

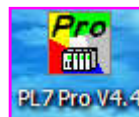
## Initiation au logiciel PL7-PRO

### 1 Introduction

L'objectif de cette partie est de connaître le minimum requis pour l'utilisation du logiciel PL7 vis à vis d'un TSX 37/57. Les langages PL7 sont des langages graphiques (Langage à Relais, Grafcet) et textuels (Structured Text, Instruction List) destinés à programmer les automates TSX. Ils offrent un environnement de programmation complet permettant à l'utilisateur de développer son application dans un langage graphique simple (éditeur), de la traduire automatiquement en langage automate (compilateur) et enfin de la charger dans la mémoire centrale de l'A.P.I. (téléchargement).

Suivant le TSX dont vous disposez (Micro ou Premium), une certaine quantité de mots et de bits sont disponibles (pour le TSX3722 Micro, on a 128 bits d'étapes %Xi, 256 bits internes %Mi, et un nombre de mots limité par la taille mémoire de l'automate.).

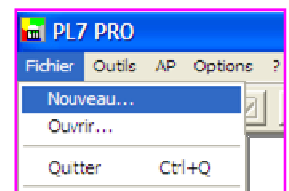
### 2 Ouverture d'une session



Cliquer sur l'icône dans le bureau ou utiliser le menu Démarrer-->Programmes-->Modicon Télémechanique-->PL7 PRO V4.4.

#### 1)- Créer un nouveau projet

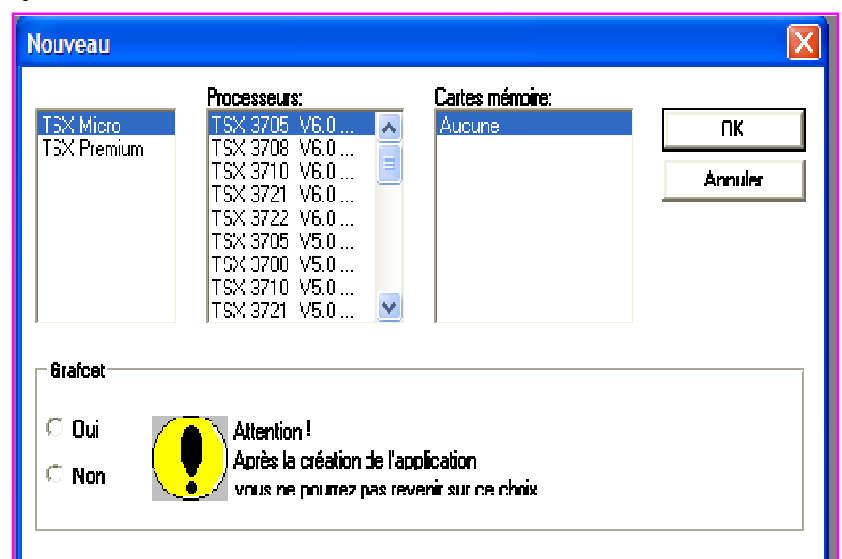
Créer une nouvelle application dans le menu "Fichier", sous menu "Nouveau...".



#### 2)- Sélectionner la version d'automate que vous utilisé

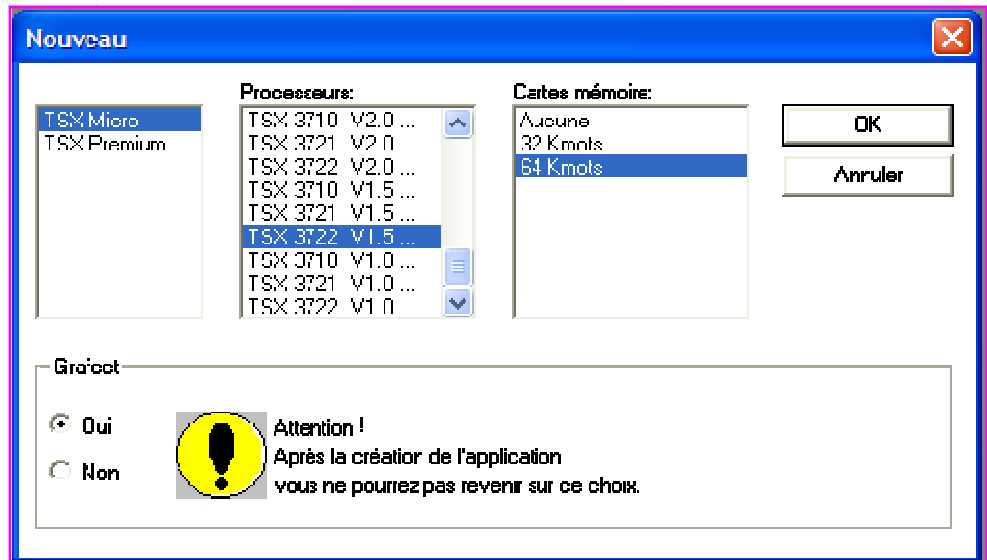
Après création du nouveau projet, on obtient cette fenêtre.

Où on doit choisir en premier le nom de l'automate utilisé soit TSX Premium (pas Langage Grafcet) ou TSX Micro (avec Langage Grafcet), puis en second on choisit le type de processeur et enfin la carte mémoire utilisée (si elle existe) choisir la programmation Grafcet oui ou non selon la solution à proposée



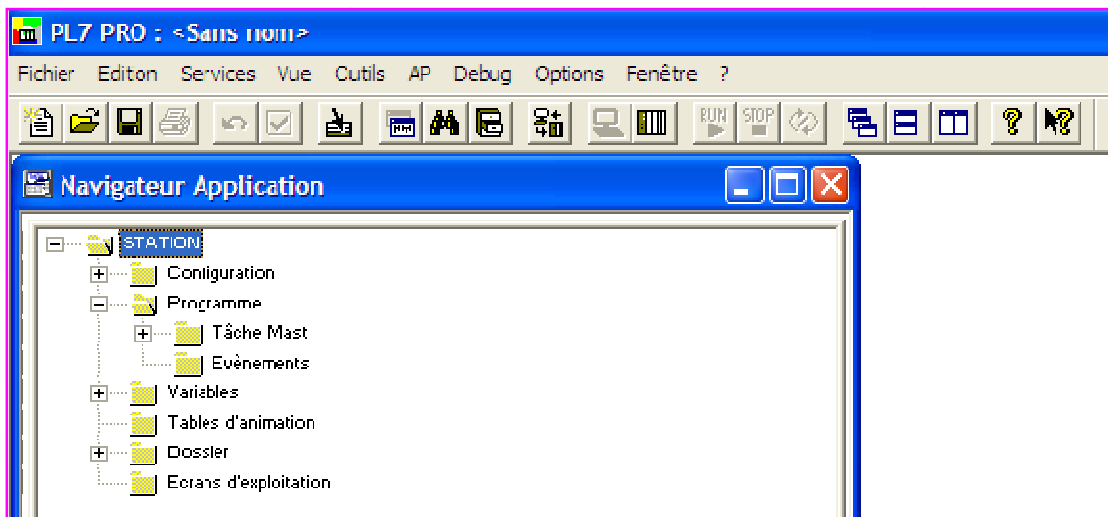
Choisir l'automate utilisé par exemple TSX3722 V3.0 , cartes mémoires « 64 Kmots » et « oui » pour la programmation Grafcet.

Valider votre choix par l'appui sur le



### 3)- Ouverture « Navigateur Application »

L'application est créée et la fenêtre "Navigateur Application" s'ouvre alors.



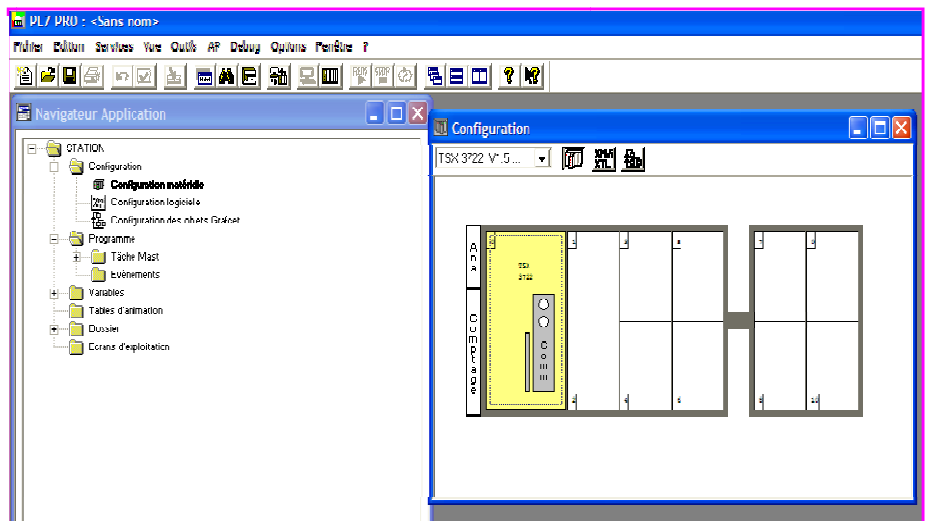
Le Navigateur "Application" présent sur la gauche de l'écran permet d'accéder aux différentes fonctionnalités de l'outil de développement : configuration, programmation, accès aux données, transfert et mise en application du programme, dossier de l'application en cours.

#### 3.1) Configuration matérielle de l'Automate

##### a) Configuration (automatique) du processeur

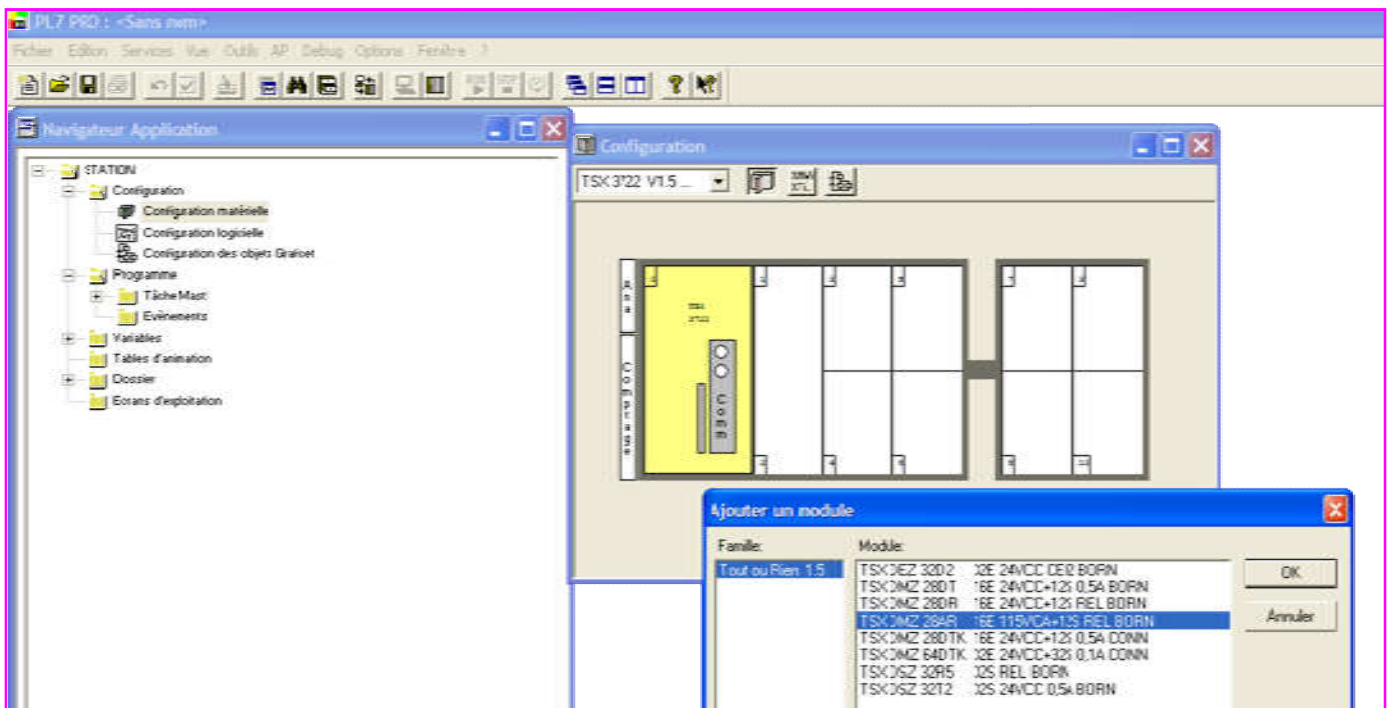
La première étape est de définir sur l'outil de développement la configuration matérielle de l'automate sur lequel le programme sera transféré puis exécuté. Sélectionner dans le dossier "Configuration"--> "Configuration matérielle".

On obtient une image du TSX choisi avec par défaut l'ensemble de ses emplacements vides. Le module de communication "Com" ou module « 0 » est par défaut correctement configuré il correspond au processeur choisit.

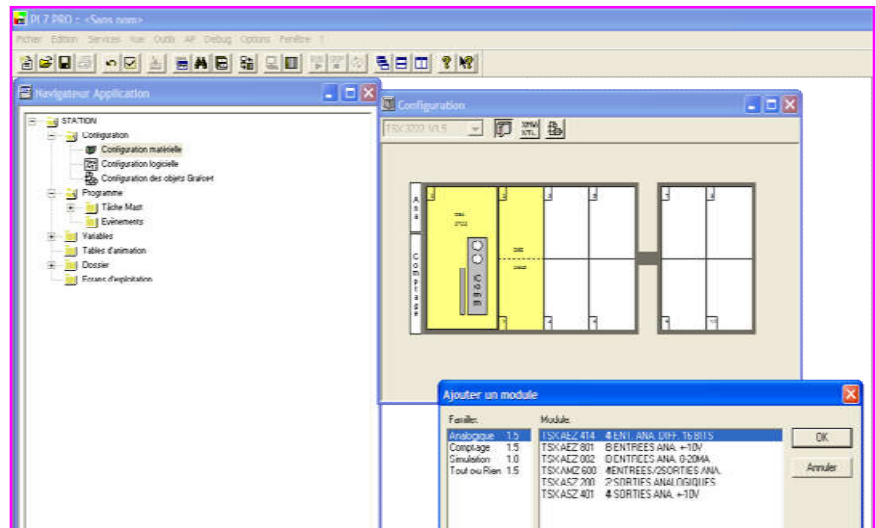


### b) Configuration module entrée / sortie

Pour déclarer les entrées sorties du TSX, cliquer sur l'emplacement 2 et ajouter le module TSXDMZ28DR . Ce module comporte 16 entrées TOR sur la position 1 et 12 sorties sur la position 2 du rack. On valide par OK.



On obtient cette fenêtre, si le nombre variable d'entrée (ou le type d'entrée) ou sortie, on peut ajouter d'autre module on double cliquant sur le module 3 et procédé de la même manière précédente

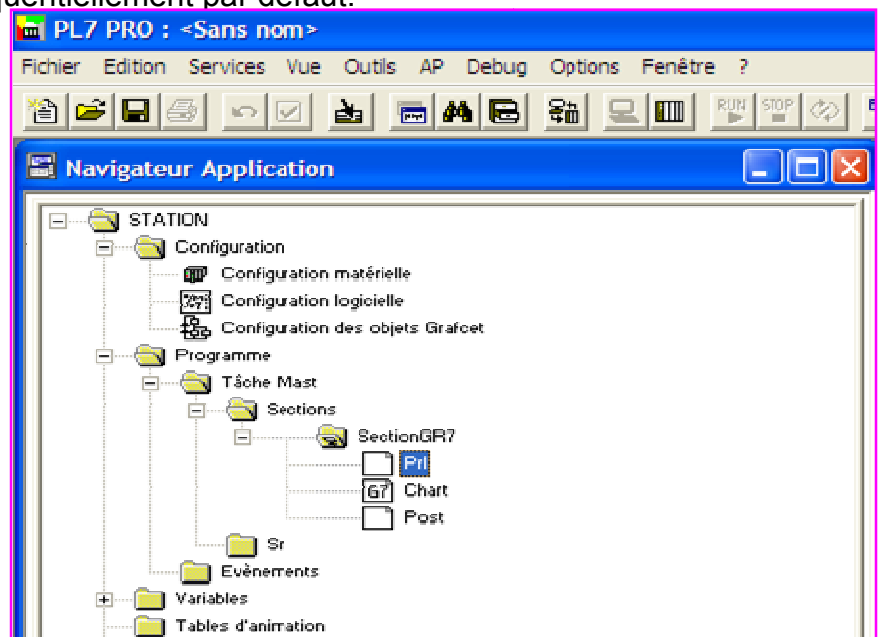


En quittant, valider la "reconfiguration matérielle"

#### 4) Programmation d'une application

Par défaut, la programmation se fait en mode "déconnecté" ou "local". Vous ne communiquez pas avec le TSX. Le navigateur d'application permet d'accéder à tous les modules de programme qui constituent l'application complète. Si vous avez choisi le mode "Grafcet" à la création de l'application, le navigateur montre les éléments de programme prédéfinis : le Grafcet en lui-même (Chart), la partie de programme exécutée avant l'exécution du Grafcet (Pré) et celle exécutée après (Post). Le langage des sections Pre et Post doit être défini par l'utilisateur lors du premier appel de la section. Vous pouvez rajouter une ou plusieurs sections de programme dans le langage de votre choix à votre application. Toutes les sections s'exécutent séquentiellement par défaut.

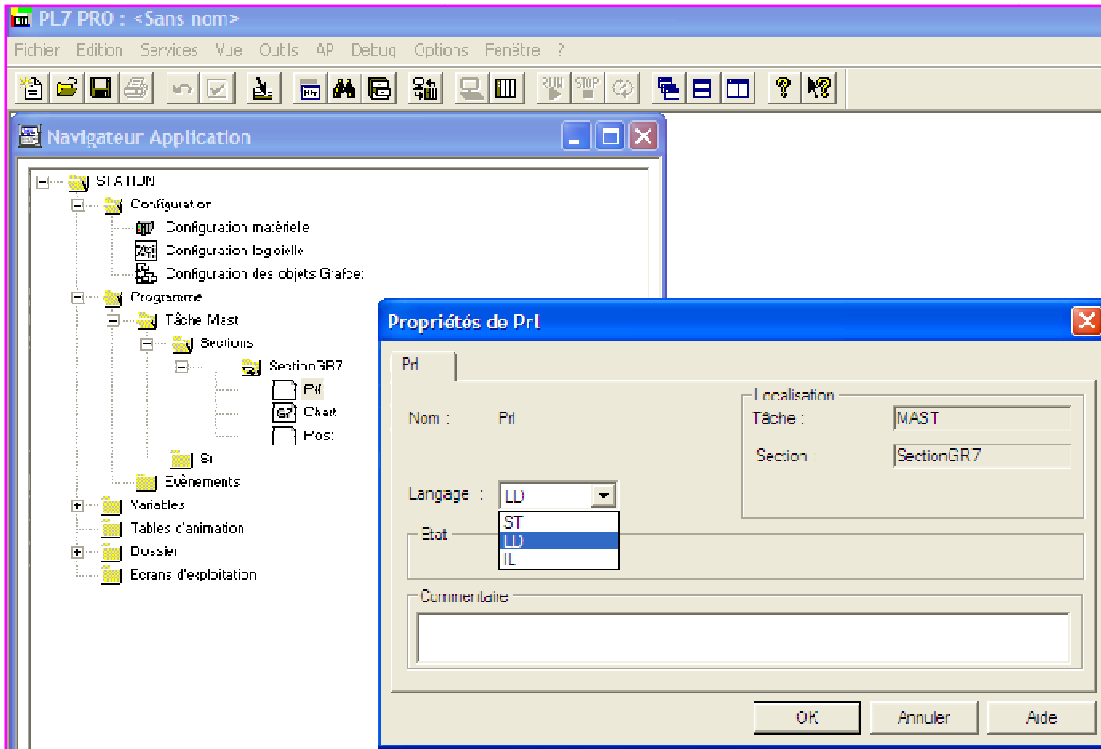
Pour commencer la programmation, on ouvre le dossier Programme, Tâche Maste, Sections, SectionGR7



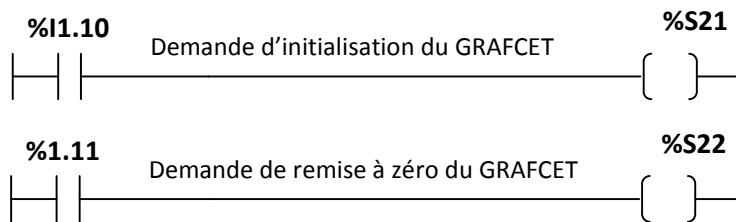
## 4-1) Traitement préliminaire

Pour une programmation d'un Grafcet, le traitement préliminaire dépend du cahier de charge.

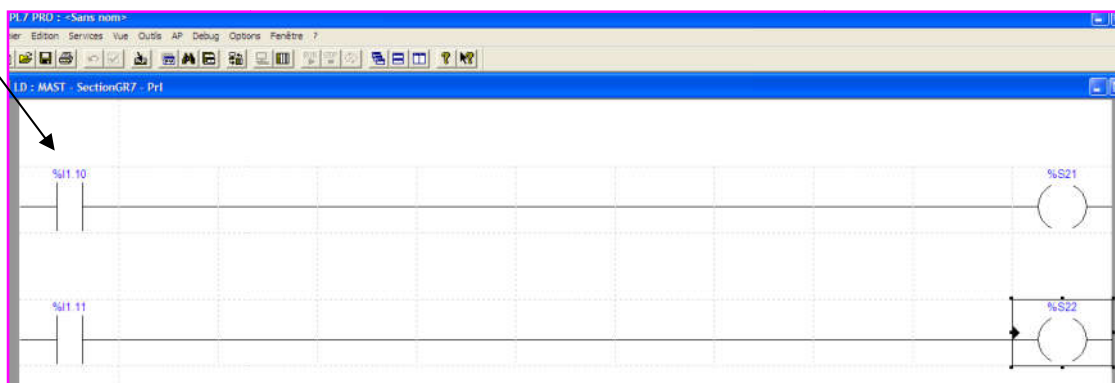
Ouvrir le traitement préliminaire en double-cliquant sur « PRL ». Il peut être programmé en Ladder (LD), structuré (ST), ou instruction list (IL), choisir « LD »



%S21 : initialisation du GRAFCET %S22 : remise à zéro du GRAFCET



Une fois le programme terminé, il faut valider, la couleur des traits et des adresses change et devient bleue.

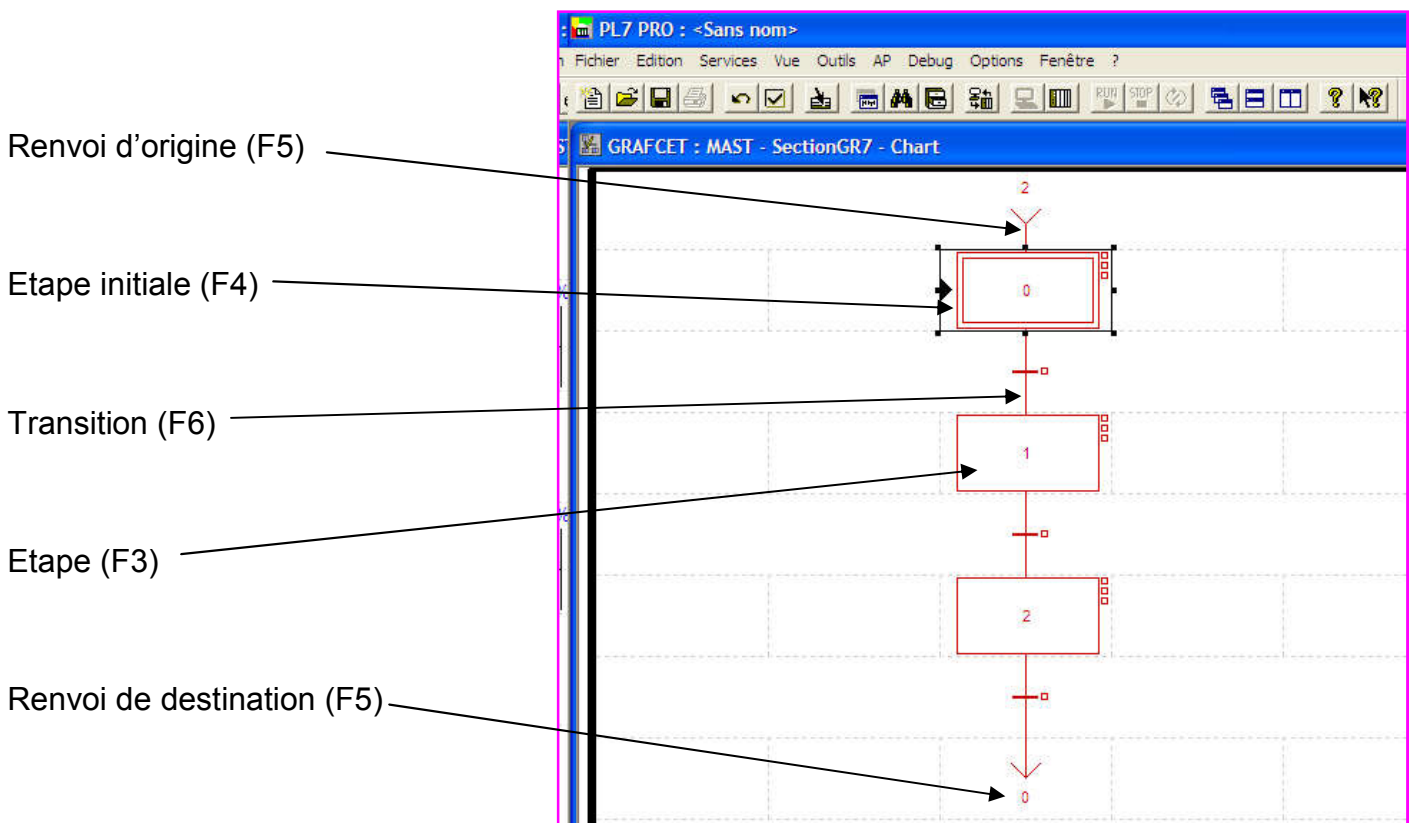


## 4-2) Traitement séquentiel

La programmation le traitement séquentiel se fait en deux étapes, la construction de l'architecture du Grafcet et la programmation des transitions.

### a) Construction de l'architecture du graphe.

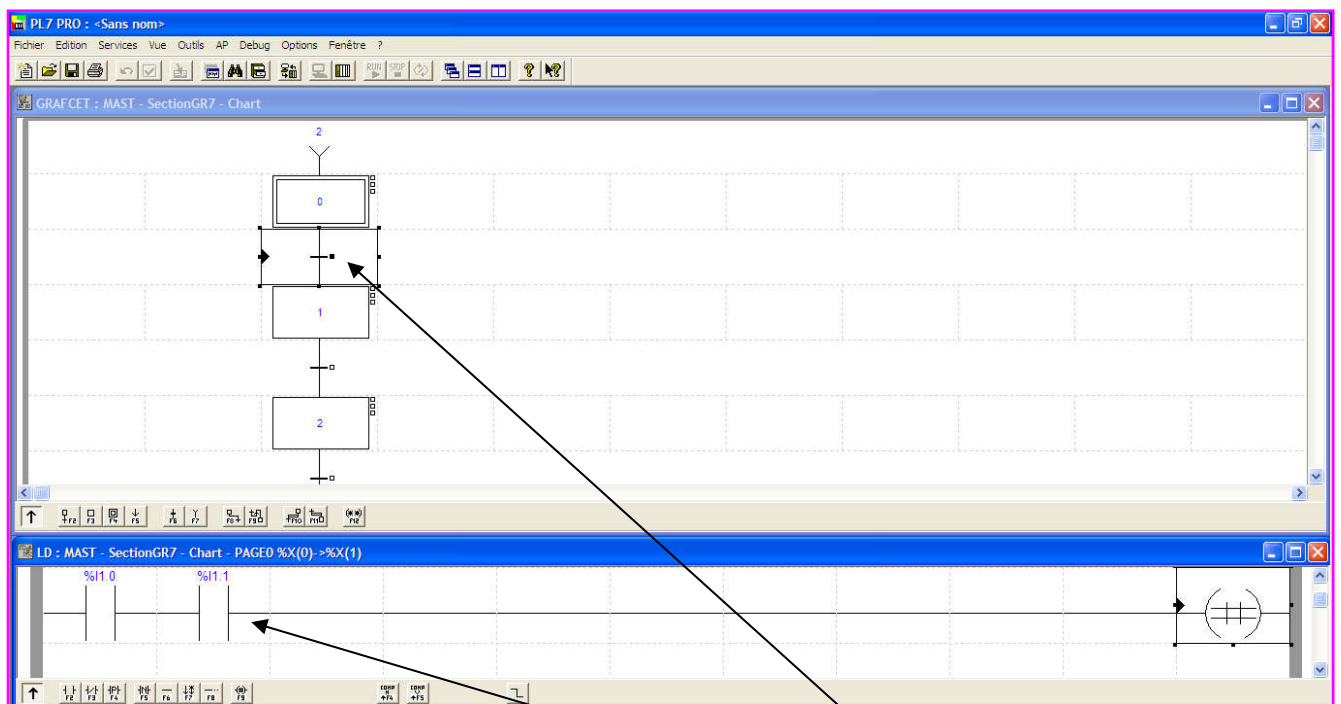
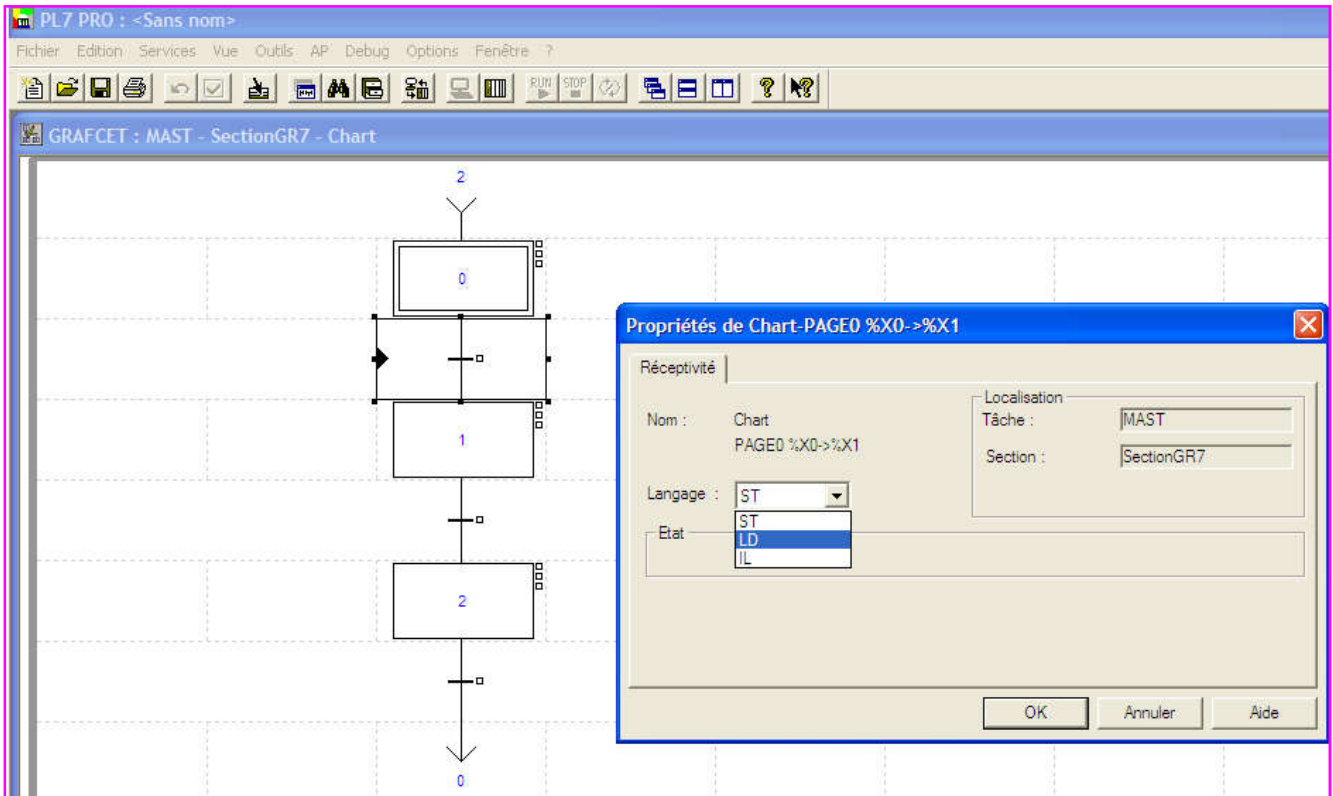
Il faut Cliquer sur **chart** (Le navigateur d'application, Programme, Tâche Maste, Sections, SectionGR7, **chart**), la page suivante s'ouvre. Compléter cette dernière en fonction du grafcet.



Une fois terminé, il faut valider, la couleur du Grafcet devient noire.

### b) Programmation des transitions

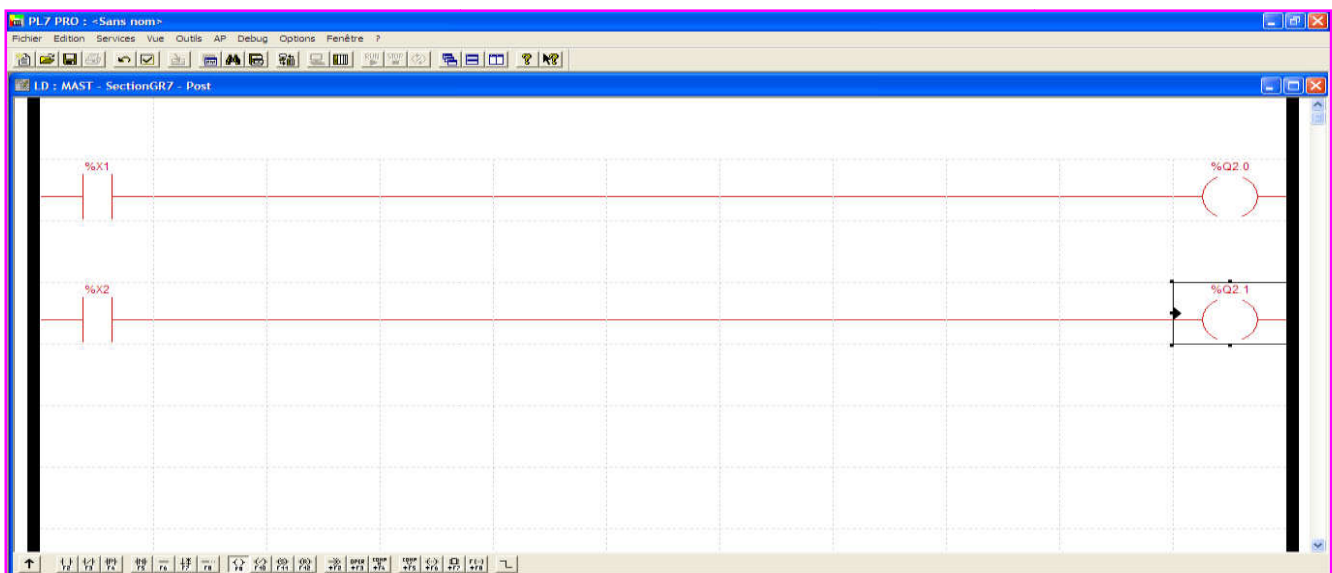
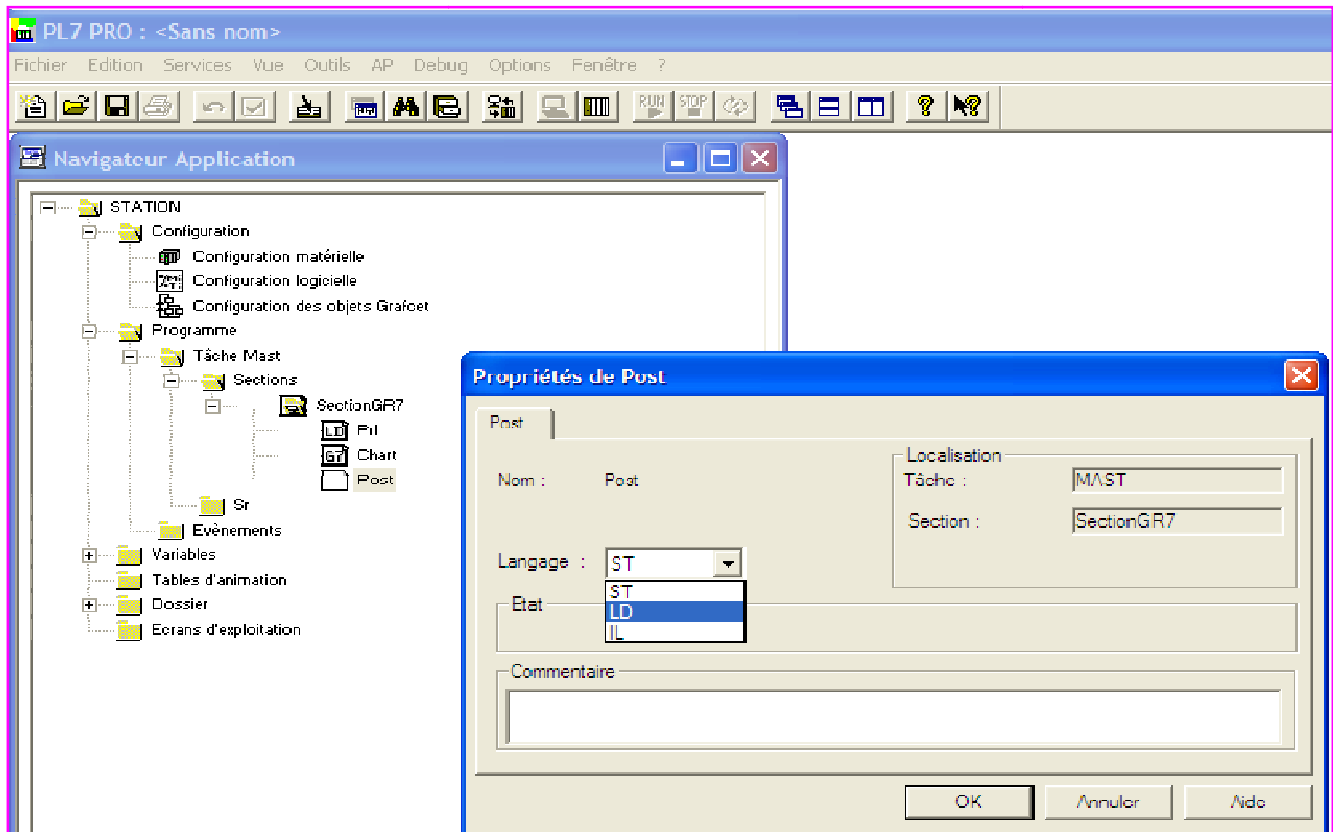
Pour la programmation des transitions, double-cliquez sur celle qui vous intéresse et sélectionnez le langage de spécification formelle de la réceptivité associée (LD pour LADDER) puis éditer la fonction combinatoire de transition (la transition agit sur la sortie implicite "#" qui est la réceptivité de l'étape suivante) .Vous validez et fermez la fenêtre de la réceptivité que vous venez de saisir afin d'en saisir une autre de la même manière.



Après validation l'écriture de l'adresse devient bleue et le carré de la transition devient noir complètement.

### 4-3) Traitement postérieur

C'est la programmation **actions associées aux étapes** . Pour traiter les actions Grafcet, ouvrir le traitement postérieur Post (Le navigateur d'application, Programme, Tâche Maste, Sections, SectionGR7, **Post**), . Sélectionner le langage de programmation (LD pour LADDER) et écrire les fonctions de sortie en utilisant les activités d'étapes %Xi .



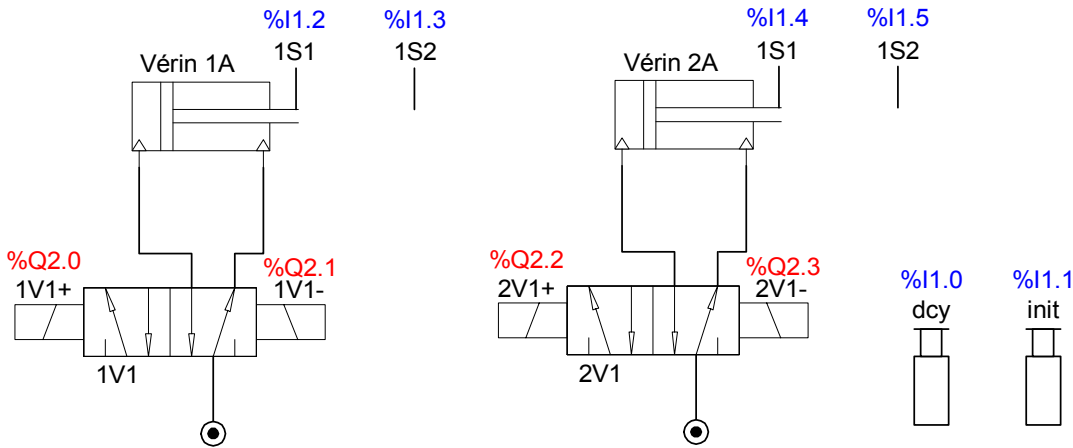
Après la programmation postérieur, il faut valider , le réseau devient noir , l'écriture des adresses bleue.

**Pour terminer il faut enregistrer le programme.**

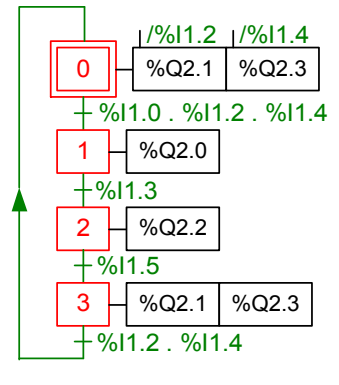


**Exemple 1 : programmation de grafcet**

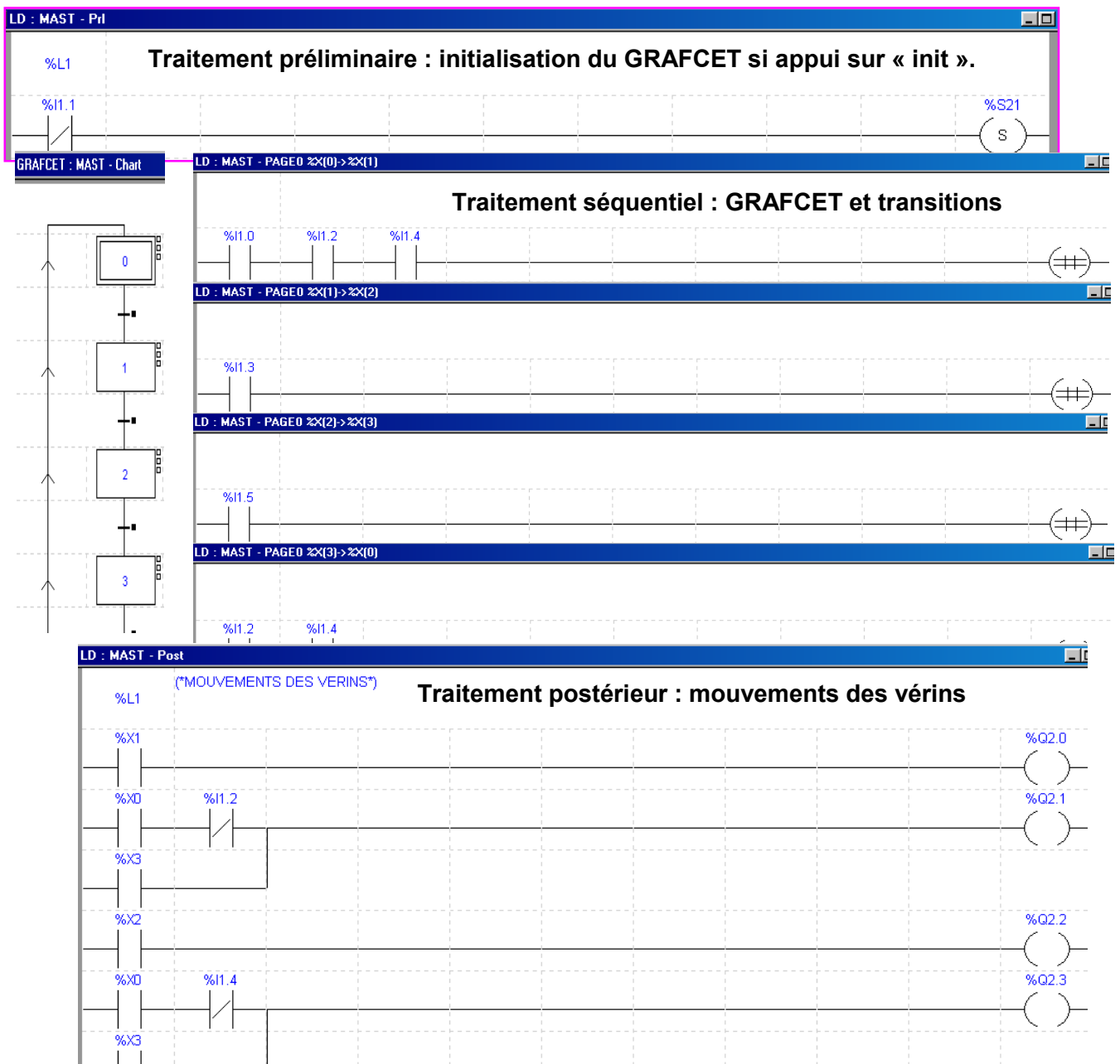
**SCHEMA PNEUMATIQUE**



**GRAFGET AUTOMATE**

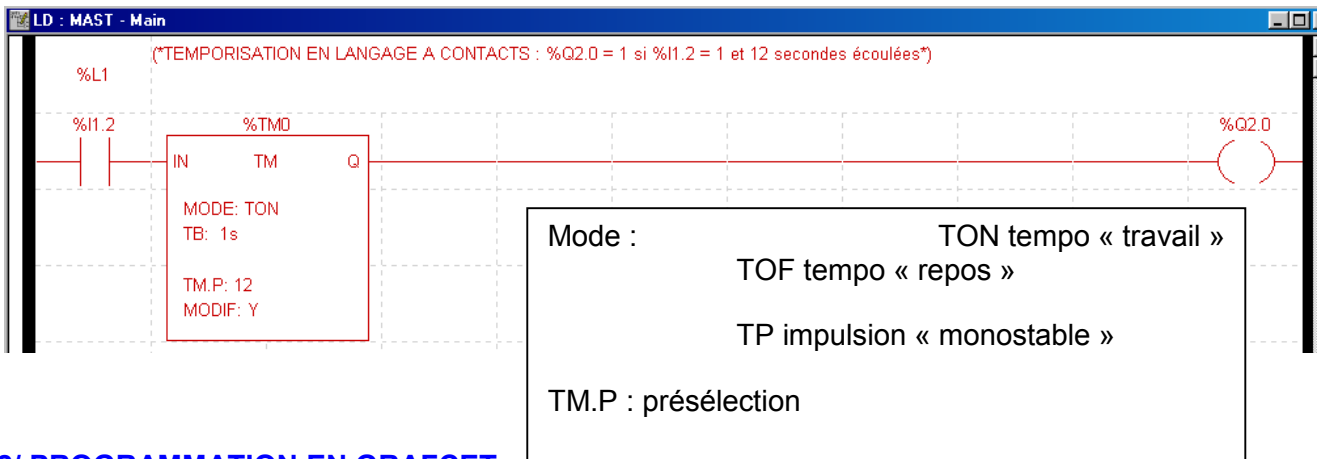


**PROGRAMME CORRESPONDANT**

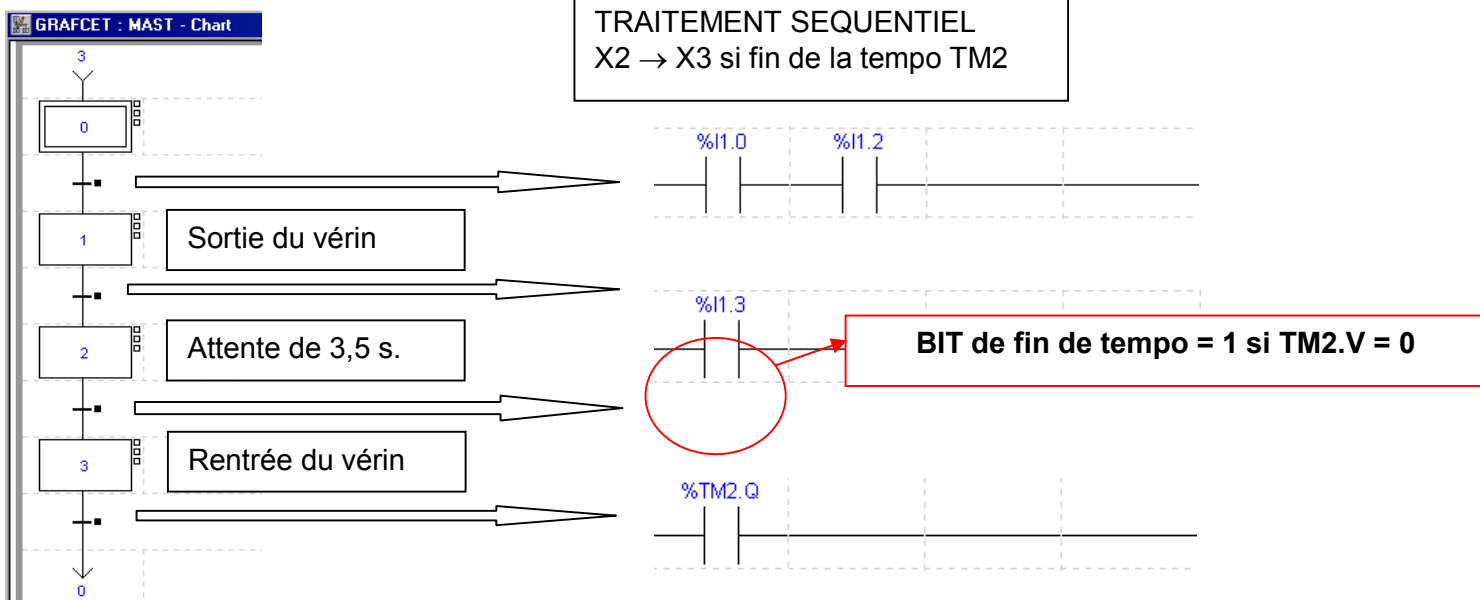
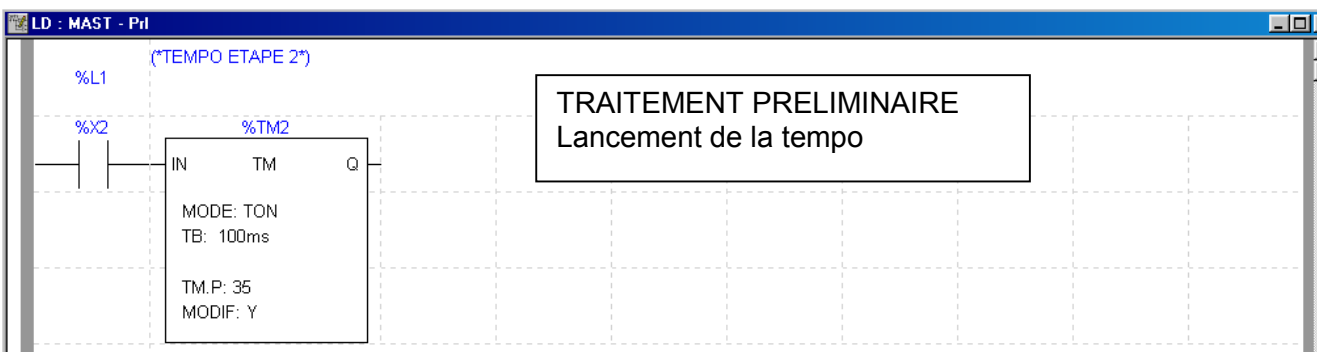


**Exemple 2 : programmation de temporisation**

**1/ PROGRAMMATION EN SCHEMA CONTACT**



**2/ PROGRAMMATION EN GRAFCET**



TRAITEMENT POSTERIEUR  
Pilotage des mouvements