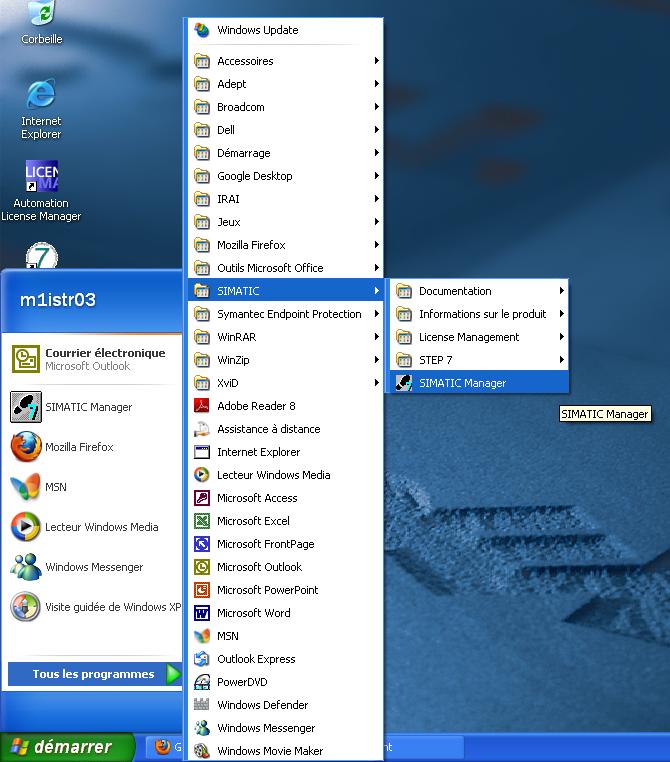
Création et implémentation d’un programme sur Simatic Manager

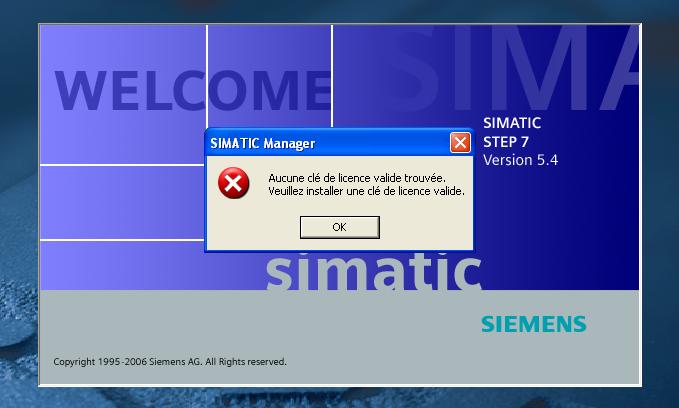
# Etape 1 : Ouvrir Simatic Manager

Commencez par ouvrir Simatic Manager (trouvable sur le bureau ou dans le menu démarrer)



# Etape 2 : Licence

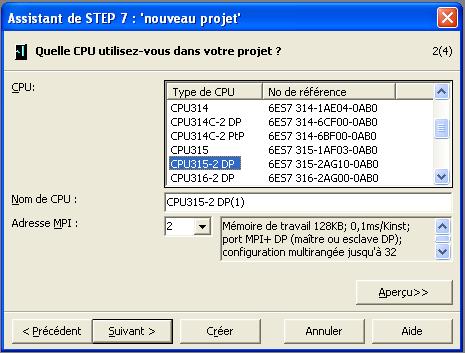
Un message de licence risque d’apparaître, cliquez sur OK dès que vous en verrez un, cela n’altèrera pas les fonctionnalités du logiciel (au cours de l’utilisation du logiciel, ce message peut apparaître plusieurs fois, continuez à faire OK).



# Etape 3 : Création du projet

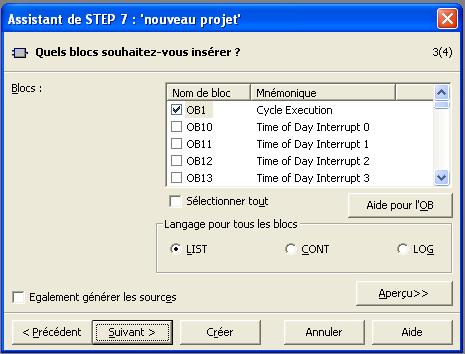
Le logiciel va directement vous demander de créer un projet.



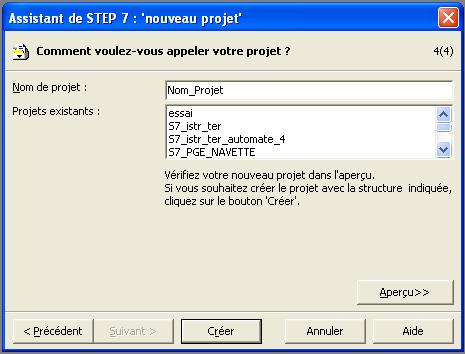
Faites suivant et sur la page 2/4 choisissez le processeur CPU315-2 DP, il correspond au processeur présent dans chacun des automates Siemens présents sur la maquette. 

Faites suivant.

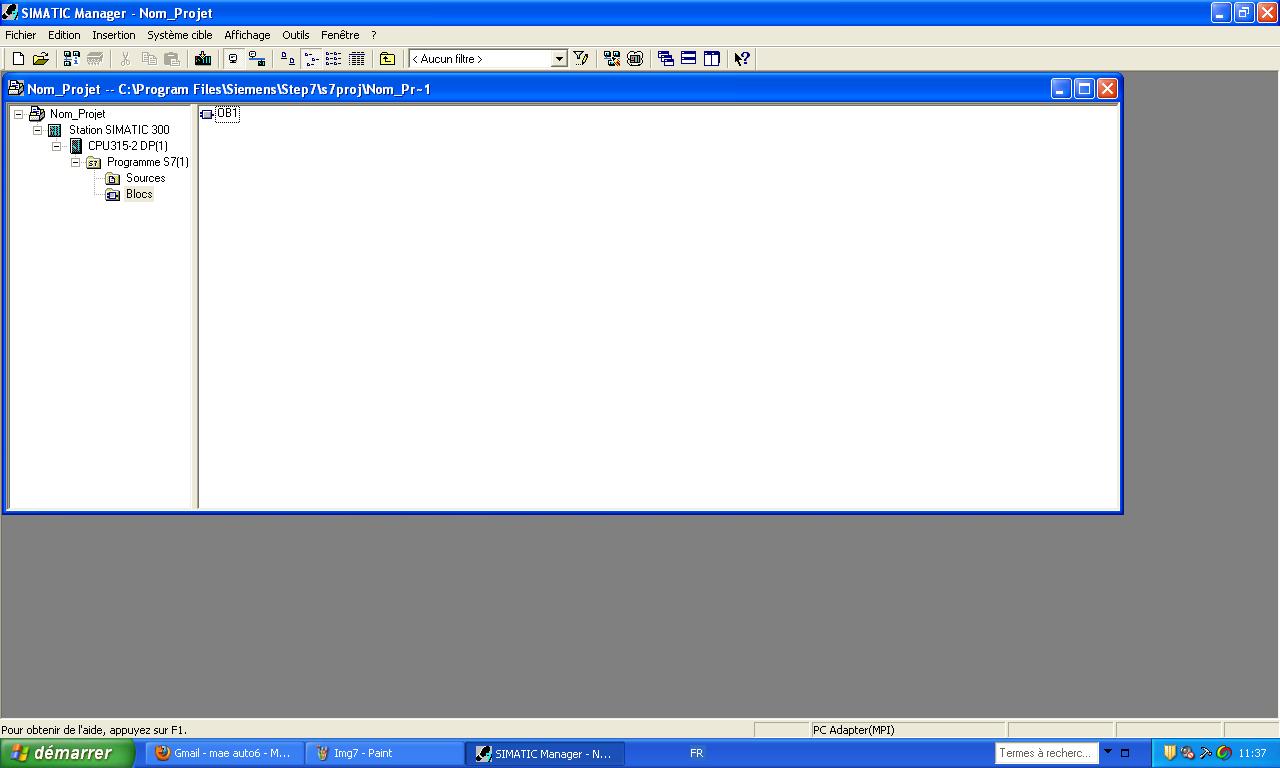
Et sur cette page sélectionnez le nombre de bloc que vous voulez (OB1 correspond au « main »). Ici nous n’aurons besoin que du bloc 1. Sélectionnez aussi le langage de programmation que vous souhaitez (ST sera celui manipulé dans la suite).



Faites suivant et définissez maintenant le nom de votre projet. Cliquez ensuite sur « créer ».

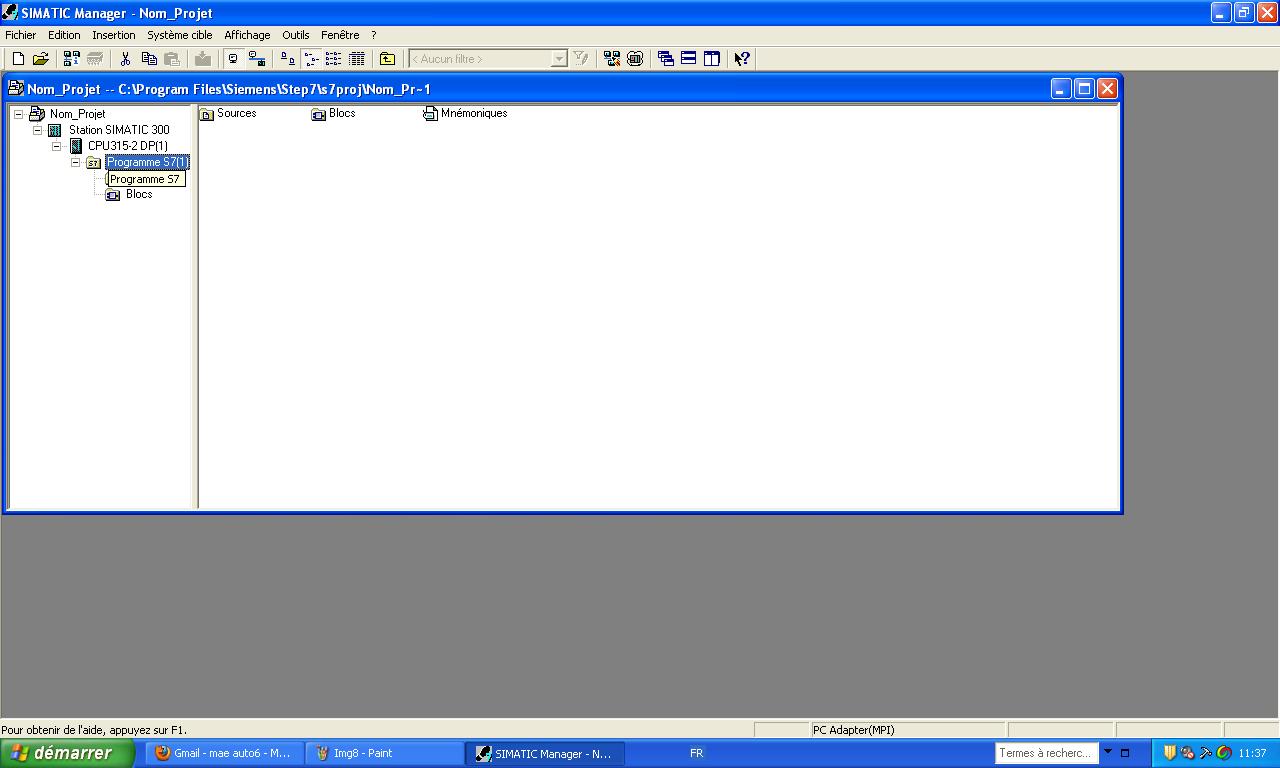


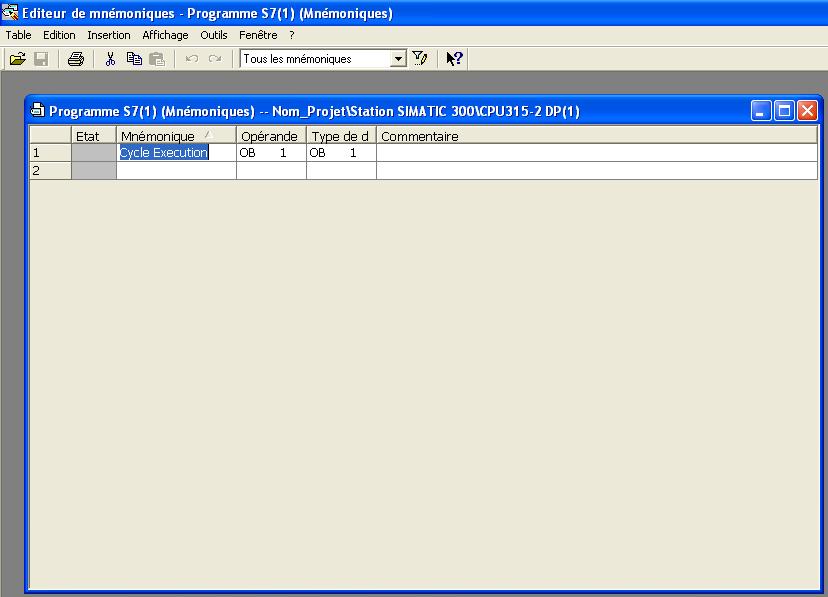
Vous arrivez maintenant sur l’environnement de Simatic Manager.



# Etape 4 : Paramétrage des entrées sorties.

Cliquez sur « Programme S7(1) » dans l’arbre du projet.



Double cliquez maintenant sur Mnémonique. 

Vous voilà dans la table des Mnémonique, c’est ici que vous allez informer le nom de vos entrées et de vos sorties en fonction des cablâges sur les ports d’entrée/sortie.

Comme ceci :



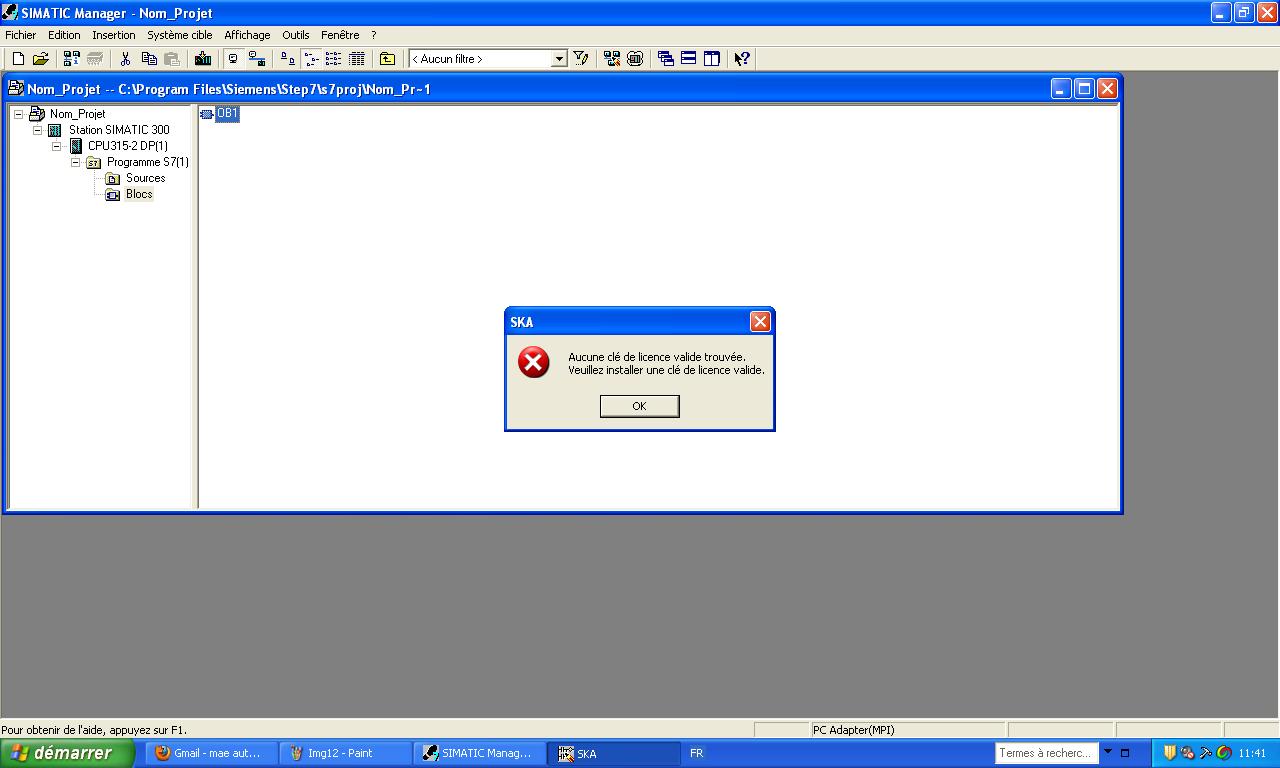
Pour les automates que nous avons, les entrées sont de E4.0 à E5.7 et les sorties vont de E8.0 à E9.7 (2 mots de 8 bits).

Remarques

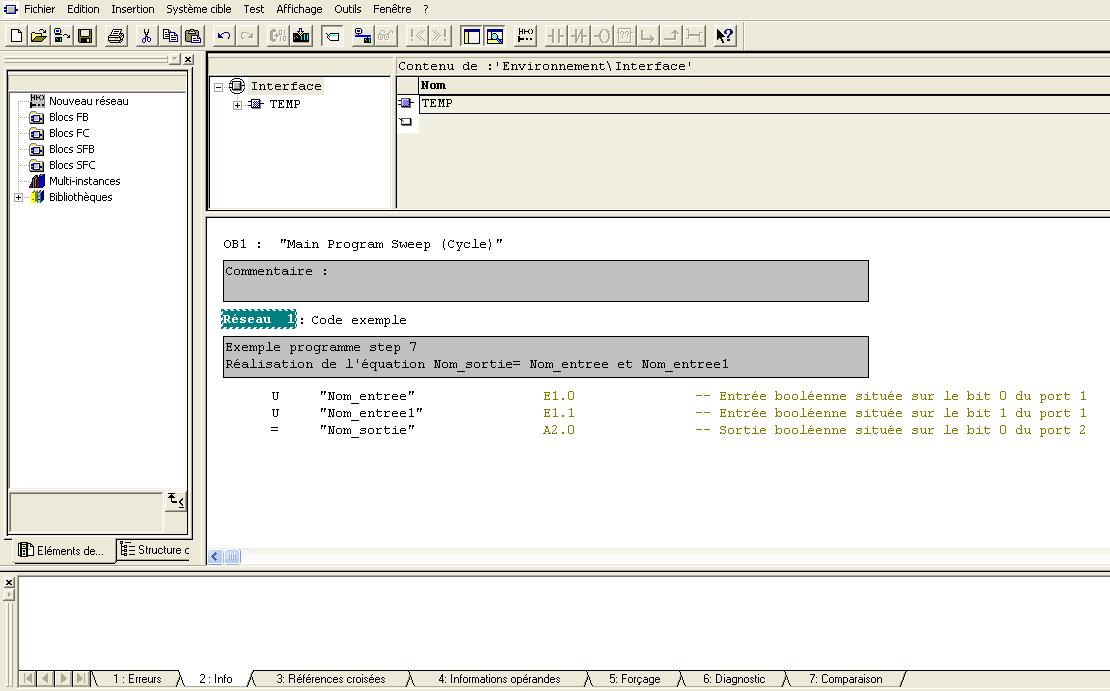
* Les entrées/sorties peuvent être manipulées comme des bits ou des mots de 8/16 ou 32 bits, attention à ne pas lire/écrire n’importe où.
* Après 4.7 vient 5.0 et non 4.8 ou encore 4.9 !

# Etape 5 : Codage en Step 7 (autre possibilité, voir 5.1)

Après avoir enregistré la table des Mnémoniques, revenez sur la fenêtre contenant le projet, cliquez sur « Blocs » dans l’arborescence du projet et double cliquez sur « OB1 » (Si un message de licence s’affiche, continuez à faire « OK »).



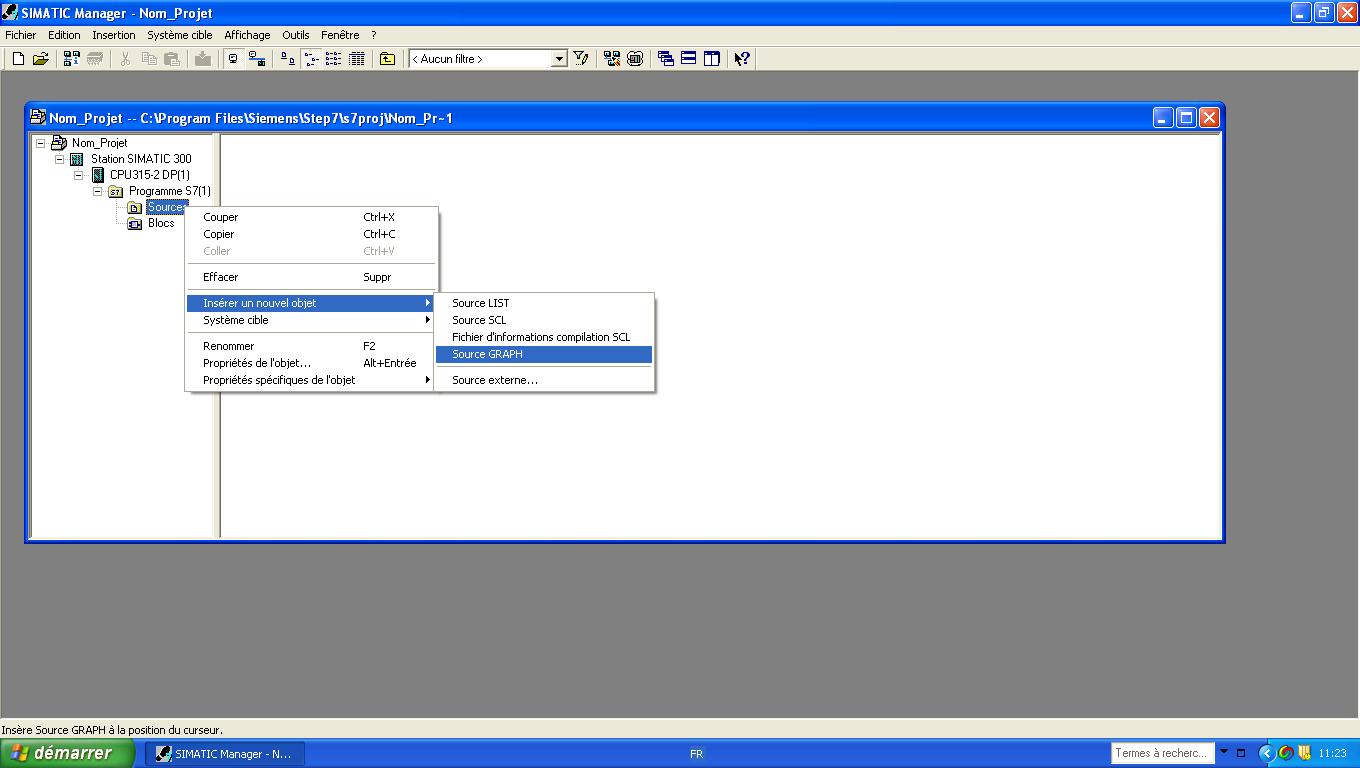
Vous devez arriver sur une fenêtre de ce type (avec les champs vides) :



Cette étape est la plus difficile à réaliser car elle demande des connaissances en programmation en Step7 (cf Annexe 1 pour plus d’informations).

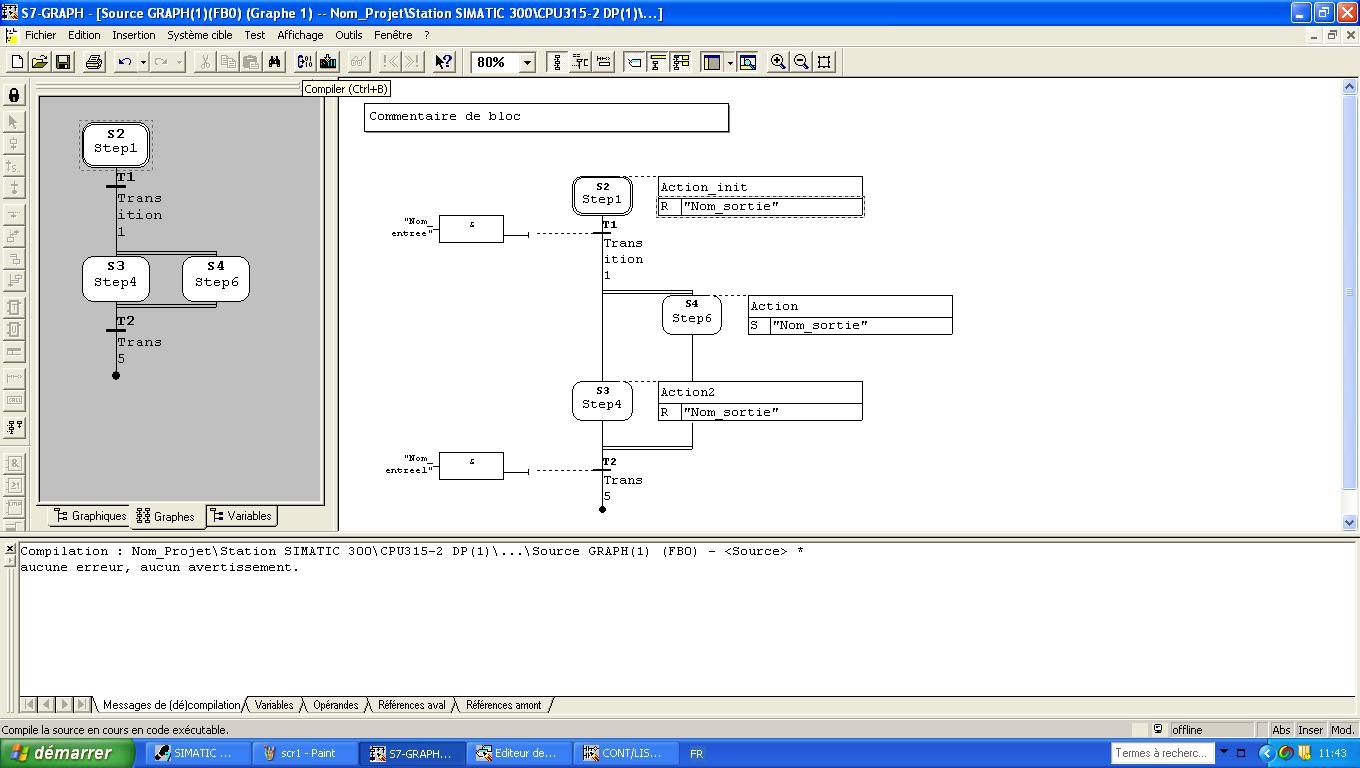
# Etape 5.1 : Implémentation d’un Grafcet

Après avoir configuré les mnémoniques, revenez sur la fenêtre du projet et faites un clic droit sur sources (dans l’arborescence) 🡪 insérer un nouvel objet 🡪 Source Graph.



Un fenêtre d’édition de Grafcet va s’ouvrir. Vous pourrez ajouter des divergences/convergences en ET/OU, des étapes, des transitions, des sauts ; pour cela vous pouvez utiliser la colonne d’icônes sur la gauche de la fenêtre. Pour programmer les transitions, vous devrez faire des équations en Ladder.

Vous pourrez vous aider de l’onglet « variables » en bas pour pouvoir insérer les noms des entrées/sorties/variables internes.



Une fois que vous aurez fini de faire votre graphe, vous devrez le compiler (en appuyant sur le bouton indiqué ci-dessous ou en faisant CTRL+B). Vous devrez peut-être créer un bloc de données.

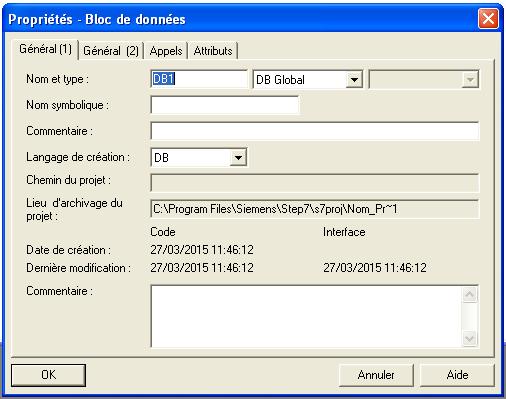


Cette compilation génèrera différents blocs (FBX, DBX…).

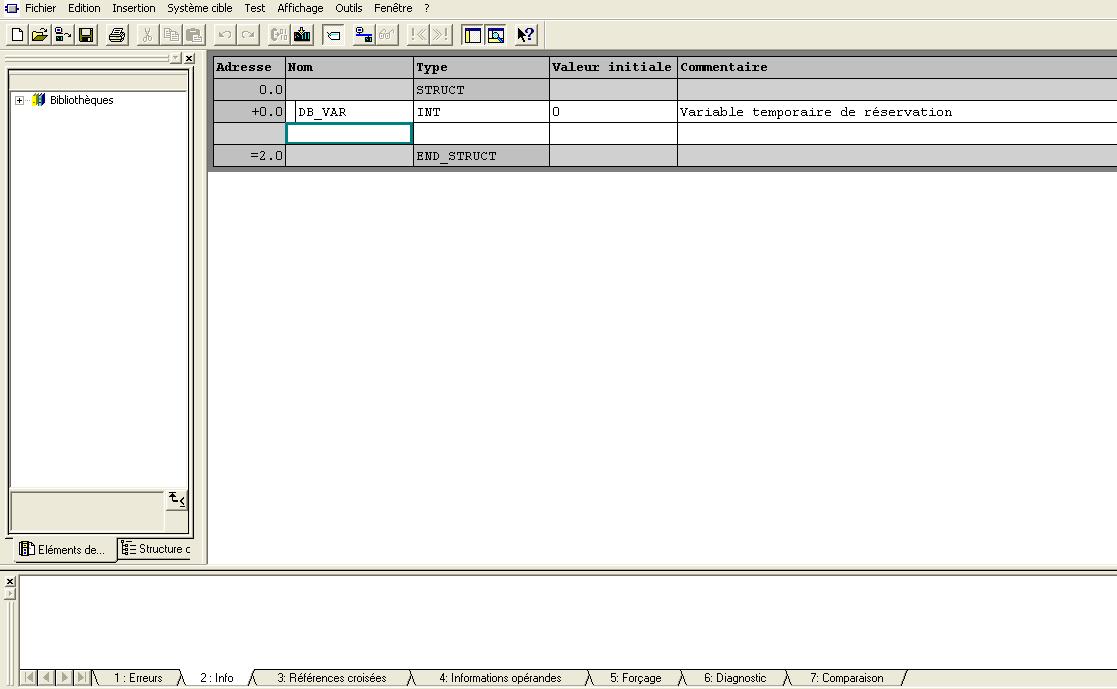
Ajouter la suite ici

# Etape 6 (facultative) : création de variables internes

Pour pouvoir créer et utiliser des variables en interne, retournez sur la fenêtre du projet et dans blocs, faites clic droit🡪insérer un nouveau bloc🡪bloc de données, vous arrivez maintenant sur cette fenêtre :



Faites OK si vous ne savez pas paramétrer le bloc et vous arrivez sur un tableau de ce type :



Pour insérer une nouvelle variable vous pouvez vous déplacer comme dans un tableur ou faites « Insertion ». Entrez son nom, son type et sa valeur initiale.

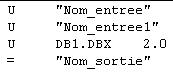
Pour pouvoir ensuite utiliser ces variables dans votre programme, ouvrez le bloc OB1 et déclarez-les comme ceci :

DB(numéro\_du\_bloc\_de\_donnée).DBX (adresse\_de\_debut\_de\_la\_variable)🡪 pour une variable binaire.

DBD pour un entier.

DBW pour un réel.

Exemple :

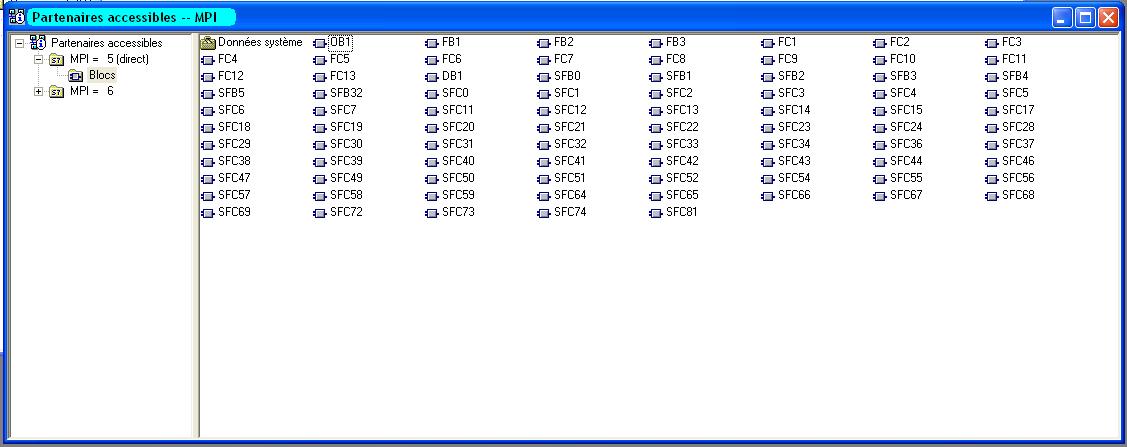


# Etape 7 : Implémentation sur automate

Maintenant que vous avez fait votre programme voulu, il faut le transférer sur l’automate.

Tout d’abord mettez l’automate sur STOP (à faire directement sur l’automate).

Ensuite, dans la fenêtre du projet, faites « Système cible »🡪 « Partenaires accessibles »

Dans la fenêtre qui s’ouvre, se trouve une adresse MPI avec écrit « direct » double cliquez dessus et vous devez atterrir sur ça : 

Tout d’abord, n’ayez pas peur et évitez de supprimer les blocs que vous ne connaissez pas.

Pour programmer l’automate il suffit juste d’y glisser/déposer votre bloc OBX (mettez y aussi votre bloc DBX si vous utilisez des variables internes).

Si vous avez fait l’étape 5.1, mettez-y aussi les autres blocs créés.

Attendez quelques secondes (si vous devez écraser des blocs, assurez-vous que aucun dysfonctionnement soit créé).

Vous pouvez maintenant mettre l’automate sur RUN-P, ce qui exécutera votre programme.

Annexes

[Documentation Simatic](http://cache.automation.siemens.com/dnl/zI/zI2Mzc1AAAA_18652056_HB/S7prv54_f.pdf) : pdf de Siemens sur l’utilisation de Simatic Manager et de ses fonctionnalités.

[Documentation langage STEP7](https://cache.automation.siemens.com/dnl/zg1NDQ0MwAA_18653496_HB/AWL_f.pdf) : pdf de Siemens sur la programmation en STEP7 ST

[Documentation simple STEP7](http://ptrau.free.fr/autom/siemens/details-list.htm) : explication de la base du langage STEP7 ST