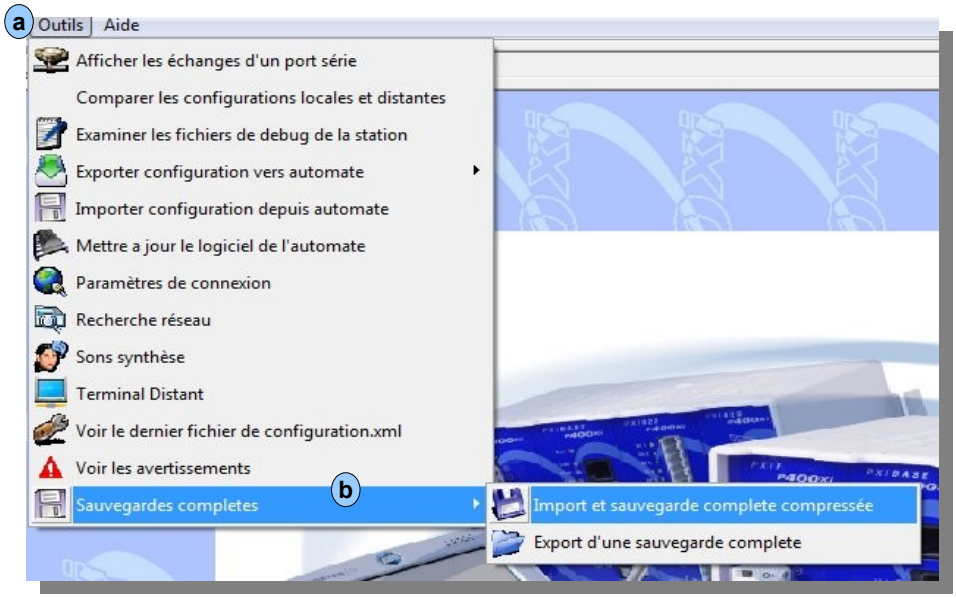
- rm /var/parameters/t5* *(caractère espace entre rm et /var)*
- (validez par la touche « Entrée »)
- kill -SIGPWR 46 *(caractère espace entre kill et -SIG)*
- (validez par la touche « Entrée »)

Le P400Xi se relance en supprimant toute sa programmation ST .

6.5 SAUVEGARDE DE VOTRE PROGRAMMATION

Lorsque vous importez la configuration d'un P400Xi, vous ne récupérez pas la programmation Straton.
Pour cela, il vous faut faire une sauvegarde complète de votre P400Xi.

Dans le logiciel Arlequin_P400Xi, faîtes « Outils » (a) , « Sauvegardes complètes » (b).



Choisissez « Import ... » pour récupérer la programmation; « Export ... » pour injecter une sauvegarde complète préalablement enregistrée dans un autre P400Xi.