



ROYAUME DU MAROC

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail
DIRECTION RECHERCHE ET INGENIERIE DE FORMATION

RESUME THEORIQUE
&
GUIDE DE TRAVAUX PRATIQUES

MODULE N°29: **SYSTÈME AUTOMATISÉ
 CONTRÔLÉ PAR API**

SECTEUR : ELECTROTECHNIQUE

SPECIALITE : ELECTROMECHANIQUE DES SYSTEMES
AUTOMATISES

NIVEAU : TECHNICIEN SPECIALISE

Module 11 : SYSTÈME AUTOMATISÉ CONTRÔLÉ PAR API

Code :

Durée : 90 h

**OBJECTIF OPÉRATIONNEL DE PREMIER NIVEAU
DE COMPORTEMENT**

COMPORTEMENT ATTENDU

Pour démontrer sa compétence l'apprenti doit
installer et réparer un système automatisé contrôlé par API
selon les conditions, les critères et les précisions qui suivent.

CONDITIONS D'ÉVALUATION

- Travail individuel.
- Sur un système automatisé électropneumatique ou électrohydraulique, fonctionnel et contrôlé par automate programmable avec E/S "tout ou rien".
- À partir d'une panne provoquée.
- À l'aide :
 - de directives et des plans du système;
 - de programmes en diagramme à échelon, en Grafcet ou en Basic;
 - de manuels du fabricant et du Code de l'électricité;
 - des outils et des instruments appropriés;
 - de composants de remplacement.

CRITÈRES GÉNÉRAUX DE PERFORMANCE

- Respect des règles de santé et de sécurité au travail.
- Respect des normes en vigueur.
- Précision du travail.
- Utilisation adéquate des outils et des instruments.
- Travail soigné et propre.
- Résultat : fonctionnement du système selon les spécifications.

(à suivre)

**OBJECTIF OPÉRATIONNEL DE PREMIER NIVEAU
DE COMPORTEMENT (suite)**

**PRÉCISIONS SUR LE
COMPORTEMENT ATTENDU**

**CRITÈRES PARTICULIERS
DE PERFORMANCE**

A. Prendre connaissance des directives, des plans et des manuels techniques.

- Interprétation exacte des symboles et des conventions du plan.
- Repérage de l'information pertinente dans les manuels techniques.
- Exactitude de la terminologie.

B. Installer les équipements.

- Installation conforme à la méthode et au plan.

C. Analyser l'état réel de l'équipement.

- Vérification minutieuse et complète de l'équipement.
- Exactitude des mesures relevées :
 - tension;
 - courant;
 - résistance.
- Justesse de la comparaison de l'état réel à l'état de référence de l'équipement.

D. Poser un diagnostic.

- Justesse du diagnostic.
- Indication valable du phénomène destructeur.
- Choix judicieux des correctifs à apporter.

E. Changer ou réparer le composant défectueux.

- Choix approprié du composant de remplacement.
- Démontage et montage précis.
- Mise en place correcte et solidité des composants de remplacement.

F. Régler les paramètres de fonctionnement.

- Réglage fonctionnel des paramètres.

G. Vérifier le fonctionnement de l'équipement.

- Prise en considération des spécifications de fonctionnement.
- Mise en marche appropriée et sécuritaire de l'équipement.
- Fonctionnement approprié :
 - des dispositifs de commande;
 - des dispositifs de protection.

H. Consigner les interventions.

- Concision et pertinence de l'information présentée.

OBJECTIFS OPÉRATIONNELS DE SECOND NIVEAU

L'apprenti DOIT MAÎTRISER LES SAVOIRS, SAVOIR-FAIRE, SAVOIR PERCEVOIR OU SAVOIR ÊTRE JUGÉS PRÉALABLES AUX APPRENTISSAGES DIRECTEMENT REQUIS POUR L'ATTEINTE DE L'OBJECTIF DE PREMIER NIVEAU, TELS QUE :

Avant d'apprendre à prendre connaissance des directives, des plans et des manuels techniques (A) :

1. Expliquer le fonctionnement d'un système automatisé.
2. Décrire les séquences de chaque mouvement d'un système automatisé.
3. Interpréter des schémas, des plans et des devis.

Avant d'apprendre à installer les équipements (B) :

4. Identifier les normes en vigueur au regard de l'installation d'un système automatisé.
5. Expliquer les méthodes d'installation.
6. Reconnaître les mesures de sécurité à prendre lors de l'installation.
7. Installer des câbles et des canalisations.

Avant d'apprendre à analyser l'état réel de l'équipement (C) :

8. Analyser un circuit à c.c.
9. Analyser un circuit à c.a.
10. Analyser des circuits à semi-conducteurs.
11. Appliquer des notions de logique combinatoire.
12. Appliquer des notions de logique séquentielle.
13. Utiliser un automate programmable.
14. Analyser des circuits pneumatiques.
15. Analyser des circuits hydrauliques.
16. Définir les étapes de mise en marche et d'arrêt du système.
17. Utiliser l'automate programmable en mode "TEST".

Avant d'apprendre à poser un diagnostic (D) :

18. Interpréter l'information obtenue par l'automate programmable.
19. Réparer le composant défectueux.
20. Déceler les pannes par la vue, l'odorat, l'ouïe et le toucher.
21. Évaluer prospectivement le temps d'une réparation.

(à suivre)

OBJECTIFS OPÉRATIONNELS DE SECOND NIVEAU

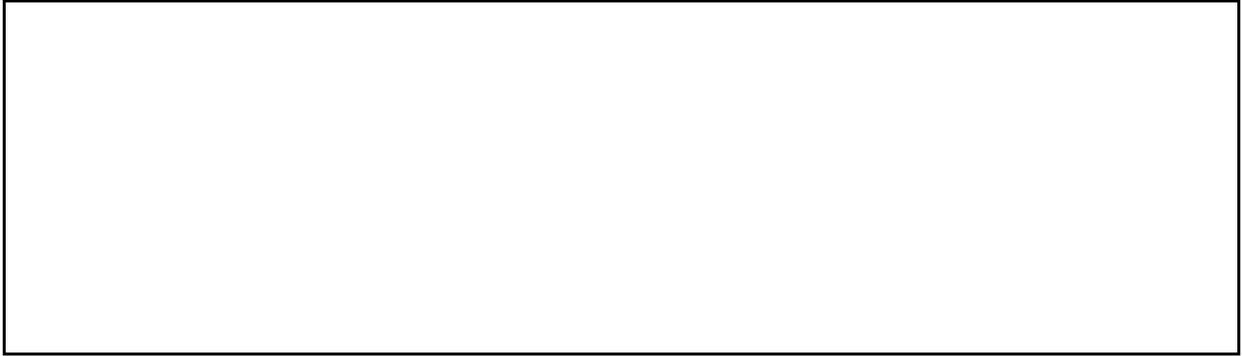
l'apprenti DOIT MAÎTRISER LES SAVOIRS, SAVOIR-FAIRE, SAVOIR PERCEVOIR OU SAVOIR ÊTRE JUGÉS PRÉALABLES AUX APPRENTISSAGES DIRECTEMENT REQUIS POUR L'ATTEINTE DE L'OBJECTIF DE PREMIER NIVEAU, TELS QUE :

Avant d'apprendre à changer ou réparer le composant de défectueux (E) :

22. Effectuer des opérations d'usinage manuel.
23. Effectuer des opérations d'usinage sur machines-outils.
24. Appliquer des techniques d'oxycoupage et de soudage à l'arc électrique.
25. Sélectionner les composants de remplacement.
26. Appliquer des méthodes d'alignement conventionnel.
27. Appliquer des techniques de gestion de la maintenance.
28. Expliquer l'importance de la qualité dans l'exécution des travaux.

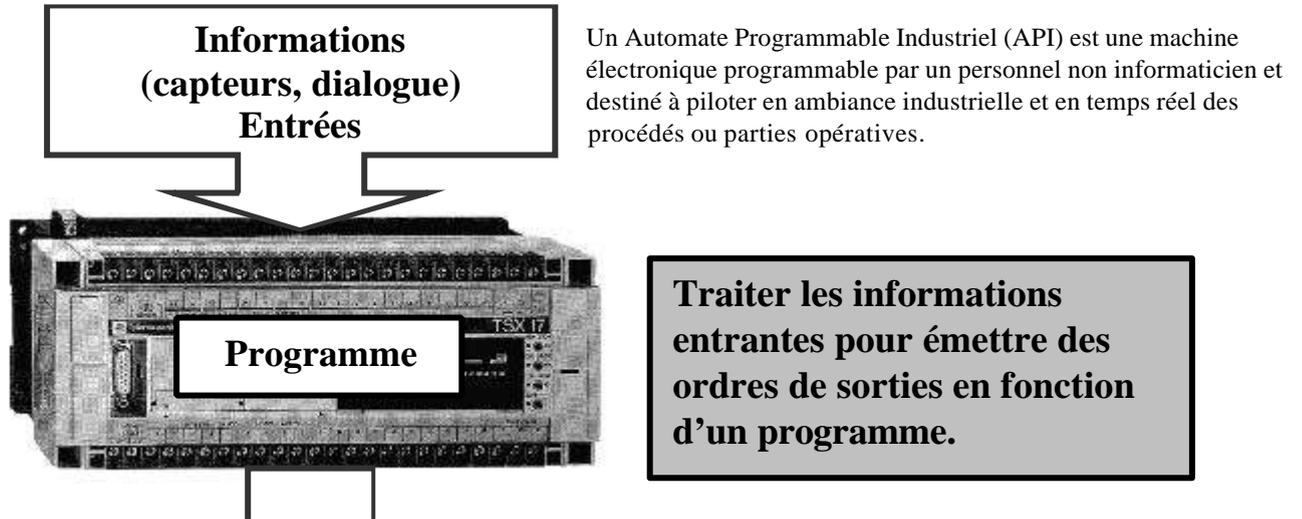
Avant d'apprendre à consigner les interventions (H) :

29. Utiliser un micro-ordinateur pour produire des documents techniques.
30. Utiliser la terminologie appropriée.
31. Expliquer l'importance de la propreté et du soin à porter aux aires de travail, aux outils et à l'équipement.



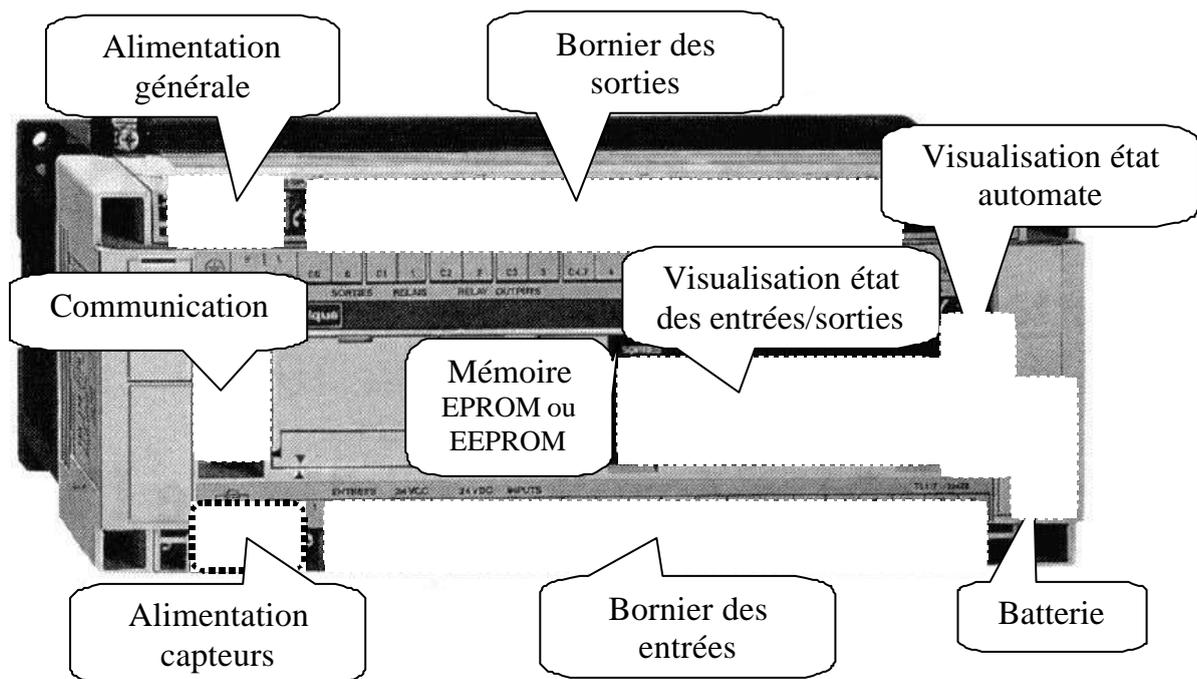
L'Automate Programmable Industriel

Définition



Un automate programmable est adaptable à un maximum d'application, d'un point de vue traitement, composants, langage.
C'est pour cela qu'il est de construction modulaire.

Structure générale



Principe de fonctionnement

Le traitement à lieu en quatre phases :

•**Phase 1 : Gestion du système**

•Autocontrôle de l'automate

•**Phase 2 : Acquisition des entrées**

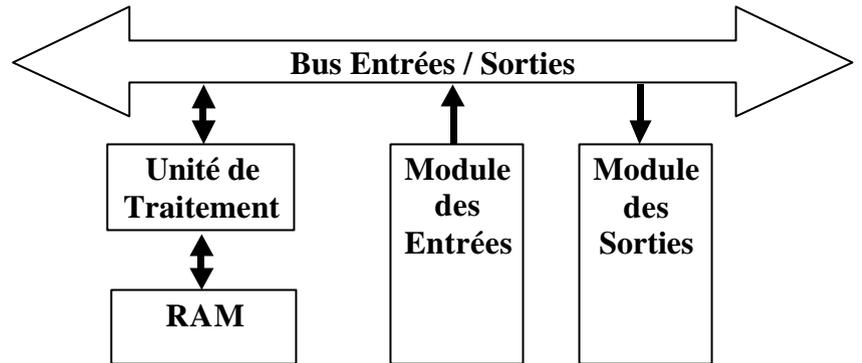
Prise en compte des informations du module d'entrées et écriture de leur valeur dans RAM (zone DONNEE).

•**Phase 3 : Traitement des données**

Lecture du programme (située dans la RAM programme) par l'unité de traitement, lecture des variables (RAM données), traitement et écriture des variables dans la RAM données.

•**Phase 4 : Emissions des ordres**

Lecture des variables de sorties dans la RAM données et transfert vers le module de sorties.



Caractéristiques techniques

Les caractéristiques principales d'un API sont :

<ul style="list-style-type: none"> •Compact ou modulaire •Tension d'alimentation •Taille mémoire •Temps de scrutation 	<ul style="list-style-type: none"> •Sauvegarde (EPROM, EEPROM, pile, ...) •Nombre d'entrées / sorties •Modules complémentaires (analogique, communication,..) •Langage
---	--

Unité Centrale

L'unité centrale est le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programmes. Les instructions sont effectuées les unes après les autres, séquencées par une horloge.

Exemple: Si deux actions doivent être simultanées, l'API les traite successivement.

Caractéristiques principales :

- Vitesses de traitement : C'est la vitesse de l'UC pour exécuter 1 K-instructions logiques. (10 à 20 ms/Kmots).
- Temps de réponse : scrutation des entrées, vitesse de traitement et affectation des sorties.

Mémoire

Deux types de mémoire cohabitent :

- La mémoire Langage** où est stocké le langage de programmation. Elle est en général figé, c'est à dire en lecture seulement. (ROM : mémoire morte)
- La mémoire Travail** utilisable en lecture-écriture pendant le fonctionnement c'est la RAM (mémoire vive).

Attribution des zones mémoire travail en RAM

Nature des Inform.	Désignations	Exploitation	Zones Mémoires
Etats des Capteurs	Variable d'entrée	Evolution de leur valeur en fonction du déroulement du cycle	Zone mémoire des Données
Ordres aux préactionneurs	Variable de sortie		
Résultats de fonctions comptage, tempo...	Variable Interne et / ou Variable mot		
Résultats intermédiaires			
Instructions du cycle dans l'API	Programme	Ecrit 1 fois et lu à chaque scrutation	Zone mémoire PROGRAMME

•**Sauvegarde :**

Sauvegarde de la RAM (programmes, configuration, données)		Sauvegarde Externe (programme, configuration)
1 heure minimum par pile interne	1an par pile externe	permanente par EPROM (effaçable par ultraviolet), EEPROM (effaçable par courant électrique)....

Le transfert de l'EPROM ou EEPROM vers la mémoire RAM de l'automate, s'effectue à chaque reprise secteur et si le contenu de celle-ci est différent.

Les Modules Entrées - Sorties

Module d'extension d'Entrées/Sorties TOR
Module réseau : communication entre automate

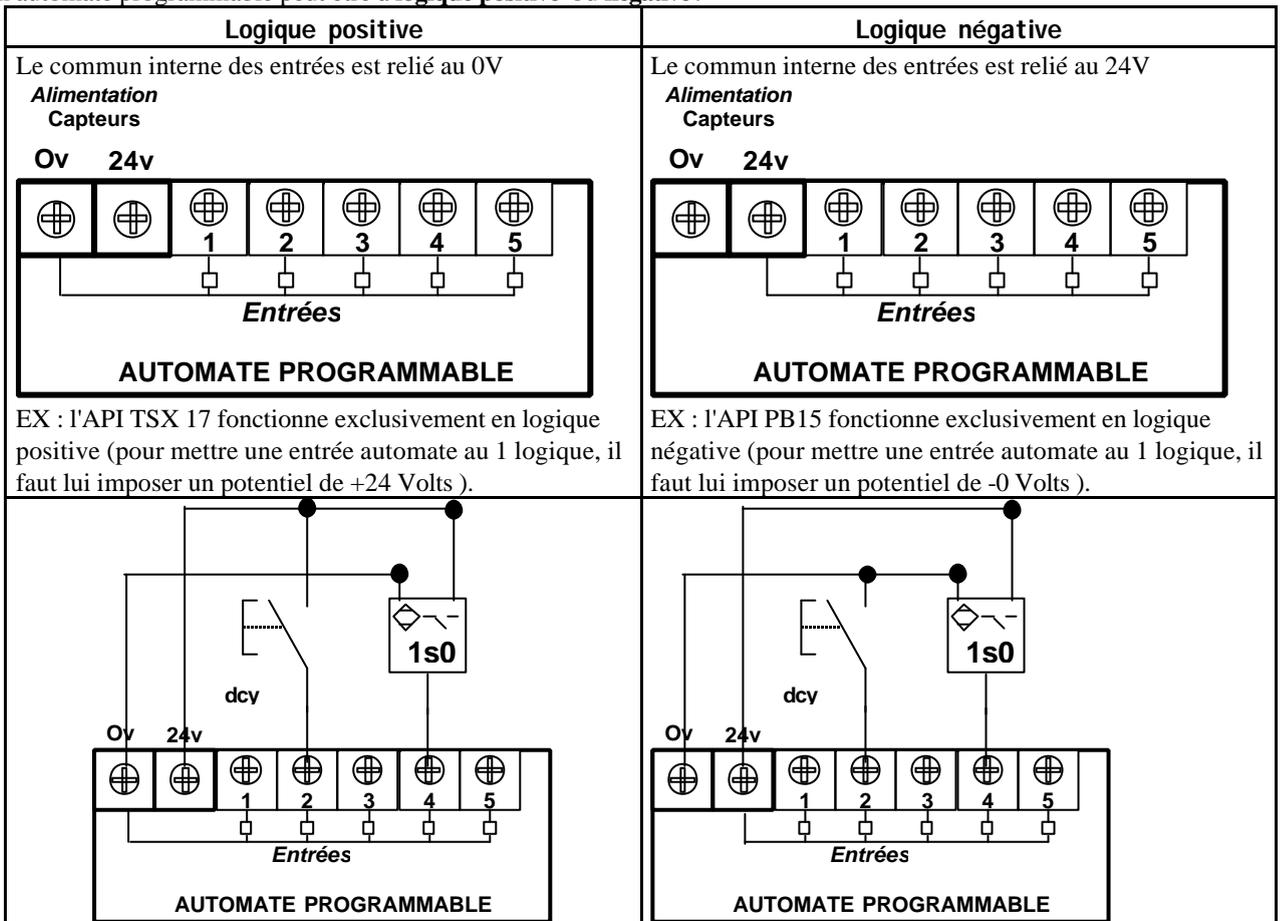
Module d'extension d'Entrées Analogiques 0-10V Module d'extension de Sorties Analogiques 0-10V

1.1.1 Branchement des Entrées TOR

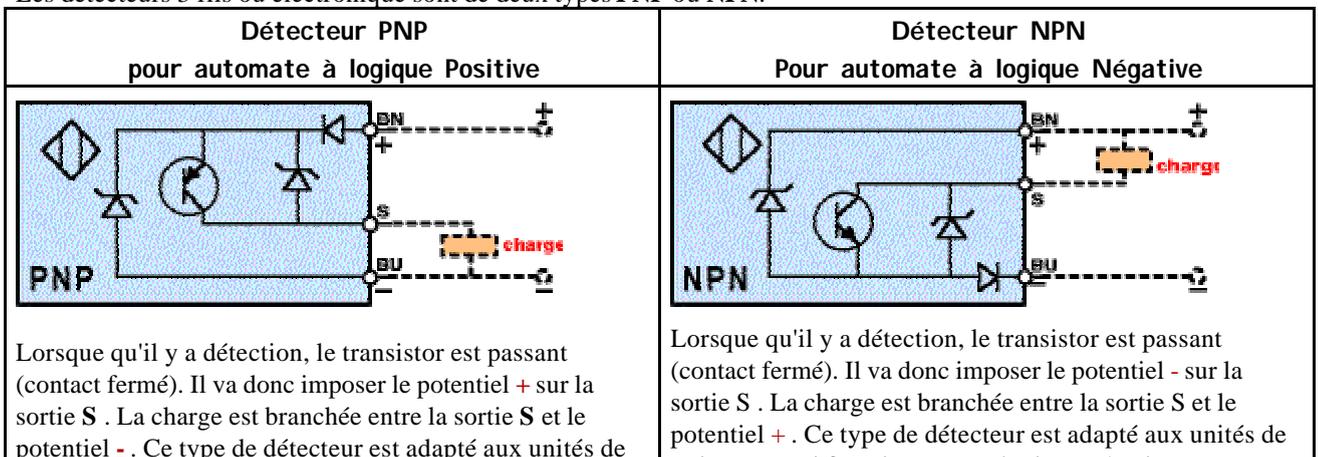
Le principe de raccordement consiste à envoyer un signal électrique vers l'entrée choisie sur l'automate dès que l'information est présente.

L'alimentation électrique peut être fournie par l'automate (en général 24V continu) ou par une source extérieure.

Un automate programmable peut être à **logique positive** ou **négative**.



Les détecteurs 3 fils ou électronique sont de deux types PNP ou NPN.



traitement qui fonctionnent en logique positive.

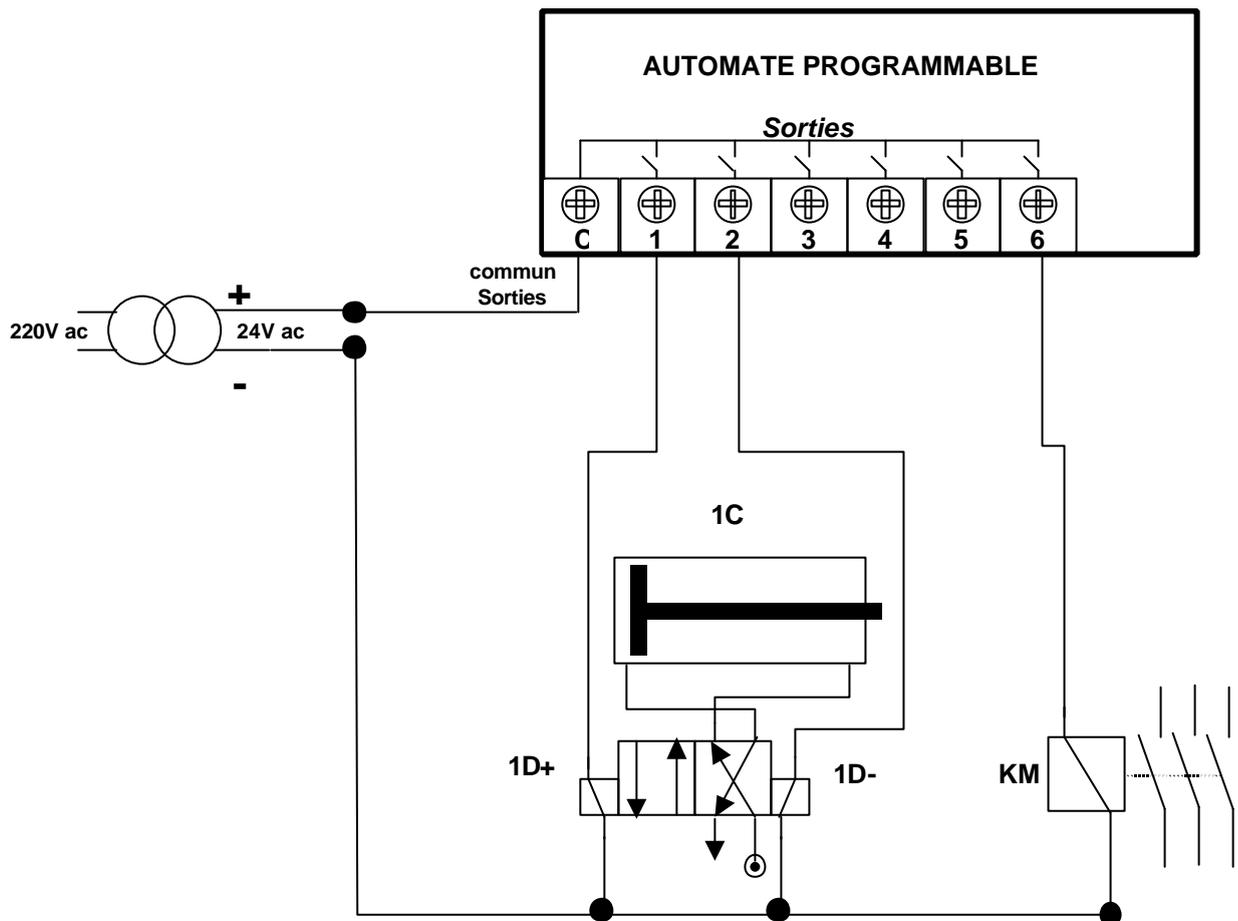
traitement qui fonctionnent en logique négative.

Pour un automate programmable la charge représente l'entrée

1.1.2 Branchement des sorties

Le principe de raccordement consiste à envoyer un signal électrique vers le préactionneur connecté à la sortie choisie de l'automate dès que l'ordre est émis.

L'alimentation électrique est fournie par une source extérieure à l'automate programmable.



Terminaux de programmation et de réglage

L'API doit permettre un dialogue avec :

- Le personnel d'étude et de réalisation pour réaliser la première mise en oeuvre (Edition programme, Transfert, Sauvegarde...)
- Le personnel de mise au point et de maintenance de réaliser des opérations sur le système (Forçage, Visualisation de l'état, Modification de paramètres temporisation, compteurs....)

Ce dialogue peut être réalisé par :

- Une Console : Elle sera utilisée sur site. Elle comporte un clavier, un écran de visualisation et le langage de programmation.
- Un Micro-ordinateur avec un logiciel d'assistance à la programmation : Il sera utilisé hors site. Il comprend plusieurs modules pour permettre l'édition, l'archivage, la mise au point des applications.

Mise en oeuvre

Préparation

La Partie Opérative du système, les grafkets de Production Normale, le Dialogue, le GEMMA (Modes de Marches et d'Arrêts), les GRAFCET de Sécurité et de Conduite étant définis, il reste à définir la Partie Commande.

Si le choix se porte sur un automate programmable, celui-ci étant relié aux préactionneurs (affectation Entrées/ Sorties) et ayant son propre langage de programmation, il faut traduire les GRAFCET précédents en un programme.

Tracer les GRAFCET adaptés à l'automate programmable.	<p>⇒ Remplacer les réceptivités et les actions par les affectations des variables d'Entrées/Sorties</p> <p>⇒ Modifier les structures GRAFCET si nécessaire en fonction des possibilités du langage de programmation.</p> <p>⇒ Préparer la programmation pour les temporisations, les compteurs, les mémorisations d'action etc.. en respectant la syntaxe du langage de programmation.</p>
Ecrire les équations de sorties	Recherche des conditions d'exécution des actions dans l'ensemble des grafkets et des équations logiques
Noter l'état initial des variables	Etapas actives au démarrage, mots de données pour tempo ou compteur)
Ecrire le programme.	<p>Il existe 2 possibilités d'édition de Programme:</p> <p>⇒ Ecrire le programme directement dans le langage programmable sur feuille de programmation. (Ex: Langage littéral booléen ou GRAFCET PB15 ou Langage Graphique Schéma à contact ou GRAFCET PL7-2 pour console TSX). Ecriture de l'ossature GRAFCET et des réceptivités, puis des équations de sorties.</p> <p>⇒ Utiliser un logiciel d'assistance à la Programmation (en général GRAPHIQUE)exemple AUTOMGEN</p>

REMARQUE: Le logiciel AUTOMGEN permet l'édition graphique proche des grafkets, puis l'affectation des entrées/sorties, la génération du programme pour l'automate concerné, la simulation du programme, le transfert et la supervision de son exécution.

Transfert du programme dans l'automate programmable

Le transfert du programme peut être fait soit :

- manuellement en entrant le programme et l'état initial à l'aide d'une console de programmation
- automatiquement en transférant le programme à l'aide du logiciel d'assistance, et en réalisant la liaison série entre l'ordinateur et l'automate.

Vérification du fonctionnement

Lors de sa première mise en oeuvre il faut réaliser la mise au point du système.

⇒ **Prendre connaissance du système** (dossier technique, des grafjets et du GEMMA, affectation des entrées / sorties, les schémas de commande et de puissance des entrées et des sorties).

⇒ **Lancer l'exécution du programme** (RUN ou MARCHE)

⇒ **Visualiser l'état des GRAFCET, des variables...**

Il existe deux façons de vérifier le fonctionnement :

- En simulation (sans Partie Opérative).
- En condition réelle (avec Partie Opérative).

Simulation sans P.O.	Condition réelle
<p>Le fonctionnement sera vérifié en simulant le comportement de la Partie Opérative, c'est à dire l'état des capteurs, en validant uniquement des entrées.</p> <p>⇒ Valider les entrées correspondant à l'état initial (position) de la Partie Opérative.</p> <p>⇒ Valider les entrées correspondant aux conditions de marche du cycle.</p> <p>⇒ Vérifier l'évolution des grafjets (étapes actives).</p> <p>⇒ Vérifier les ordres émis (Leds de sorties).</p> <p>⇒ Modifier l'état des entrées en fonction des ordres émis (état transitoire de la P.O.).</p> <p>⇒ Modifier l'état des entrées en fonction des ordres émis (état final de la P.O.).</p> <p>⇒</p> <p>Toutes les évolutions du GEMMA et des grafjets doivent être vérifiées.</p>	<p>Le fonctionnement sera vérifié en suivant le comportement de la P.O.</p> <p>⇒ Positionner la P.O. dans sa position initiale.</p> <p>⇒ Valider les conditions de marche du cycle.</p> <p>⇒ Vérifier l'évolution des grafjets et le comportement de la P.O.</p> <p>⇒ ...</p> <p>Toutes les évolutions du GEMMA et des grafjets doivent être vérifiées.</p>

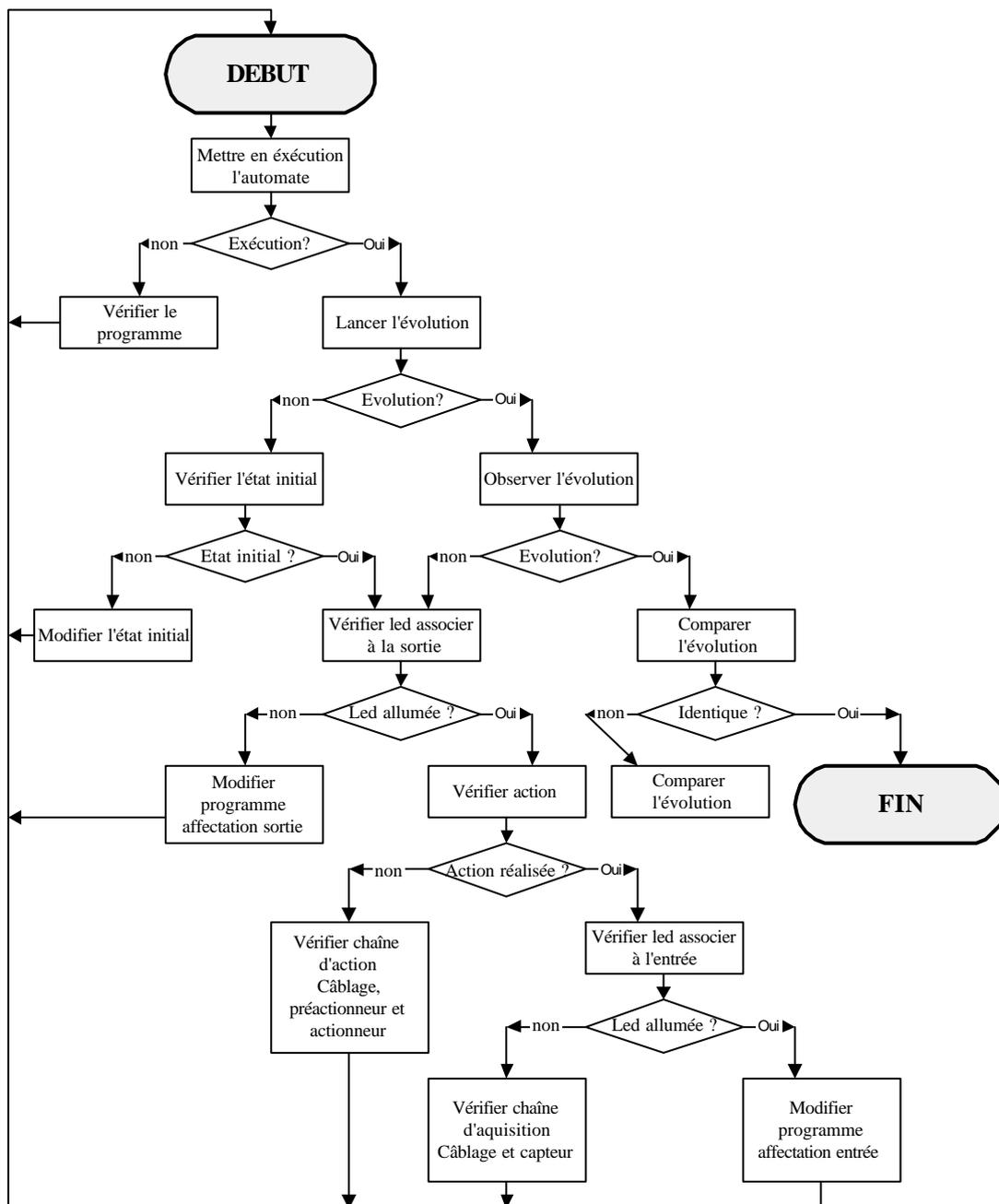
Recherche des dysfonctionnements

1.1.3 Causes de dysfonctionnements

Un dysfonctionnement peut avoir pour origine :

- un composant mécanique défaillant (préactionneur, actionneur, détecteur,...).
- un câblage incorrect ou défaillant (entrées, sorties).
- un composant électrique ou électronique défectueux (interface d'entrée ou de sortie).
- une erreur de programmation (affectation d'entrées-sorties, ou d'écriture).
- un système non initialisé (étape, conditions initiales...).

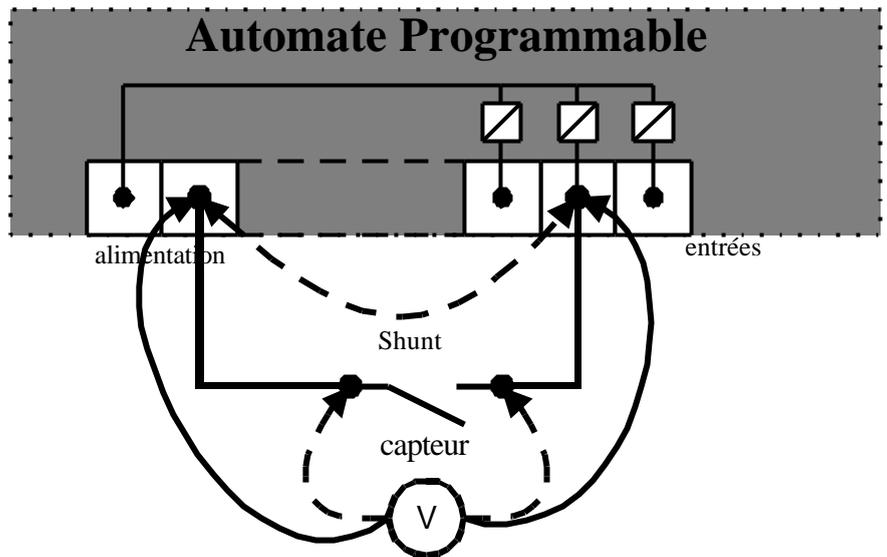
1.1.4 Méthode de recherche des causes de dysfonctionnement



1.1.5 Vérification du câblage d'une entrée à masse commune

Cette vérification se réalise à l'aide d'un voltmètre-ohmmètre et d'un shunt (morceau de fil électrique).

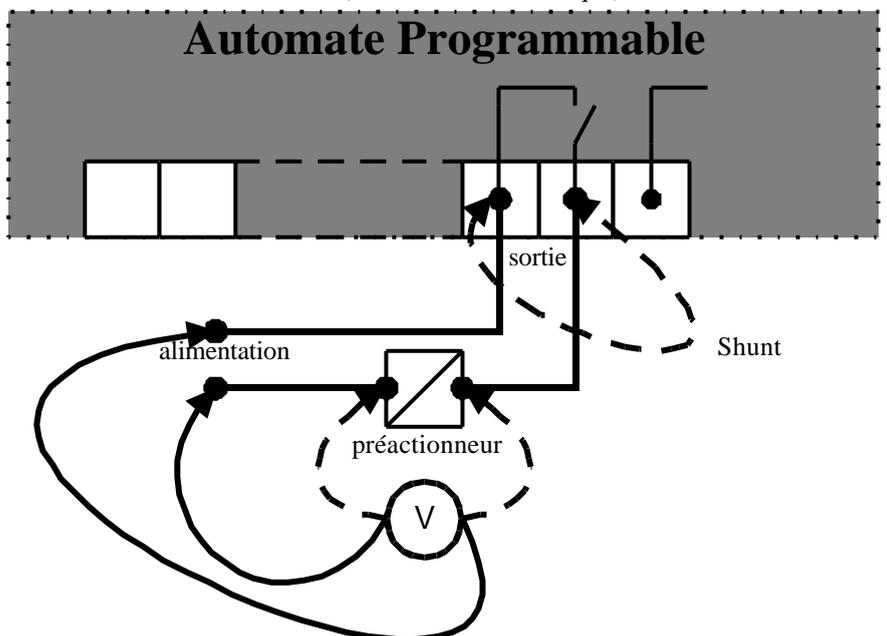
- ⇒ Vérifier l'alimentation des entrées à l'aide d'un voltmètre.
- ⇒ Pour vérifier le capteur et son câblage, tester aux différents points indiqués à l'aide d'un ohmmètre, contact du capteur ouvert, contact du capteur fermé.
- ⇒ Pour vérifier l'interface d'entrée court-circuiter le capteur par un shunt, le voyant d'entrée doit s'allumer.



1.1.6 Vérification du câblage d'une sortie à relais

Cette vérification se réalise à l'aide d'un voltmètre-ohmmètre et d'un shunt (morceau de fil électrique).

- ⇒ Vérifier que U alimentation existe à l'aide du voltmètre.
- ⇒ Forcer à l'aide du shunt la sortie automate. Si le préactionneur fonctionne, c'est le module de sortie qui est défectueux. Sinon vérifier le préactionneur et son câblage.
- ⇒ Pour vérifier le câblage tester aux différents points de connexion à l'aide d'un Ohmmètre en laissant le shunt.

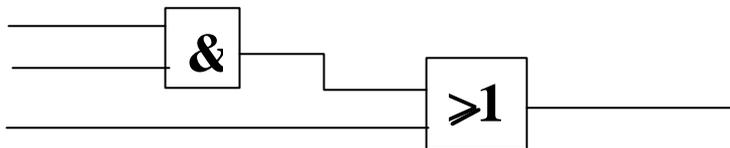


L'AUTOMATE PROGRAMMABLE

A.P.I (T.S.X 17)

1°) Les A.P.I et les systèmes automatisés

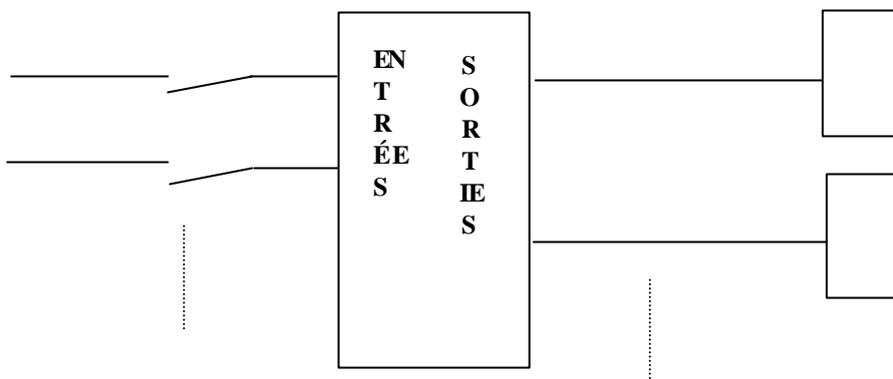
- Les premiers automatismes réalisés, l'étaient à l'aide de circuits à portes logiques (ET, OU, NAND, ...). Ces circuits étaient fragiles et non modulables, donc non adaptés à de petites modifications car il fallait tous revoir d'où du temps et une énorme perte de production



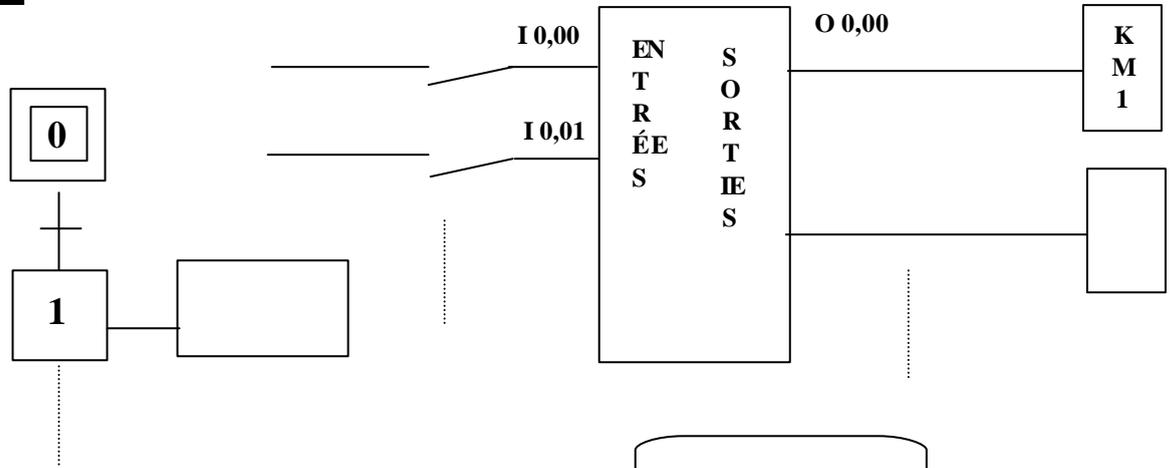
- Au fil du progrès, les automates ont vu le jour, ce qui modifia le traitement des informations, réduisit les cabines de câblage et rendit les systèmes plus flexibles. En fait les automates reprennent le fonctionnement des portes logiques mais maintenant elles sont programmables et réduites à un faibles encombrement.

2°) Comment fonctionne un A.P.I.

- Un API est comme un cerveau, il reçoit des informations de ces capteurs, les compare à son programme et active ou pas des sorties.

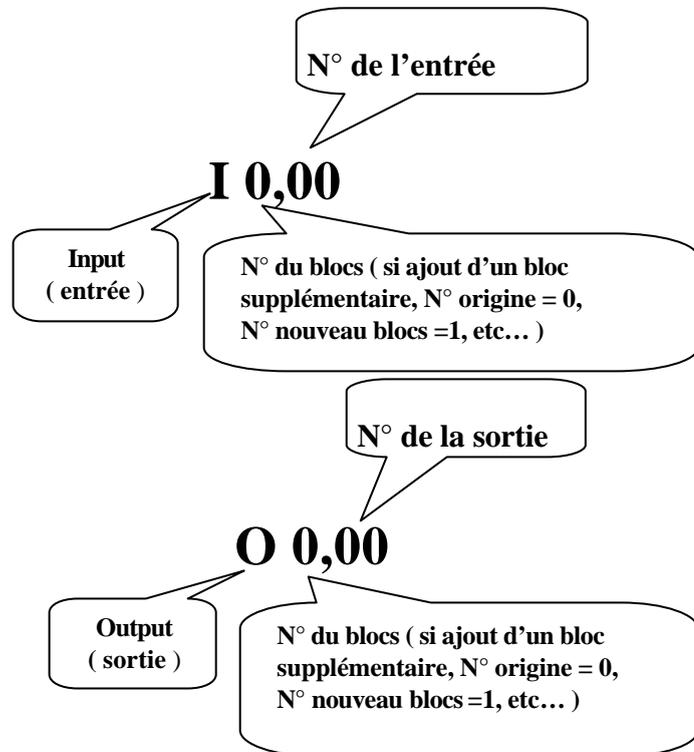


3°) Exemple

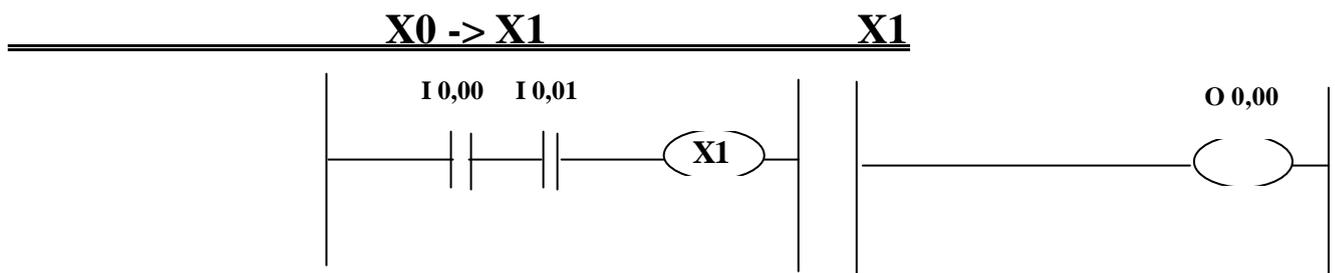


- Traduction pour l'API :

- S1 = **I 0,00**
- S2 = **I 0,01**
- KM1 = **O 0,00**



- Programmation de l'A.P.I. (ici en séquentiel)



4°) En résumé :

- On programme l'A.P.I d'après un grafcet séquentiel de fonctionnement, on rattache des capteurs à des entrées API nommées I X,XX et des relais, contacteurs à des sorties API nommées O X,XX.

5°) Le langage A.P.I (TSX 17).

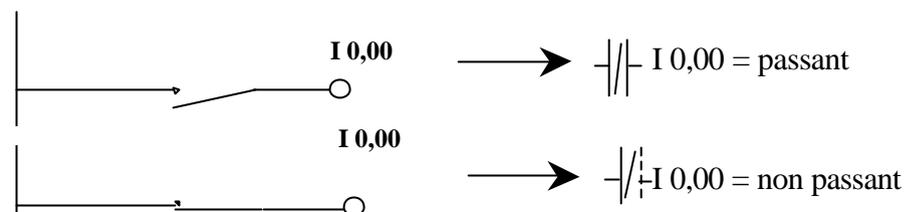
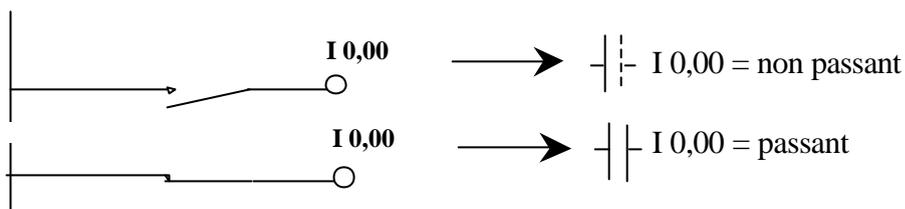
5.1°) Les entrées :

- Il existe 2 types d'entrées appelé « test » pour l'API :

les test directes :  qui s'active si l'entrée est active.

les tests indirectes :  qui s'active si l'entrée n'est pas active (idem cellule inverse).

- Exemple :



5.2°) Les sorties :

- Il existe plusieurs types de sorties pour l'API, mais nous n'allons en étudier que 3 sortes :

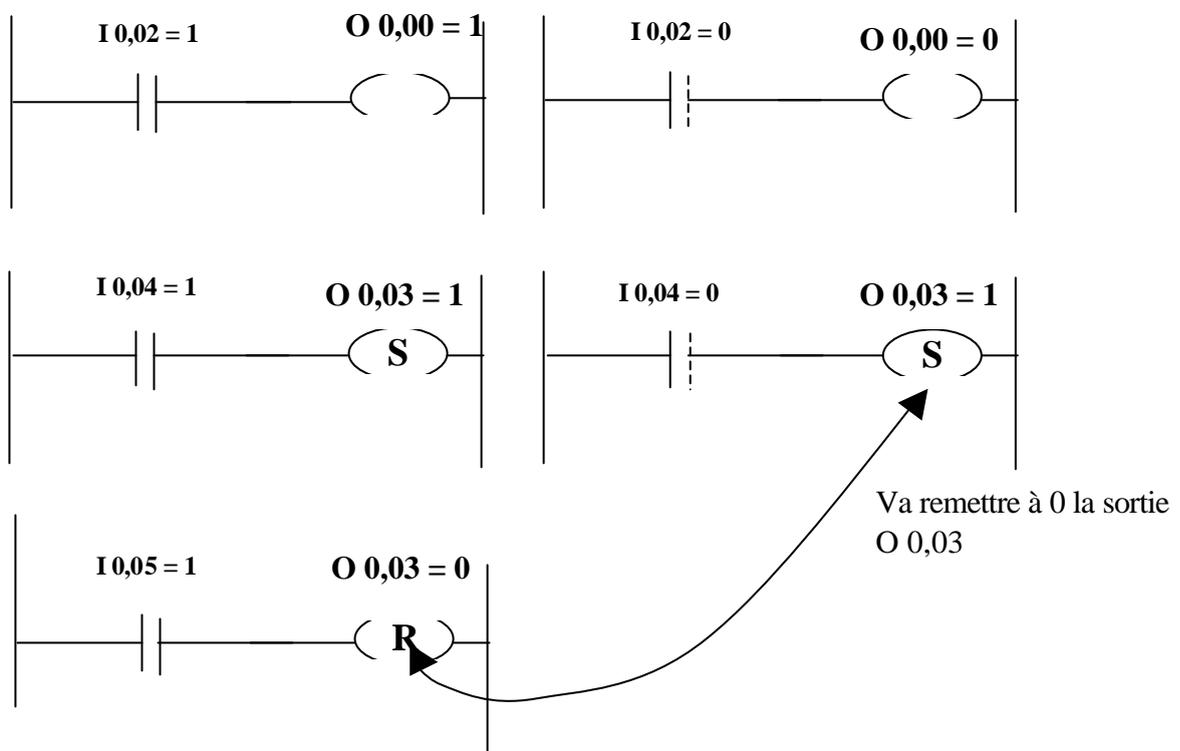
- Sortie directe :  qui fonctionne comme l'entrée direct.

- Sortie SET (S) :  qui agit comme une mémoire et reste à 1.

- Sortie RESET (R) :  met à zéro la sortie S.

Nota : Les sorties S et R sont surtout utilisées pour les préactionneurs monostables ou des information à mémoriser (auto-maintien de contacteur à maintenir activé pour ne pas qu'il retombe à l'état repos et pour la **programmation des étapes du grafcet en mode séquentiel**.

5.3°) exemple :



6°) La programmation :

- il existe 2 modes de programmations ; le mode **LADDER** et le mode **SEQUENTIEL**

- La principale différence réside dans l'ordre d'exécution, en LADDER, il n'y a pas d'ordre dans les LABELS, le label 102 peut s'exécuter avant le label 1 ! donc il n'est pas ou mal adapté à une programmation par séquence (type grafcet) ; contrairement au séquentiel qui lui suit directement une programmation grafcet.

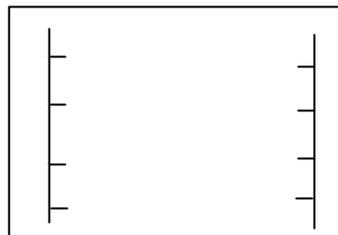
6.1°) Le mode de programmation LADDER.

- Le **LADDER** ou *langage à contact* se programme par ligne avec les entrées (test) et les sorties sur les mêmes pages de programmations.

- Chaque page se nomme un LABEL, ils vont de 0 à 999, chaque label comprend 4 lignes de programmations, il faut le nommer pour que l'A.P.I. le reconnaisse comme ligne de programmation. Pour le nommer, il faut utiliser la touche « LAB » et le numéroter de 0 à 999.

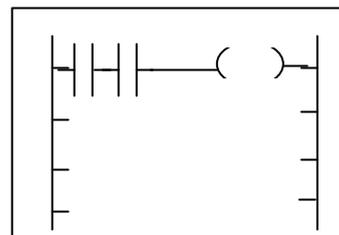
- les labels **ne comportent pas d'ordre d'exécution**, c'est-à-dire que le label 50 peut s'exécuter avant le label 1 ; on ne peut pas programmer suivant un ordre séquentiel (suivant un grafcet) car *une entrée peut enclencher plusieurs sorties si elle est utilisée dans plusieurs labels.*

- L'écran se compose de 4 lignes de programmation :



- Sur ces lignes on peut programmer des entrées et des sorties pour former une ligne de programme.

- Les sorties peuvent être des bobines ou des compteurs, des temporisations , ...



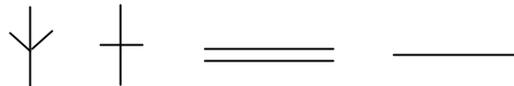
6.2°) Le mode de programmation SEQUENTIEL .:

- Dans le mode SEQUENTIEL, on trouve 3 parties :
- Le préliminaire ou « **PRE** » qui s'occupe de la *gestion des mises en marche* après arrêts désirés ou non (arrêts d'urgences, ...)
- Le séquentiel ou « **SEQ** » dans lequel on va écrire la structure ou « squelette » du grafcet et où l'on peut aussi *programmer ce grafcet* (à condition qu'il ne soit pas trop important : 4 lignes de programme par étape et uniquement en sortie **SET** et **RESET**). .
- Le postérieure ou « **POS** » dans lequel on programme comme en langage *LADDER* mais où l'on peut utiliser le grafcet pour ordonner les *LABELS* (en incluant des tests directs appartenants aux bits d'étapes, exemple : X11); chose impossible en mode *LADDER*.

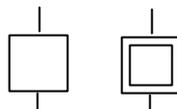
- **6.2.1°) Le séquentiel :**

- La première chose à réaliser est la création de la structure du grafcet, pour cela vous devez vous *positionner dans le mode séquentiel*. Tout d'abord vous devez *initialiser l'automate* en mode « **SEQUENTIEL** ».
- Vous arrivez ensuite dans l'écran de création du grafcet, cet écran est en fait une partie de page composer de 14 lignes horizontales et 8 colonnes. Chaque lignes comprend différents symboles et chaque ligne à ses symboles :

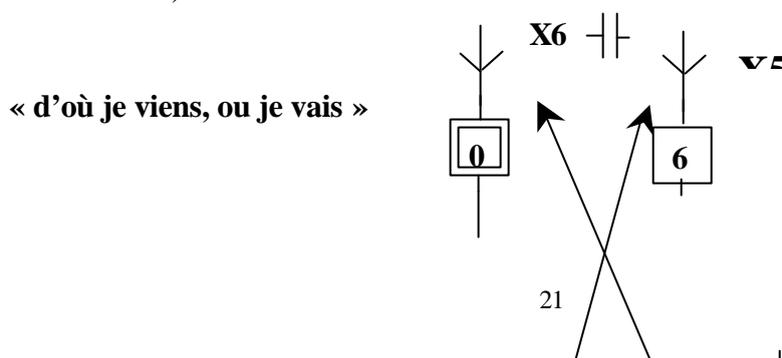
- Lignes paires : renvoies, transitions, lignes, et, ou,

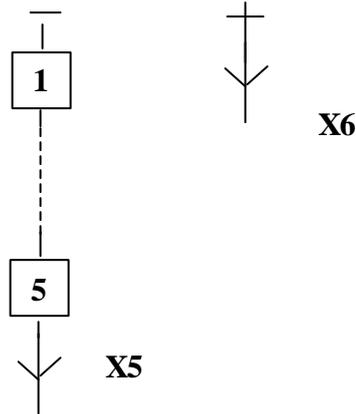


- Lignes impaires : étapes,



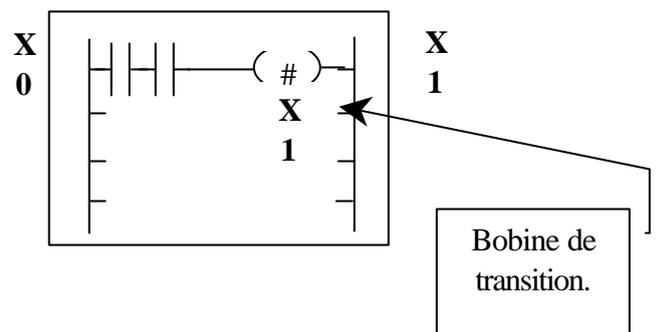
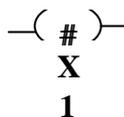
- Vous rentrer ensuite le grafcet en utilisant les renvoies s'il dépasse les 5 étapes (voir ci-dessous).





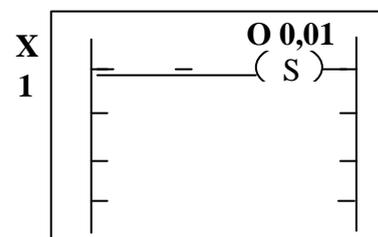
- Une fois le grafcet entré dans l'A.P.I., vous pouvez le programmer à condition que le programme ne soit pas très conséquent (pas plus que **4 lignes**). Pour cela vous devez positionner le curseur devant l'étape à programmer à l'aide des *touches de direction* et « **ZOOMER** » dans l'étape ou la transition choisie à l'aide de la touche « **ZOOM** ».
- Vous pouvez ensuite rentrer votre programme mais il faut savoir que les seules « **sorties** » disponibles sont de la forme « **SET** » et « **RESET** ». *Il faut donc penser à toujours « riseter » une étape après l'avoir « seter » car sinon le programme se bloque : deux bobines de distributeur ne peuvent et ne doivent pas être enclenchées en même temps.*
- La programmation s'effectue donc sur des étapes et sur des transitions mais les écrans n'ont pas les mêmes fonctions :
- Les transitions : représentation des entrées.

Elles ne contiennent pas de sorties, car elle enclenche l'étape 1, représenté par :



- Les étapes : représentations des sorties

Elles ne contiennent pas, en règle



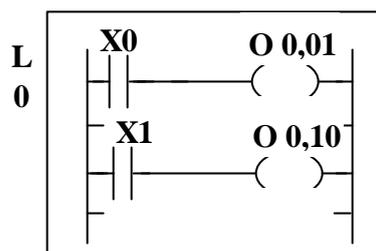
générale d'entrées sauf sécurité ou
bit de temporisation., de compteur,

O 0,00
(R)

Il ne faut pas oublier de « reseter » une bobine que l'on utilise plus

6.2.2°) Le postérieur :

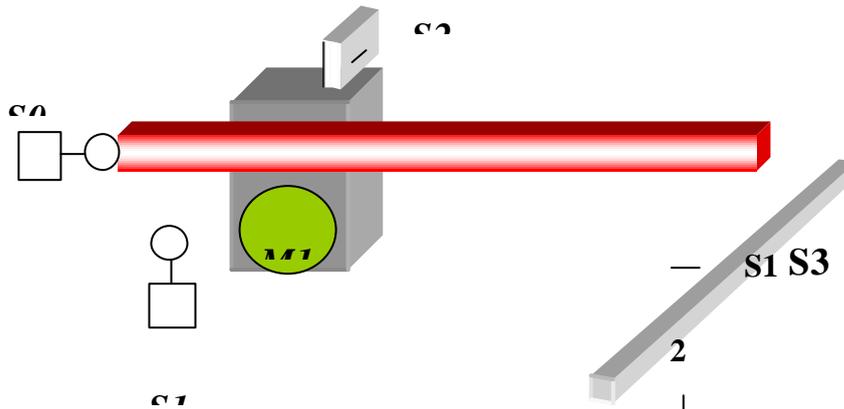
- Le postérieure se programme comme le **LADDER** mais on utilise les *bits interne d'étape* de l'automate pour pouvoir programmer d'après le grafcet. Chaque étape activée active un bit interne du même nom que l'étape, (étape X1 = bit X1). Ce bit est utilisé ensuite comme entrées permettant d'activé des sorties (voir exemple ci-dessous)



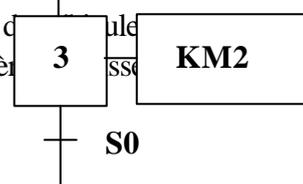
- Dans le postérieure comme dans le LADDER, toutes les sorties sont disponibles (set ,reste, direct, jump,...) .

7°) Exercice :

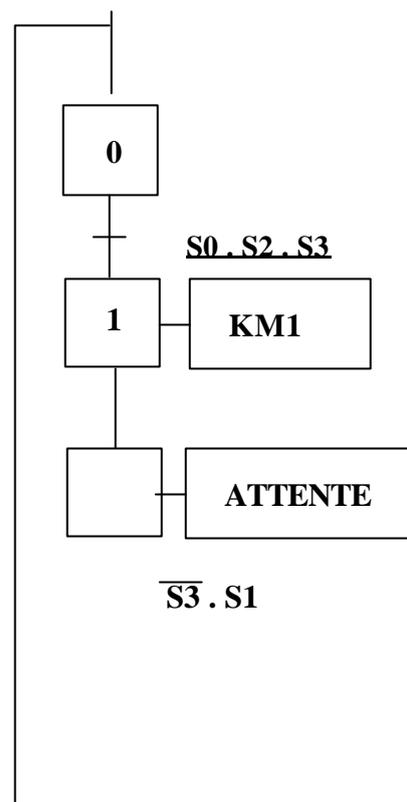
- Nous allons réaliser une programmation étape par étape d'après un grafcet de fonctionnement séquentiel.



- Pour entrer dans un parking il faut une carte et une présence d'un véhicule. Lorsque la cellule s3 indique que le véhicule est passé, la barrière se lève et



- GRAFCET :



7.1°) Identification du nombre d'entrées et de sorties

- Entrées = _____
- Sorties = _____

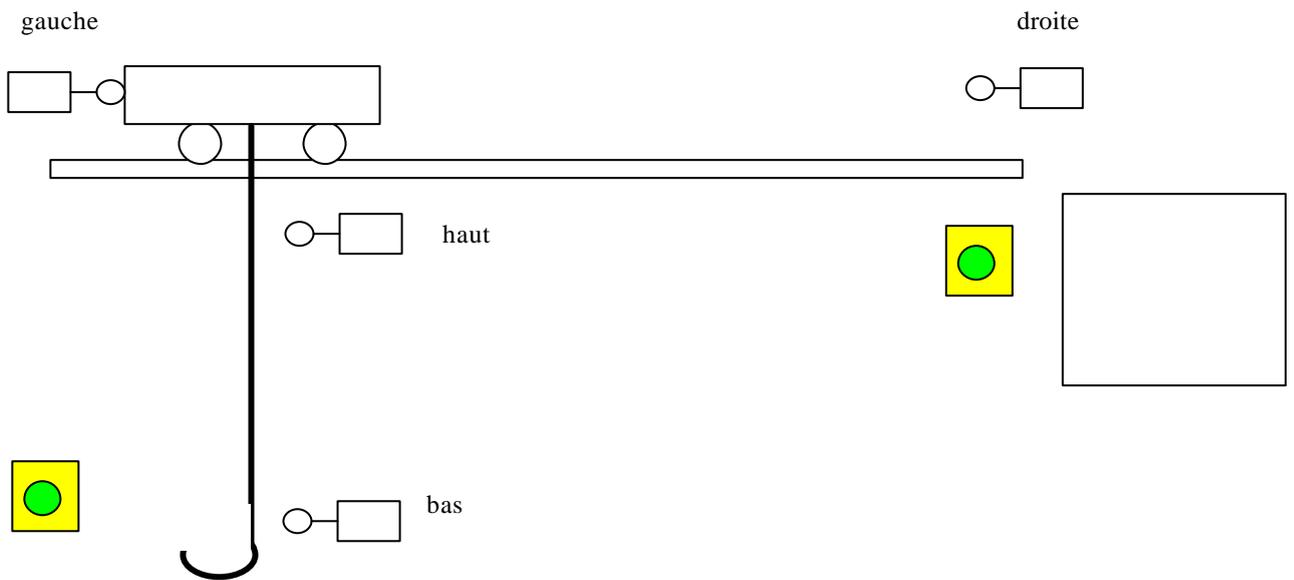
7.2 °) Traduction des entrées, sorties en langage API :

- S0 = I __ , __ __
- S1 = I __ , __ __
- S2 = I __ , __ __
- S3 = I __ , __ __
- KM1 = O __ , __ __
- KM2 = O __ , __ __

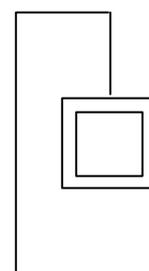
7.3°) Réaliser le programme en **LADDER** et indiquer vos constatations, puis réaliser ce même programme en **SEQUENTIEL** et indiquer vos constatations :

Nota : Pour cet exercices, nous allons utiliser des « *bits internes* » qui serviront de relais (ou de mémoire) pour différentier certaines parties du cycle et éviter des « nœuds » (répétition de deux actions au même moment).

- Le bit interne s'utilise comme une « sortie », mais elle n'occupe pas de place extérieure, c'est à dire qu'elle ne peut pas être câblée.



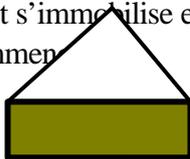
8°) Application



- **8.1°) LE PALAN :**

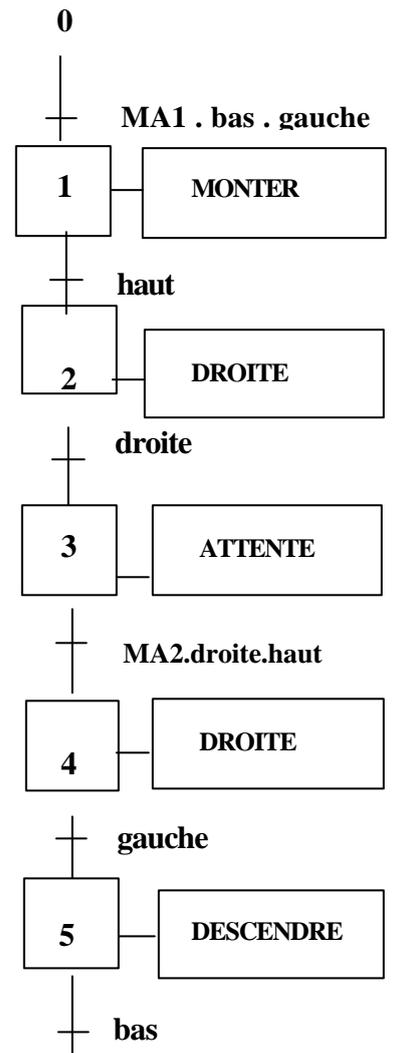
Ce palan fonctionne comme indiqué sur le grafcet ci-contre, une 1^{ère} impulsion sur le bouton MA1 fait monter la charge qui une fois en haut se dirige vers la droite, arrivée à droite tout s'arrête pour laisser un opérateur décharger la palette.

Une fois la palette déchargée, l'opérateur appuie sur bouton MA2 et refais partir la palette vers la gauche. Une fois à gauche, la palette redescend et s'immobilise en bas pour être de nouveau chargée et le cycle recommence.



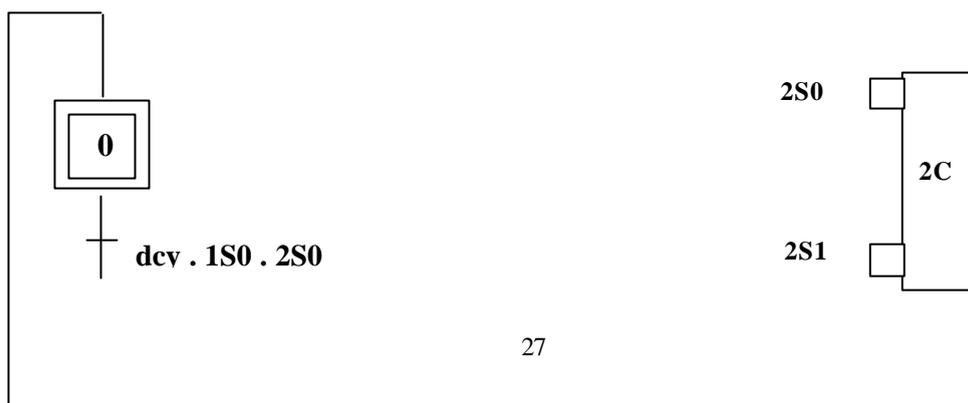
- **Travail demandé :**

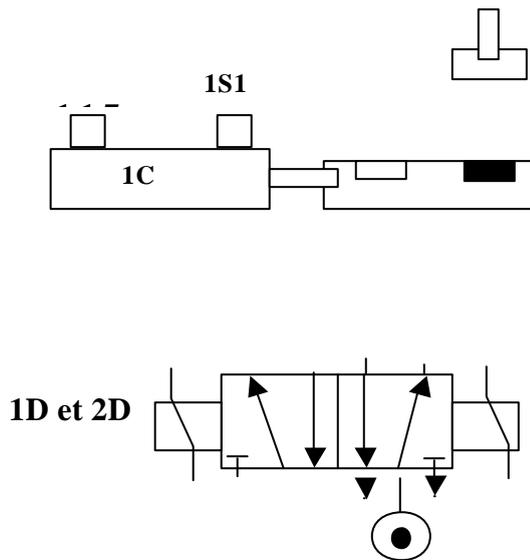
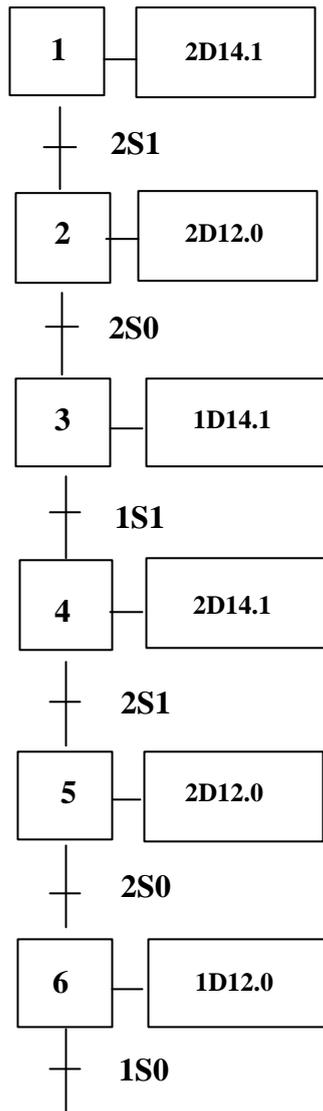
- 1°) Nommer toutes les ENTREES / SORTIES
- 2°) Ecrire le programme en **LADDER**.
- 3°) Programmer l'A.P.I
- 4°) Rajouter une temporisation.
- 5°) Rajouter un compteur.
- 6°) Réaliser la programmation en **SEQUENTIEL**.



- **8.2°) LA TABLE D'IMPRIMERIE :**

Une table d'imprimerie permet de réaliser des petites cartes de visites à la demande, il vous est demandé de réaliser le programme suivant le grafcet ci-dessous :





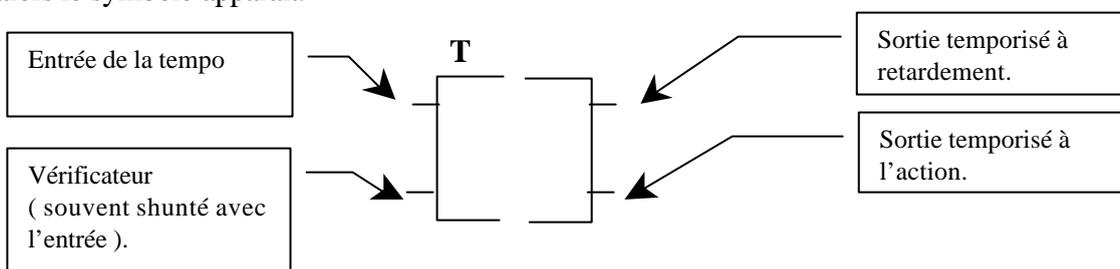
14.0

12.0

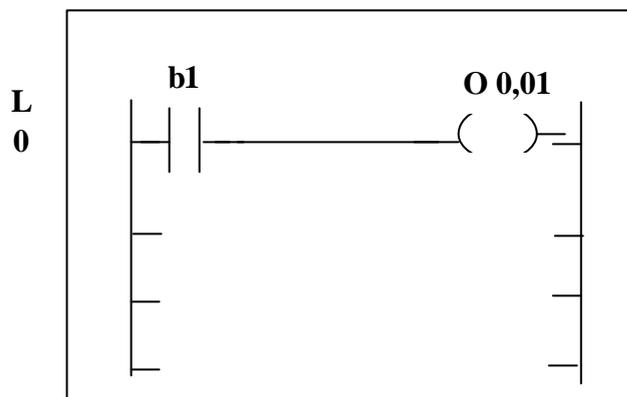
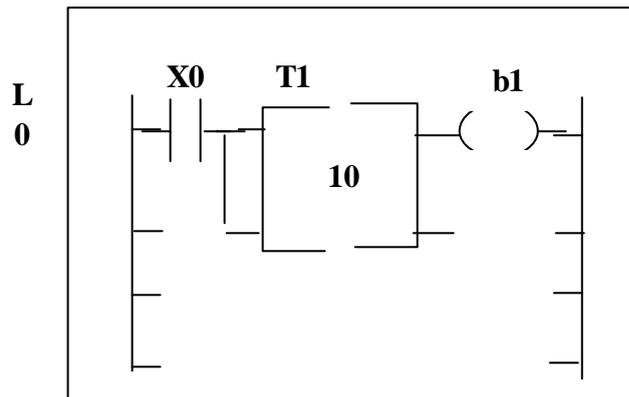
- 1°) En premier lieu il convient de dénommer chaque entrées et chaque sorties.
- 2°) Ecrire le programme sous la forme de « LABEL » sur le format ci-joint.
- 3°) Une fois validé par le formateur, vous pouvez essayer sur l'A.P.I en utilisant le manuel « programmation TSX 17 ».

9°) Les temporisations :

- La temporisation est utilisée pour temporiser une action ou sortie ; elle peut être à l'action ou au retardement. Pour trouver le symbole d'une tempo, il faut appuyer sur l'icône [?], cet icône englobe les tempos mais aussi les compteurs, les comparateurs, Pour sélectionner la tempo, choisir l'icône [T] et alors le symbole apparaît.



- Pour programmer la tempo, il faut « zoomer » dans la tempo, et programmer le temps de base « TB » de 10 ms, 100 ms, 1 s, 1 mn et la durée « PRE » de 1 à 9999.
- Ensuite il faut choisir le contact de sortie suivant l'effet temporisé désiré, et le raccorder à la sortie ou au bit à activer.



Nota : La temporisation peut être programmée dans le grafctet, si le **programme** n'est pas trop **conséquent** sinon il faudra le faire dans le postérieure.

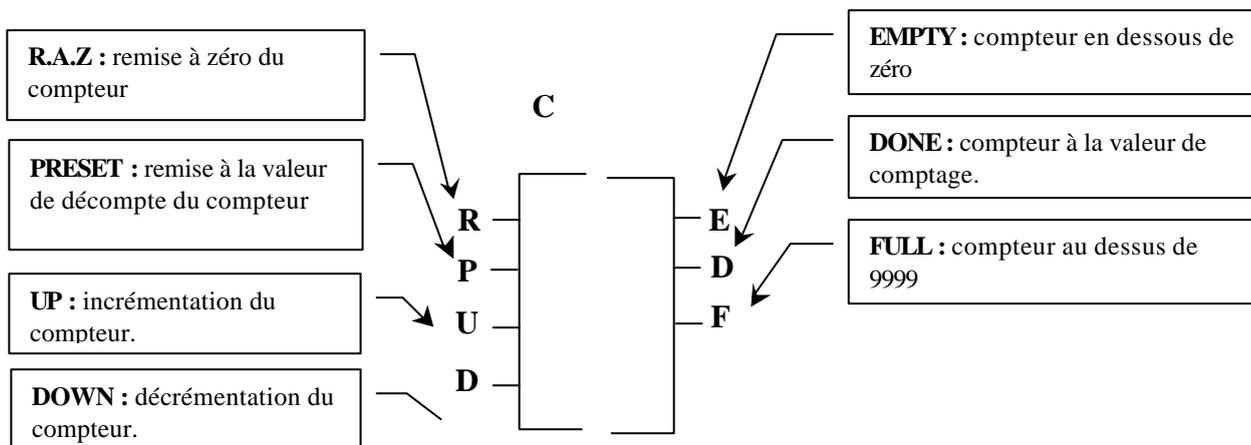
10°) Exercices :

- Pour réalisé cet exercice, positionnez vous dans le postérieure.
- Créer un nouveau LABEL à la suite de ceux de l'exercice précédent, et utilisé comme test direct une entrée non utilisée ; puis allé sélectionner une tempo. L'API vous demandera de lui donner un numéro de 0 à 20, choisissez en un et valider.

- Raccordez à chacune des deux sorties de la tempo, une bobine de sortie direct non utilisée.
- Programmer la tempo (deux fois « ZOOM ») pour qu'elle compte en seconde (TB) jusqu'à 5 (PRESET).
- Que constatez-vous ?

11°) Le compteur :

- Le processus pour implanter un compteur est exactement le même que pour la temporisation ; seul la fonction et le câblage diffèrent.

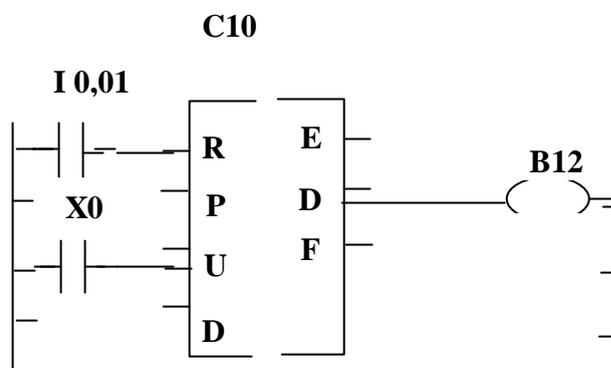


- Le compteur permet de compter des impulsions (ou des mises à 1 de tests d'entrées) et d'incrémenter ou de décrémentation de 1 la valeur de comptage du compteur.
- Si la valeur de comptage est atteinte, la sortie « DONE » se met à un.
- Si la valeur de comptage dépasse 9999, la sortie « FULL » se met à 1.
- Si la valeur de comptage décroît en dessous de 0, la sortie « EMPTY » se met à 1.
- Pour remettre le compteur à zéro en cas de comptage par incrémentation, il faut activer l'entrée « R.A.Z ».
- Pour remettre le compteur à la valeur programmée en cas de décrémentation, il faut activer l'entrée « PRESET ».

Nota :

1. La sortie « DONE » se met à un lorsque le compteur a atteint sa valeur programmée, si le comptage continu, **la sortie se remet à zéro**. Si l'on veut garder l'information plus longtemps, il faudra activer une sortie « SET » et la remettre à zéro (RESET) par l'intermédiaire de l'entrée « R.A.Z » ou « PRESET ».
2. Le compteur se programme toujours dans le postérieur dans un label entier (4 lignes).

- Exemple de câblage :



12°) Exercices :

12.1°) Réaliser le comptage jusqu'à 5 de la mise à un du test d'une entrée de votre choix dans le postérieure (sans toucher au programmes précédents), la fin du comptage activera une sortie non utilisée. Une fois la programmation du label effectué mettez vous en mode « run » et effectué vos essais.

12.2°) Insérez une tempo dans le cycle de la platine d'imprimerie pour chaque descente du tampon :
première tempo de 5 secondes (à l'encrage) pour bien encrer le tampon,
deuxième tempo de 2 secondes (au marquage) pour bien encrer la carte de visite.

12.3°) Insérer un compteur pour compter 5 cycles et enclencher une sortie, raccorder à un voyant, pour prévenir l'opérateur qu'il faut nettoyer les lettres du tampon. Essayer ensuite de mettre en série avec la sortie, un test d'entrée nommer « SY6 » et noter ce qu'il se passe.

12.4°) Insérer un compteur pour compter 5 cycles et enclencher une sortie, raccorder à un voyant, pour prévenir l'opérateur qu'il faut nettoyer les lettres du tampon. Essayer ensuite de mettre en série avec la sortie, un test d'entrée nommer « SY5 » et noter ce qu'il se passe.

12.5°) Même énoncé, mais le cycle doit se bloquer et n'être déverrouillable que par une entrée non utilisée précédemment .

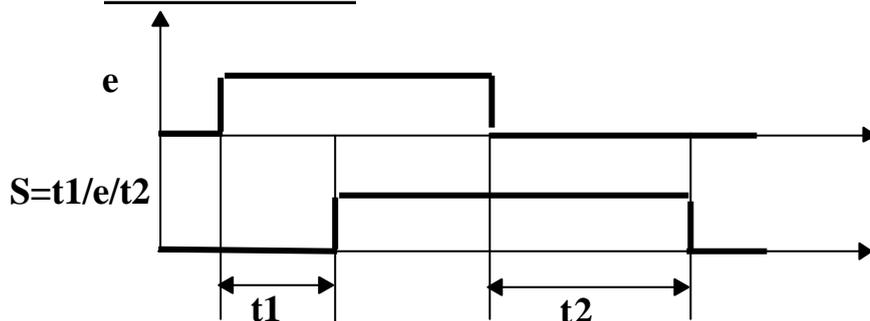
FONCTION RETARD OU TEMPORISATION

1. Symbole logique de la fonction retard ou temporisation

Notation	Symbole
$S = t1/e/t2$	<p style="text-align: center;">TP</p>

Désignations	Fonctions
e	Signal d'entrée tout ou rien(0,1) qui enclenche la tempo
t1	temps de retard par rapport au passage à 1 du signal d'entrée
t2	temps de retard par rapport au passage à 0 du signal d'entrée
$S = t1/e/t2$	Signal de sortie (fin de tempo)

2. Fonctionnement



Le début du signal de sortie **S** est retardé du temps **t1** par rapport au début du signal **e**.
La fin du signal de sortie **S** est retardé du temps **t2** par rapport à la fin du signal **e**.

REMARQUE : Dans la plupart des cas $t2 = 0$. Donc $S = t1/e$.

3. Réalisations et composants d'une temporisation

Lorsqu'un signal TOR d'entrée **e** est présent, un compteur s'incrémente à chaque impulsion d'une base de temps (en seconde, 1/10 s, 1/100 s,...).

Lorsque le contenu du compteur est égal à la valeur de consigne **t**, qui contient le temps de retard, le signal de sortie **S** passe à 1.

Composants :

- Signal d'entrée
- Compteur de temps
- Base de temps
- Consigne
- Signal de sortie, fin de temporisation

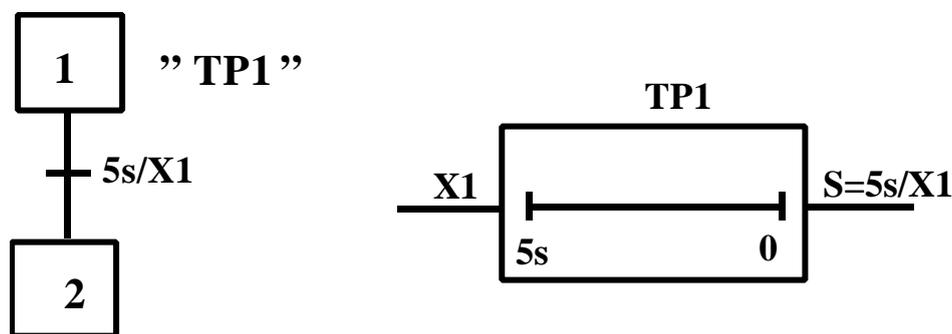
4. Représentation dans un Grafcet

Une temporisation dans un grafcet conditionne le franchissement de la transition donc l'activation de l'étape suivante, en fonction du temps d'activation de l'étape à laquelle elle est associée étape.

- Le signal d'entrée est l'étape.
- Le signal de sortie ou de fin temporisation est la réceptivité.

Un grafcet décrit le fonctionnement en fonction des entrées/sorties d'une frontière. La temporisation étant interne à cette frontière, elle ne doit pas apparaître en action dans le grafcet, mais seulement en commentaire. Elle sera donc définie par son logigramme.

Exemple :



5. Exercice

Sur une presse, pour des raisons de sécurité, l'opérateur doit commander sa descente à l'aide des deux mains, avec un écart entre les 2 appuis de 0.3 seconde maxi.



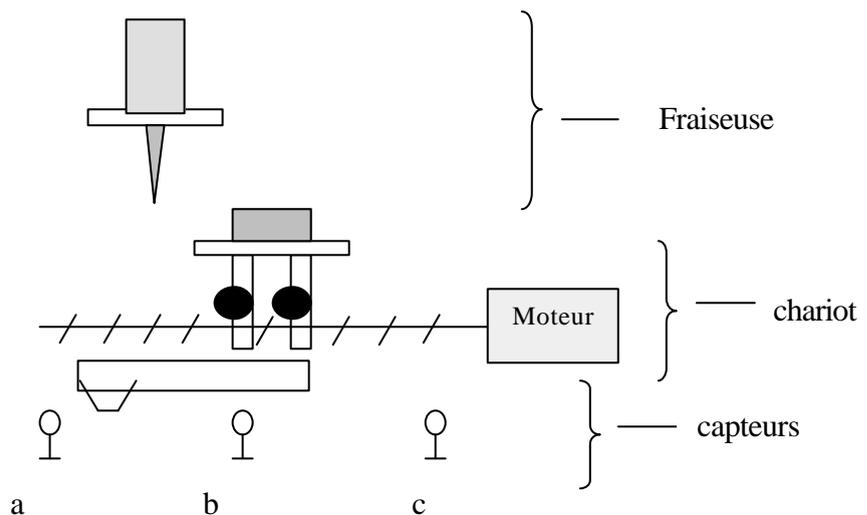
Tracer le grafcet décrivant le fonctionnement de cette commande bi-manuelle.

Outils de description du fonctionnement d'un automate – TP N° 12

Durée 3 h

1/ Présentation :

Le système dont vous disposez est la modélisation d'un chariot de fraiseuse, nous le symboliserons de la façon suivante :



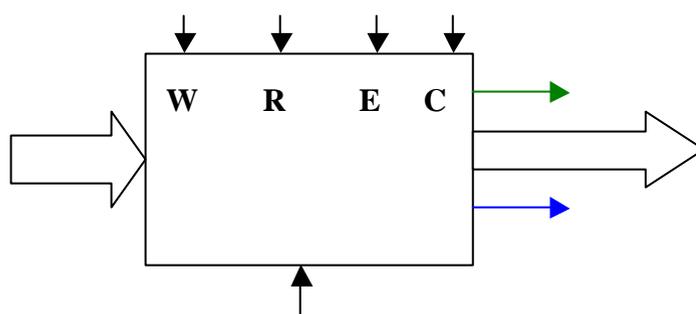
1.1/ Cahier des charges

Afin d'usiner une pièce, le chariot apporte la pièce vers la fraise en grande vitesse (GV) vers la gauche. Au contact du capteur b le déplacement de la pièce s'effectue en petite vitesse (PV). Une fois le capteur a actionné, le déplacement du chariot s'arrête et repart sur la droite en grande vitesse, jusqu'au contact de c.

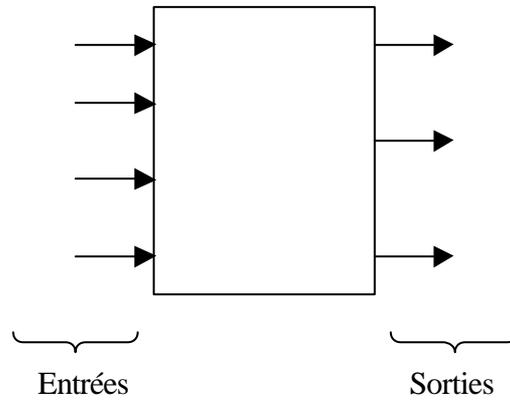
Ici la petite vitesse est nécessaire afin d'obtenir un usinage correct,
Ce cycle sera commandé par un automate programmable le TSX 17.

2/ Etude du système :

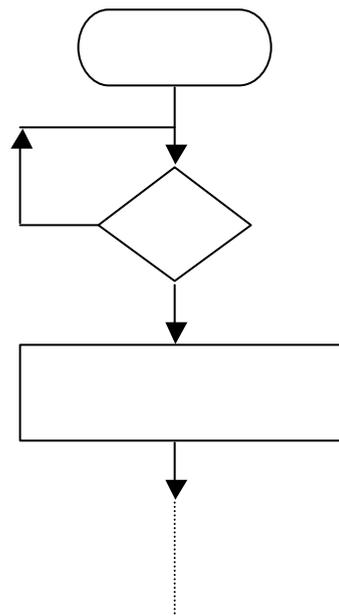
2.1/ La fonction globale du système : complétez l'actigramme (SADT)



2.2/ Déterminer les Entrées/Sorties

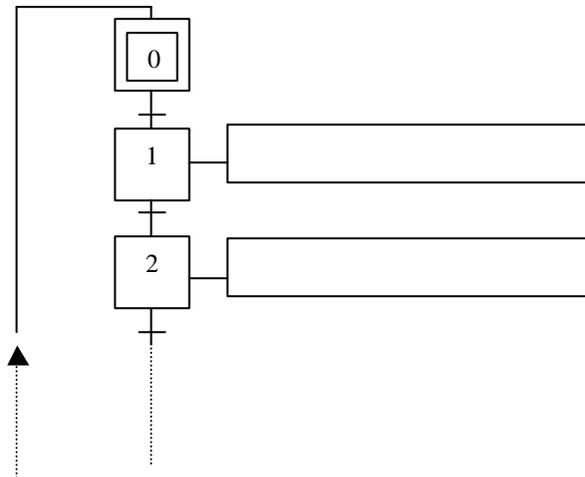


2.3/ Compléter l'organigramme

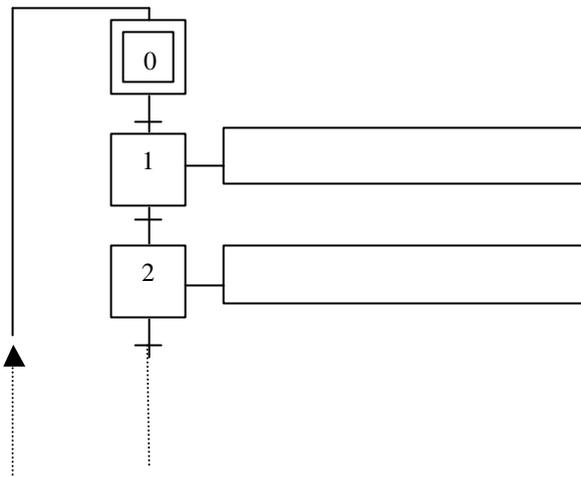


2.3/ Réalisation des grafjets du système

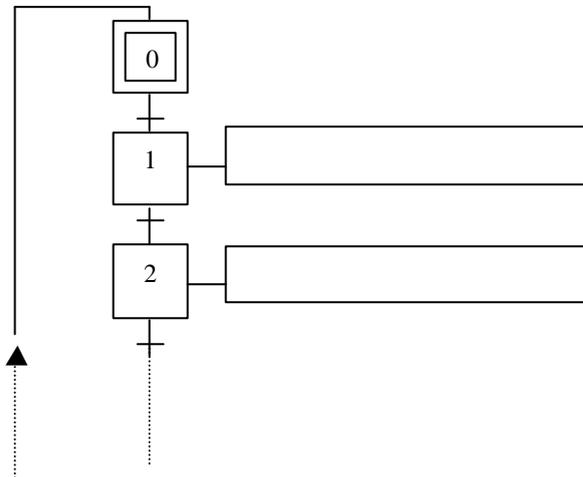
2.31/ Réaliser le grafjet du point de vue de la partie opérative (P.O)



2.31/ Réaliser le grafjet du point de vue de la partie commande (P.C)



2.31/ Réaliser le grafcet du point de vue de l'automate



2.4/ Mettre en œuvre le système avec l'automate TSX-17

- Ecrire le programme du TSX-17 sur l'Annexe 1.
- Tracer le schéma de commande du système automatisé (Annexe 1).
- Lancer l'essai après vérification de la programmation par l'enseignant.
- Vérifier la conformité du cycle par rapport au cahier des charges.

- **Dans ce travail nous nous conformerons aux définitions suivantes**

Entrées de l'automate :

I0,0 : Entrée → capteur c
 I0,1 : Entrée → capteur b.
 I0,2 : Entrée → du capteur a.
 I0,4 : dcy.
 I0,3 : Init

Sortie de l'automate :

O0,0 : Commande de droite.
 O0,1 : Commande de gauche.
 O0,3 : Commande de petite vitesse.

2.5/ Conclusion

- Le cahier des charges est-il respecté ?
- Enoncez les difficultés rencontrées.
- Dans quel domaine les organigrammes sont-ils plus utilisés ?

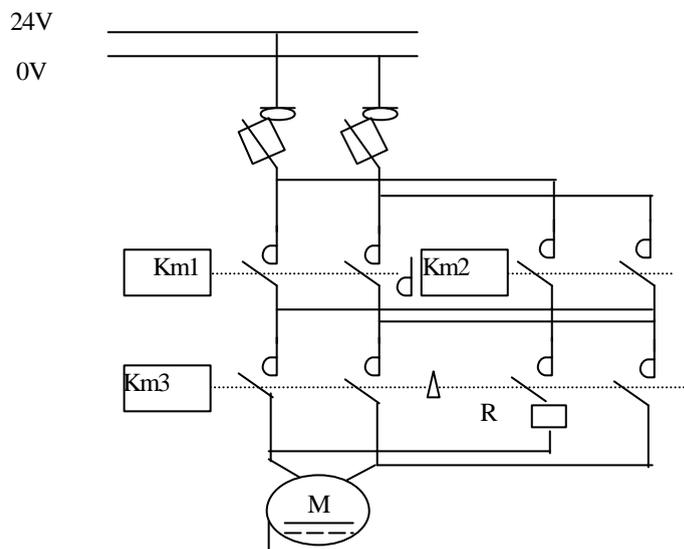
Annexe 1

1 Schéma de puissance :

Sur le système , le moteur utilisé est à courant continu.

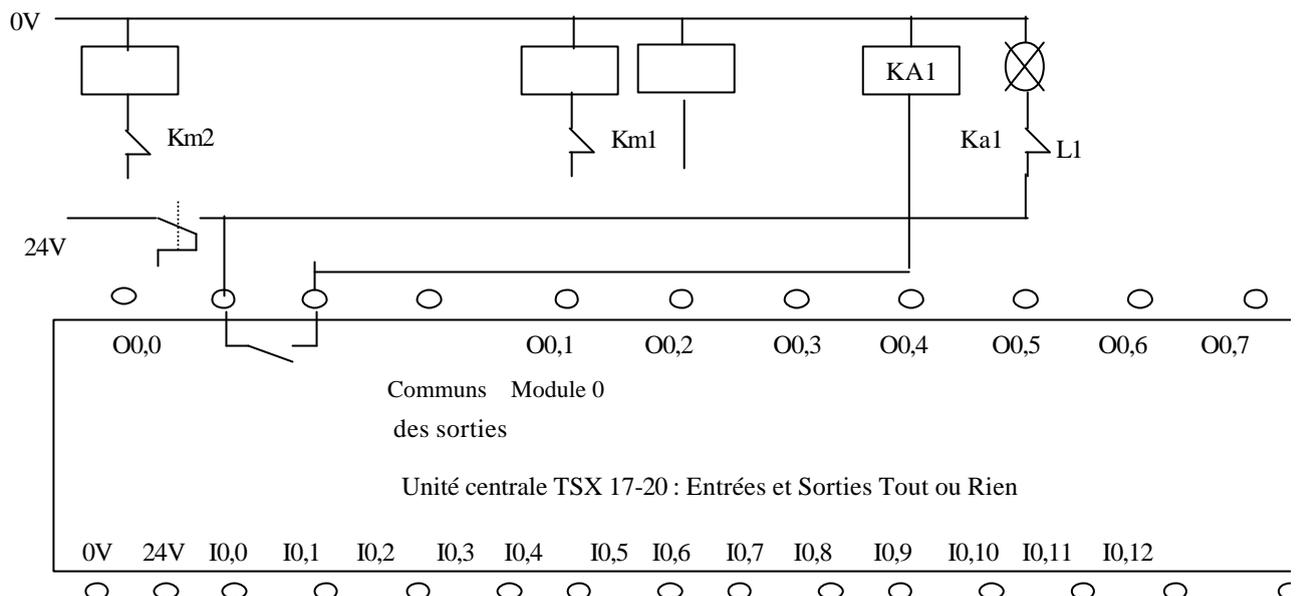
Il est alimenté avec une tension de 24 V, pour une puissance de 0,7 KW.

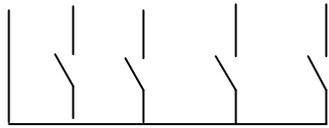
-Avec Km1 : Marche droite ;Km2 : Marche gauche ;Km3 : Marche petite vitesse.



Km3

3/ Schéma de câblage de la partie commande

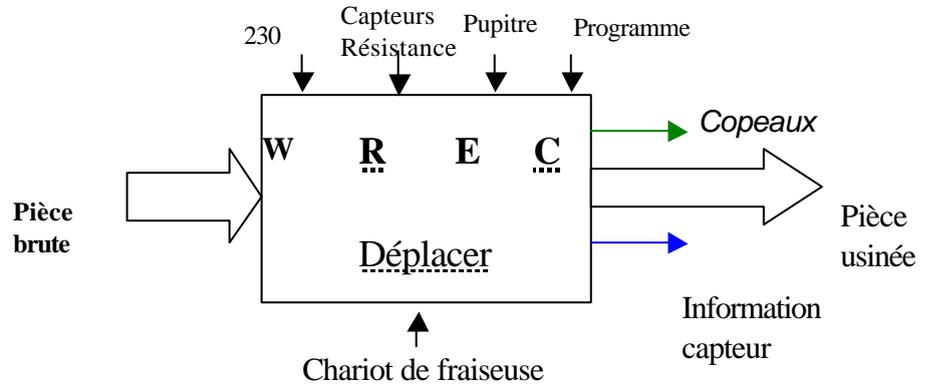




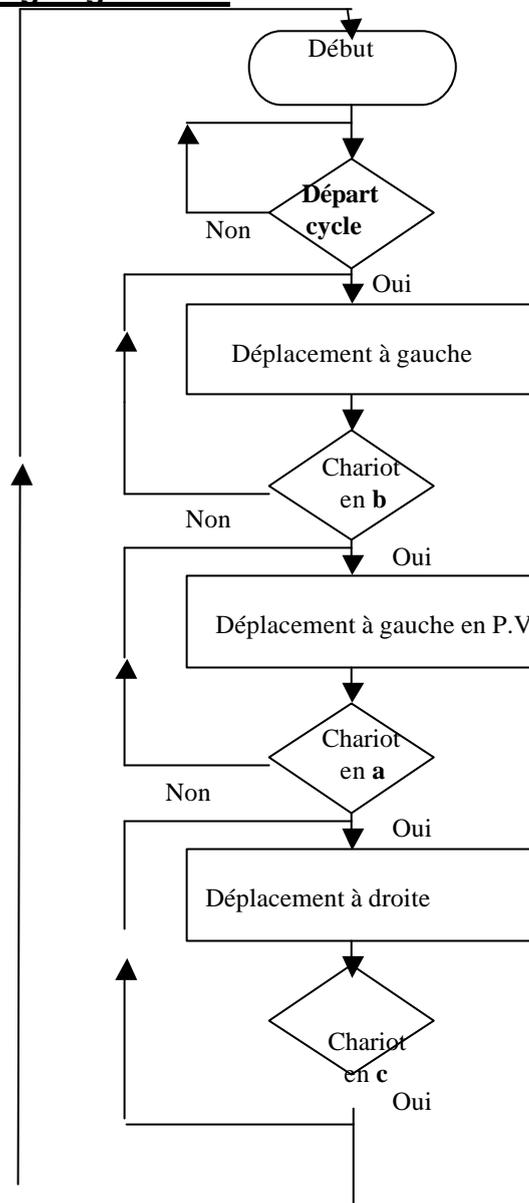
Correction

2/ Etude du système

2.1/ La fonction globale du système (SADT).

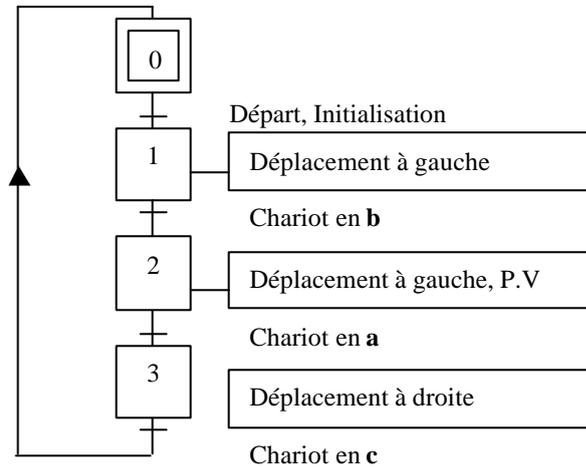


2.3/ Compléter l'organigramme :

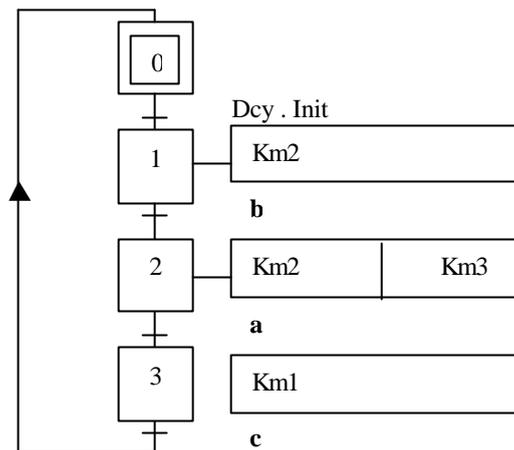


2.3/ Réalisation des grafquets du système

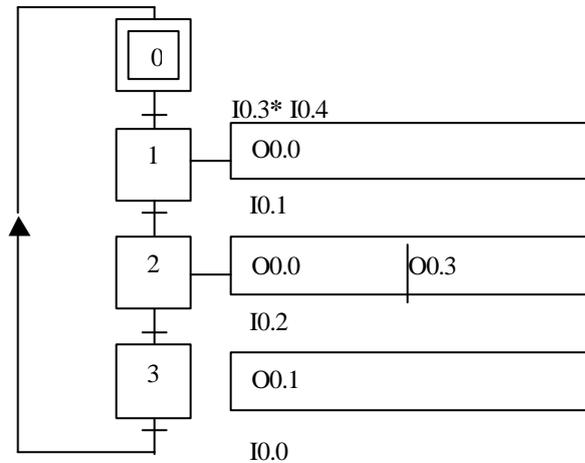
2.3.1/ Réaliser le grafcet du point de vue de la partie opérative (P.O)



2.3.2/ Réaliser le grafcet du point de vue de la partie opérative (P.C)



2.3.3/ Réaliser le grafcet du point de vue de la partie opérative (P.C)



Légende :

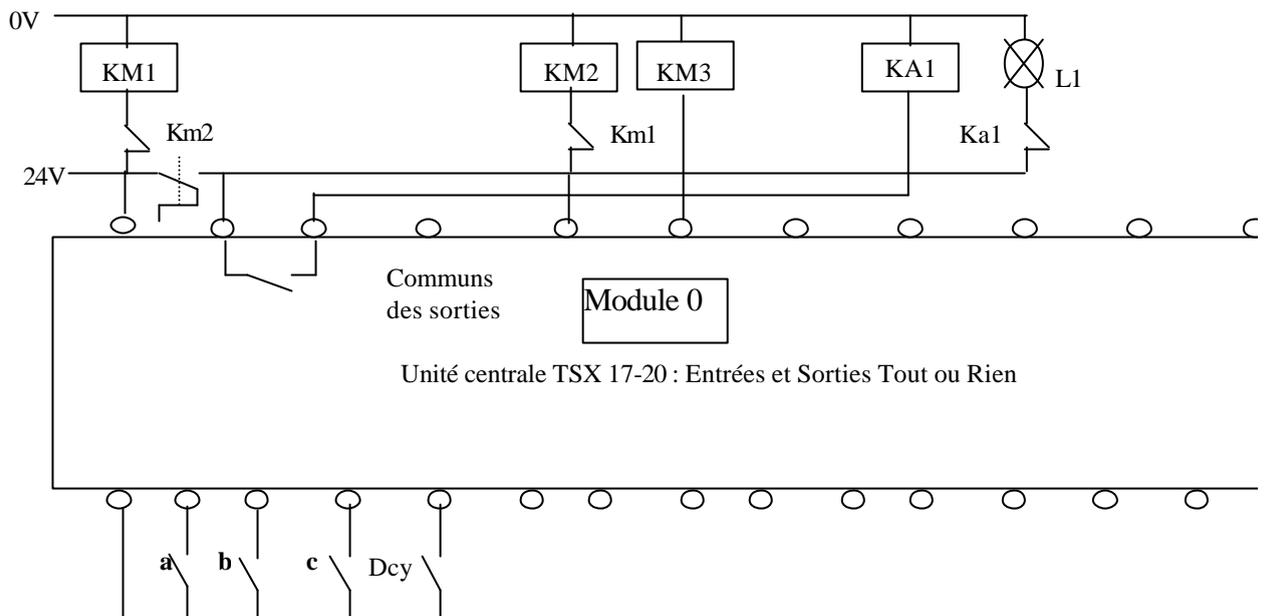
Entrées de l'automate :

- I0.0 : Entrée → capteur c
- I0.1: Entrée → capteur b.
- I0.2: Entrée → du capteur a.
- I0.4 : dcy.
- I0.3 : Init

Sortie de l'automate :

- O0.0 : Commande de Km2
- O0.1 : Commande de Km1
- O0.3 :: Commande de Km3

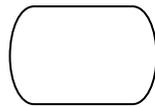
2.4/ Mettre en œuvre le système avec l'automate TSX-17



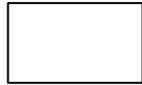
Dossier Ressource

Présentation générale de l'automate TSX-17

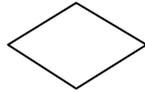
L'Organigramme



Terminaison (Départ, fin de cycle)



Procédé



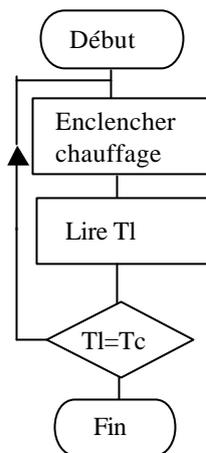
Décision (oui, non)

Exemples d'utilisations

Exemple 1 :

Un dispositif doit porter à une température de 75°C un liquide stocké dans une cuve, à la température ambiante de 20°C. Le traitement thermique est interrompu dès que la température du liquide atteint la température de consigne.

Organigramme



Langage algorithmique

Début algorithme : Chauffe

Variables

Temps consigne :Tc=75°C

Temps liquide :Tl

CH, chauffage 0 « non enclenché »

CH, chauffage 1 « enclenché »

Début action :

Répéter

CH 1, «enclencher chauffage», lire « Tl ».

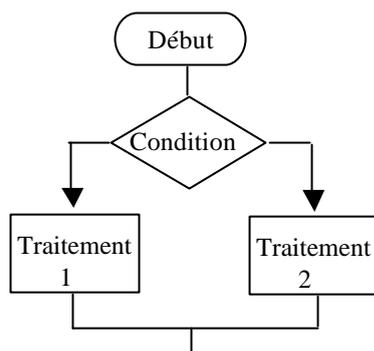
Jusqu'à « Tl=Tc »

Fin action

Fin algorithme

Exemple 2 :

Organigramme



Langage algorithmique

Début algorithme :

Structure alternative

Débu action

SI condition vraie

Alors faire « traitement 1 »

Sinon faire « traitement 2 »

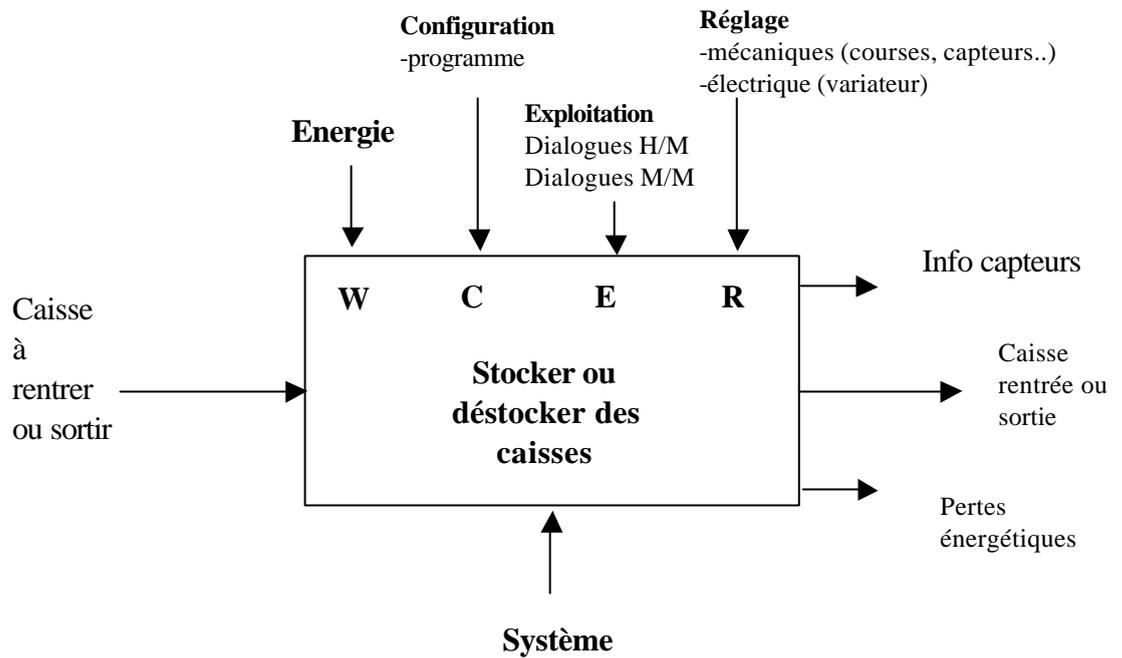
Fin si

Fin action

Fin algorithme.

Fin

L'Analyse fonctionnelle



Evaluation Formative :

Tâches à réaliser	Correct	Correct (avec aide)	A revoir	Non compris
Fonction globale du système				
Organigramme du système				
Graphe des entrées / Sorties				
Grafcet pt vu PO				
Grafcet pt vu PC				
Grafcet pt vu Automate				
Mise en œuvre Du programme				

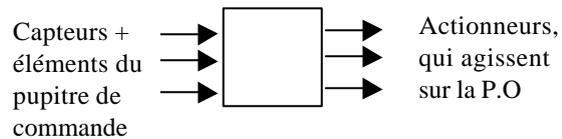
Appréciation
globale :

Remarque :

Fiche de synthèse

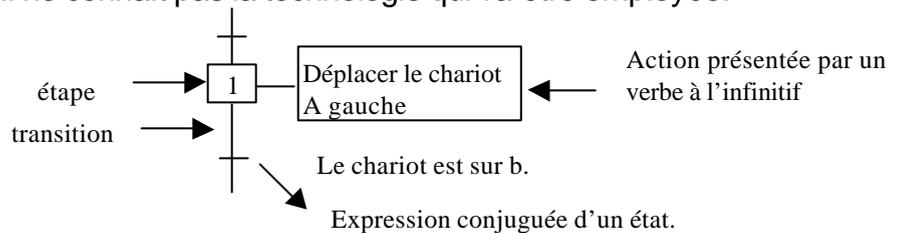
Les Organigrammes : sont d'avantage utilisés au niveau de l'informatique.

Le graphe des entrées /Sorties :



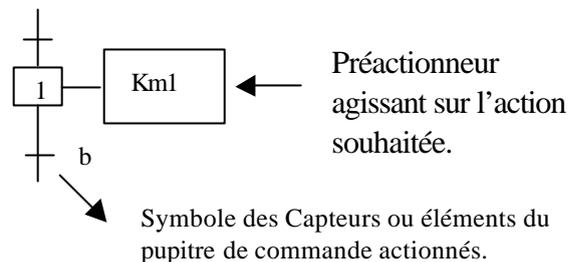
Grafcet du point de vue P.O :

Ici nous sommes à la place d'un industriel qui souhaite un système automatisé avec un fonctionnement précis. Il ne connaît pas la technologie qui va être employée.

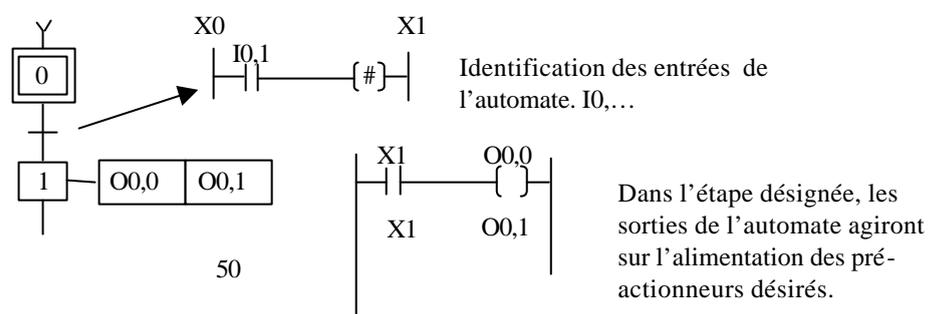


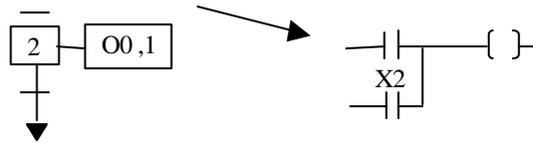
Grafcet du point de vue PC :

Dans ce cas nous sommes le concepteur qui connaît la technologie du système, pouvant définir les éléments à commander.



Grafcet du point de vue automate :



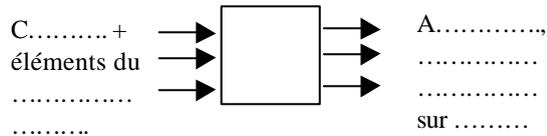


Programmation : En traitement Postérieur chaque sortie est nommée une seule fois.

Fiche de synthèse

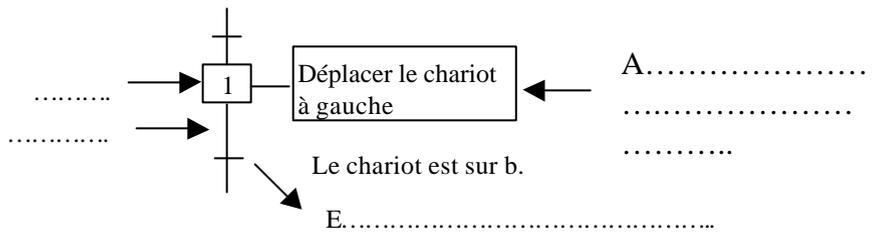
Les Organigrammes : sont d'avantage utilisés au niveau de l'informatique.

Le graphe des entrées /Sorties :



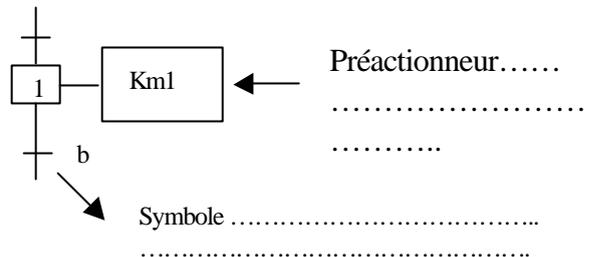
Grafcet du point de vue P.O :

Ici nous sommes à la place

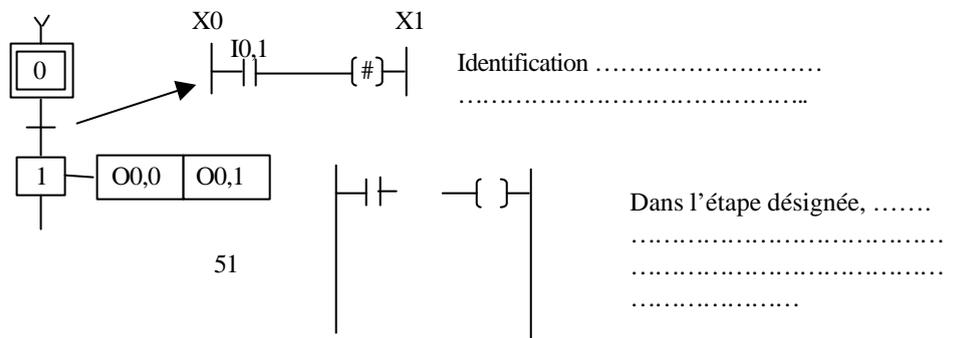


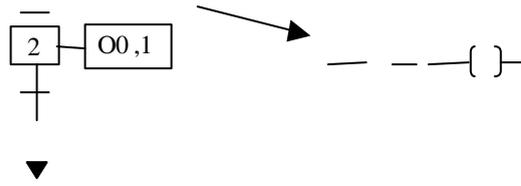
Grafcet du point de vue PC :

Dans ce cas nous sommes



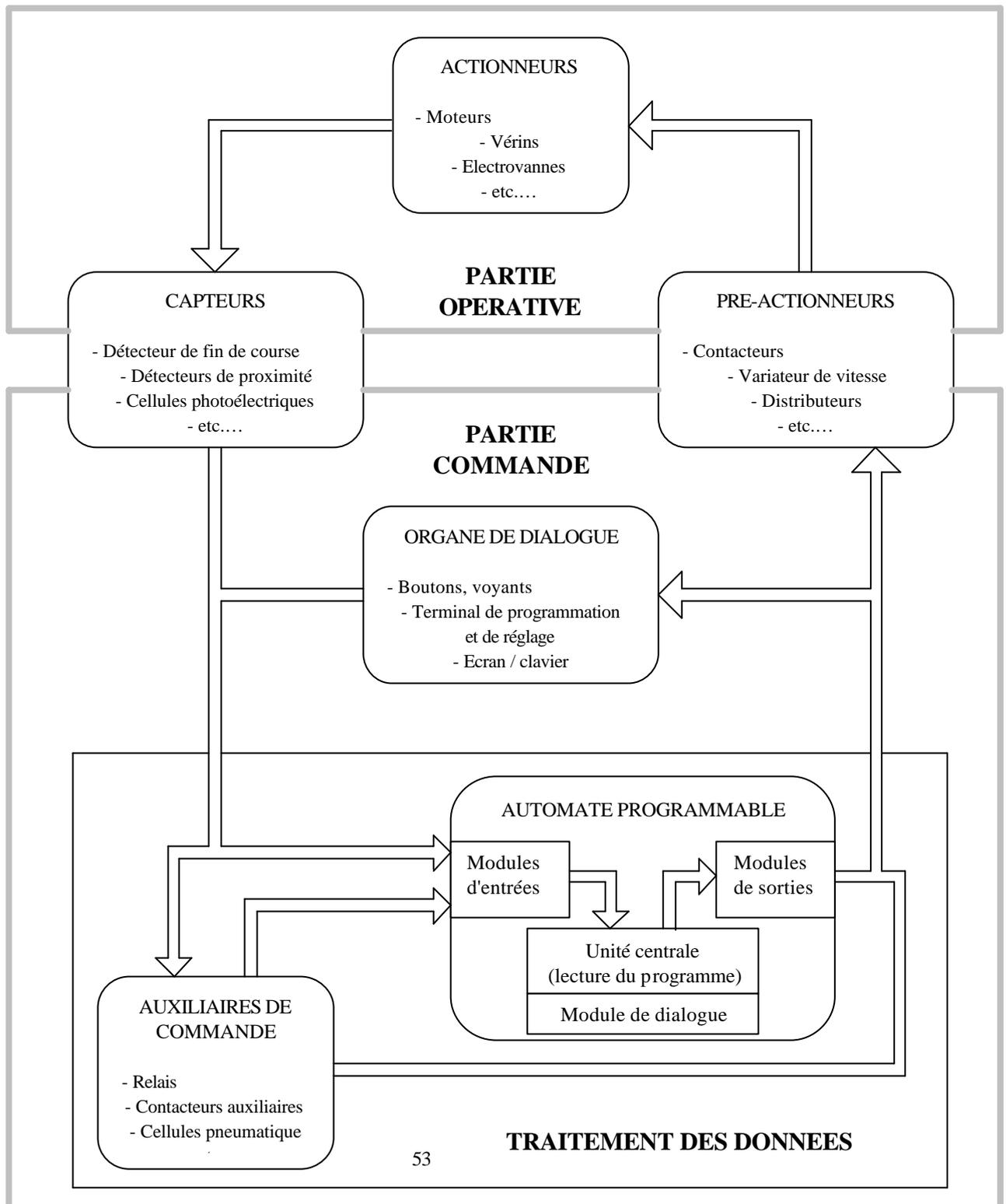
Grafcet du point de vue automate :



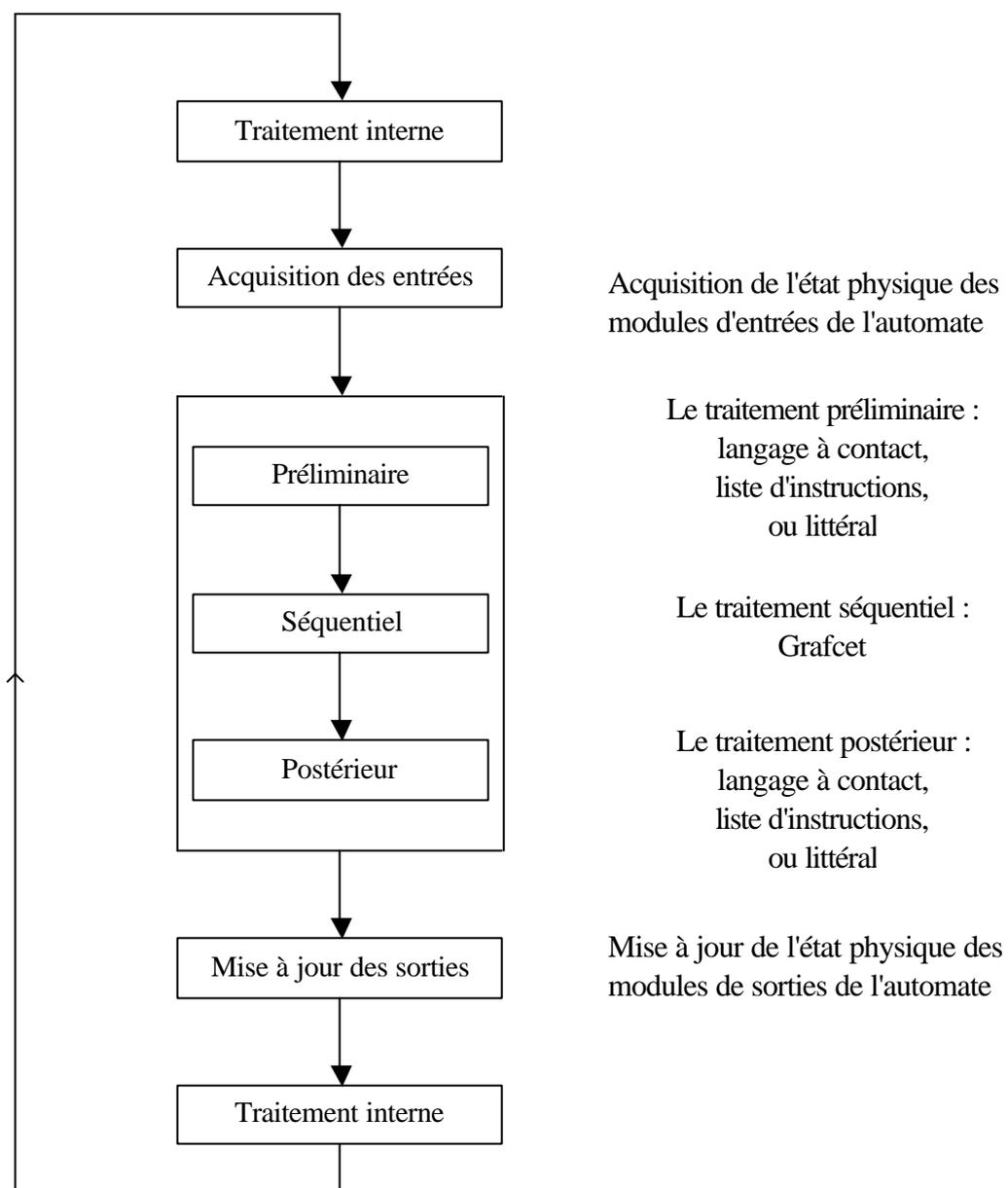


Programmation : En traitement Postérieur chaque sortie est nommée une seule fois.

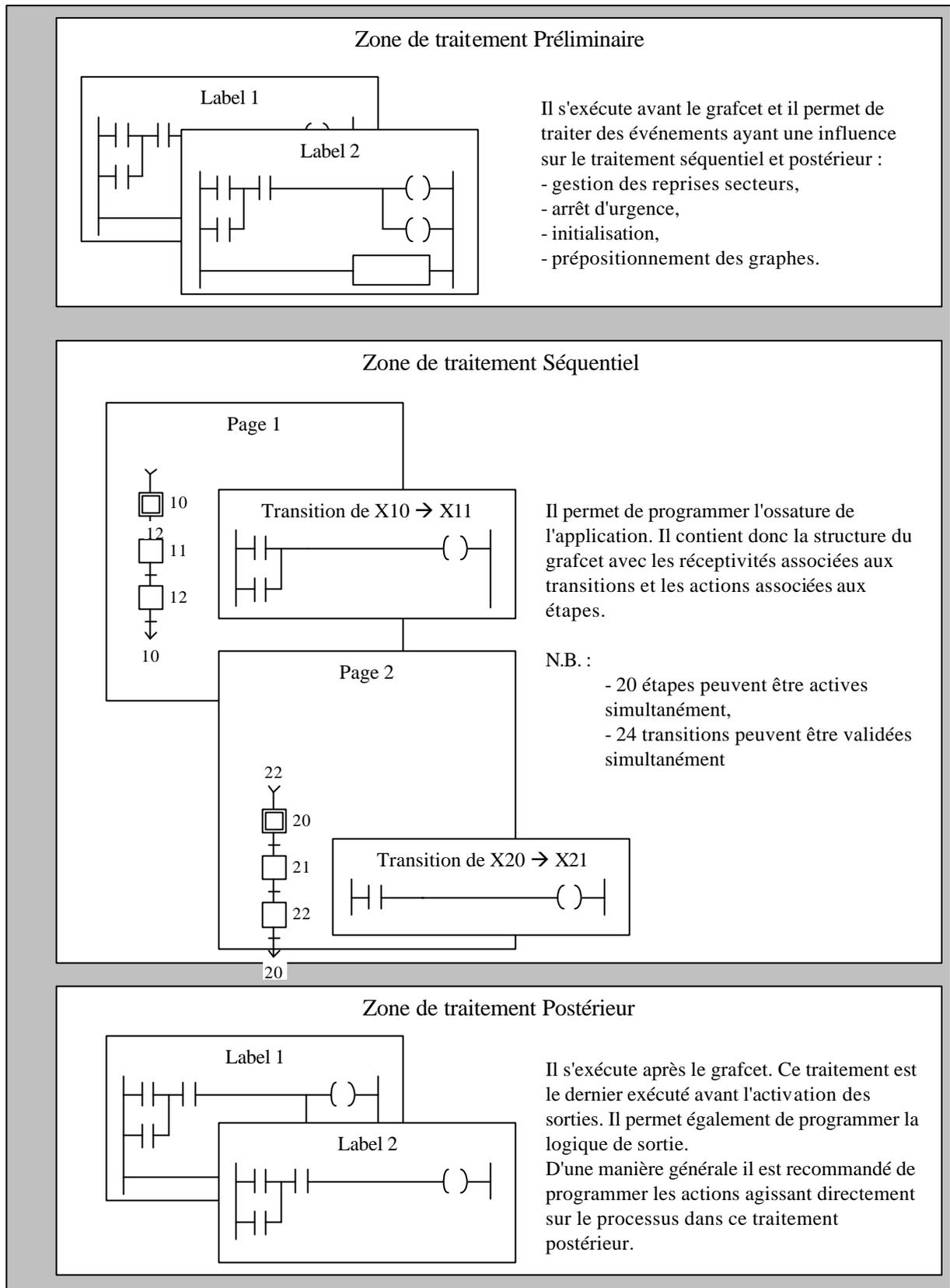
STRUCTURE D'UN AUTOMATISME PILOTE PAR UN A.P.I.



PRINCIPE DE FONCTIONNEMENT DU TSX37 AVEC UN PROGRAMME GRAFCET



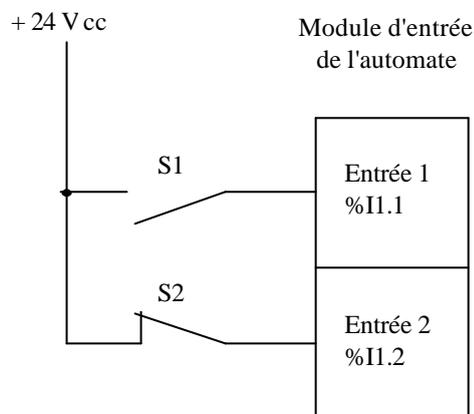
STRUCTURE D'UN PROGRAMME GRAFCET



INTERPRETATION D'UN LANGAGE DE PROGRAMMATION EN LADDER (TRAITEMENT PRELIMINAIRE, TRANSITION DU TRAITEMENT SEQUENTIEL, TRAITEMENT POSTERIEUR)

Correspondance entre la continuité électrique d'un contact et le niveau de l'entrée associée.

Exemple : 1 contact à fermeture (S1) et 1 contact à ouverture (S2) alimentés en 24 Vcc reliés à un module d'entrées d'automate programmable.

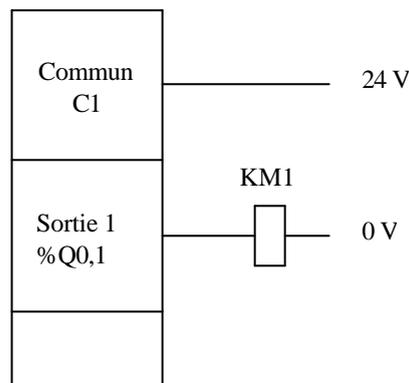


Représentation de chaque contact électromécanique	Niveaux logiques Des entrées associées	Symboles graphiques de programmation	
		Test de l'état du bit de l'entrée Associée au contact	Test de l'état inverse du bit de L'entrée associée au contact
 S1 "Actionné" "Repos"	"1"	Résultat du test : "1"	Résultat du test : "0"
	"0"	Résultat du test : "0"	Résultat du test : "1"
 S2 "Actionné" "Repos"	"0"	Résultat du test : "0"	Résultat du test : "1"
	"1"	Résultat du test : "1"	Résultat du test : "0"

Correspondance entre le niveau logique de la sortie et la continuité de la bobine associée.

Exemple : 1 bobine de contacteur (KM1) alimentée en 24 Vcc pilotée par une sortie relais d'automate.

Module de sorties
de l'automate



Résultat de la fonction logique	Symboles graphiques de programmation		
		Transfert du résultat de la fonction logique dans le bit de la sortie associée à la bobine	
"0"	Résultat du transf : "0"	Résultat du transf : "1"	
"1"	Résultat du transf : "1"	Résultat du transf : "0"	

Etat de la bobine KM1 :
 - "Activée" si le résultat du transfert est "1"
 - "Repos" si le résultat du transfert est "0"

LE PL7 MICRO

Le PL7 Micro est un logiciel sous environnement Windows

Il se lance par un "double clic" sur le fichier "*.stx" contenant l'application ou par l'intermédiaire du menu démarrer, programme, modicon télémechanique, PL7 micro Vx.x.

1) Nouvelle application :

Pour une nouvelle application, dans le menu principal, faire :

Fichier → Nouveau.

La fenêtre ci dessous s'ouvre.



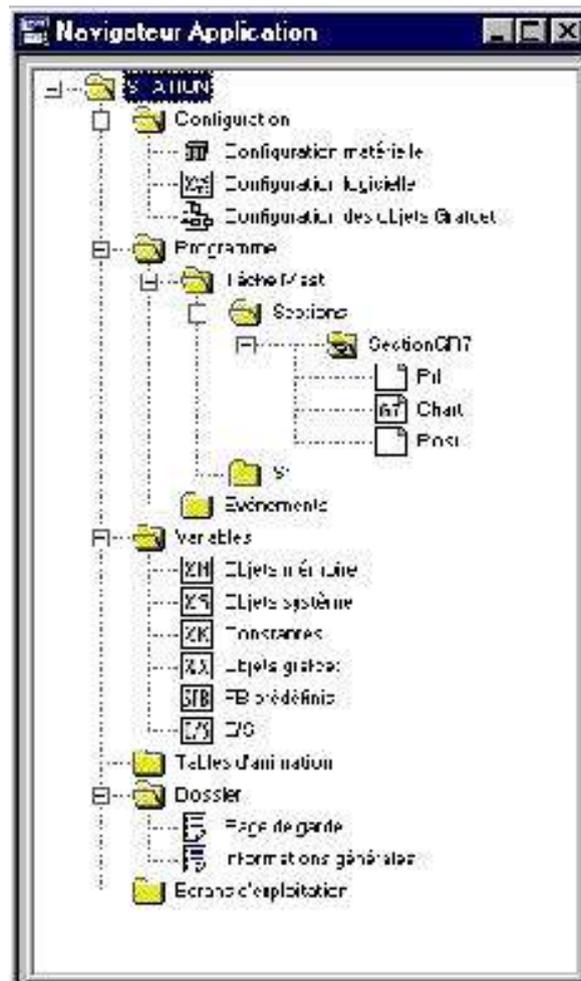
Choisir l'automate utilisé par exemple TSX3721 V3.0 et choisir la programmation Grafset.

Valider votre choix par l'appui sur le bouton "OK".

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

2) Application existante:

Après un "Double clic" sur le fichier contenant l'application "*.stx", ou par l'intermédiaire de "Fichier → Ouvrir" ou encore "AP → Connecter", la fenêtre "Navigateur Application" s'ouvre.



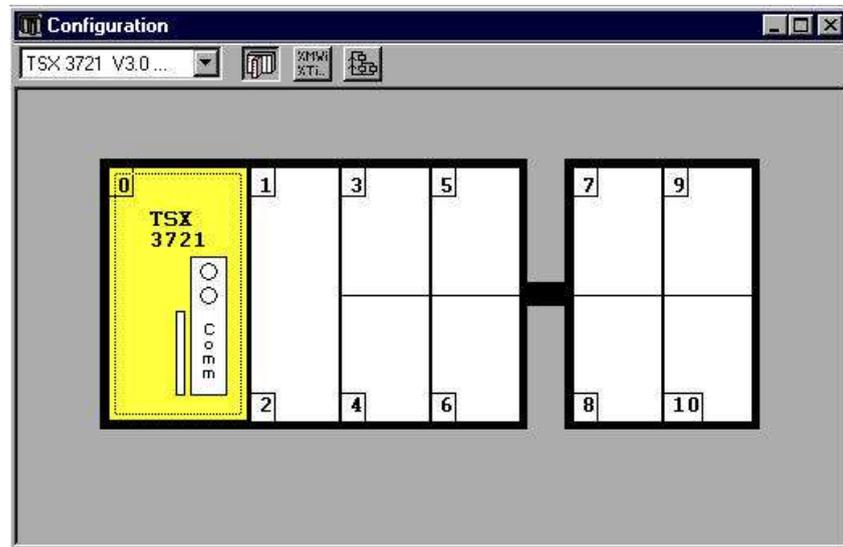
Cette fenêtre est de la forme "Explorateur" et contient toutes les informations relatives à l'application, c'est à dire la configuration de l'automate, son programme, ses variables, les tables d'animations,...

Un simple "Double clic" ou "Clic droit" sur le dossier choisi par le programmeur, permet d'afficher un menu, d'ouvrir le dossier correspondant, d'obtenir certaines propriétés.

Dans la tâche maître, nous retrouvons les trois traitements principaux :

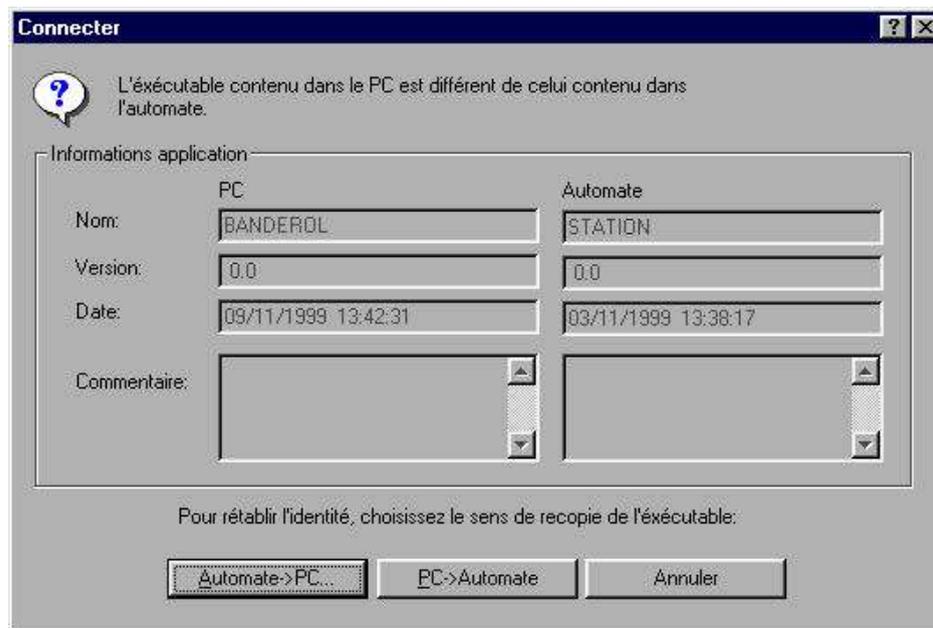
Pré	traitement préliminaire,
Chart	traitement séquentiel.
Post	traitement postérieur.

Après avoir sélectionné la configuration matérielle et un "clic droit", nous avons la possibilité d'ouvrir la configuration. Nous retrouvons la fenêtre ci-dessous :



Un double clic sur le module correspondant permet la configuration de ce dernier ou encore la mise au point en mode connecté (connexion établie entre l'automate et l'ordinateur de travail).

Lorsque l'utilisateur fait la commande "AP → Connecter" et que le fichier implanté dans la mémoire de l'ordinateur PC est différent de celui dans la mémoire de l'automate, la fenêtre ci-dessous s'ouvre.

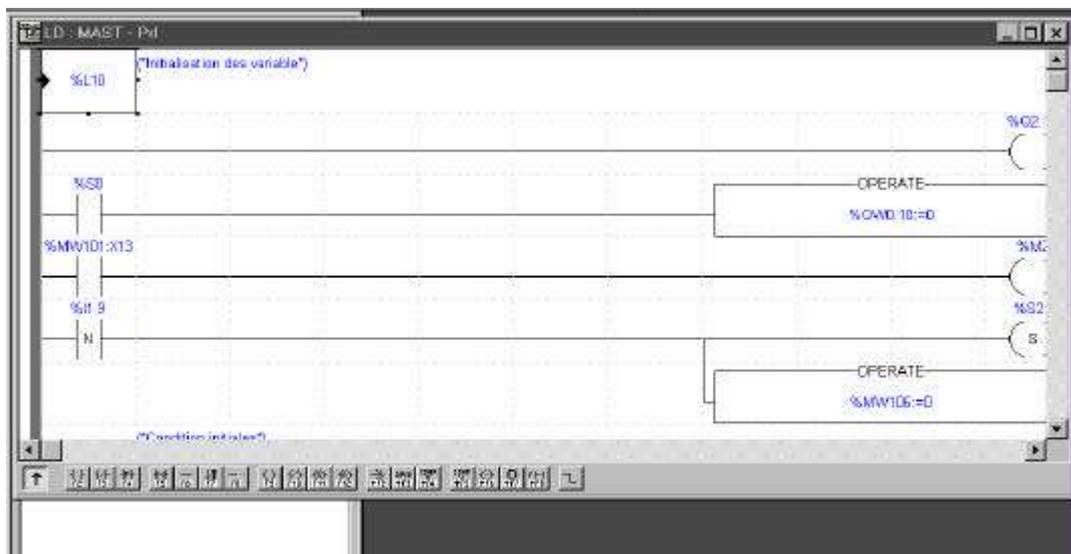
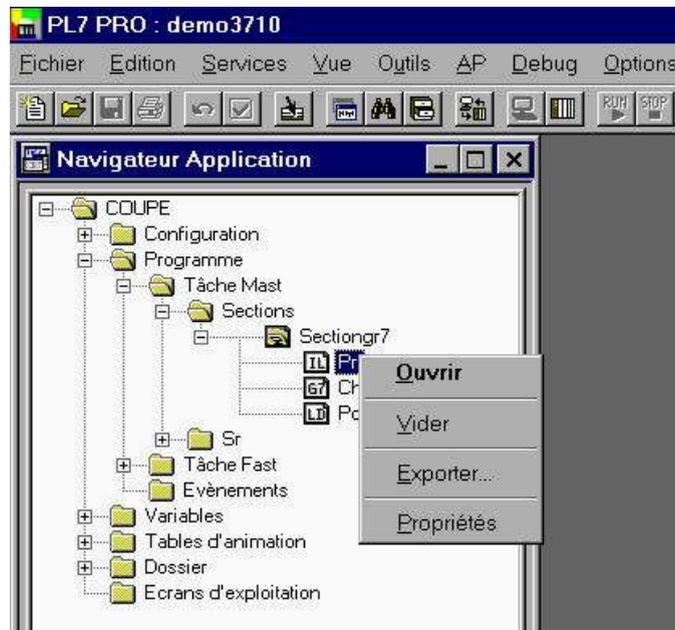


Le bouton "Automate → PC" permet de transférer le programme contenu dans la mémoire de l'automate vers l'ordinateur PC.

Le bouton "PC → Automate" permet de transférer le programme de contenu dans la mémoire de l'ordinateur PC vers la mémoire de l'automate. Lors de cette commande, l'automate doit être en "Stop". Ne pas oublier de le remettre en "Run" une fois cette opération terminée.

Traitement préliminaire

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



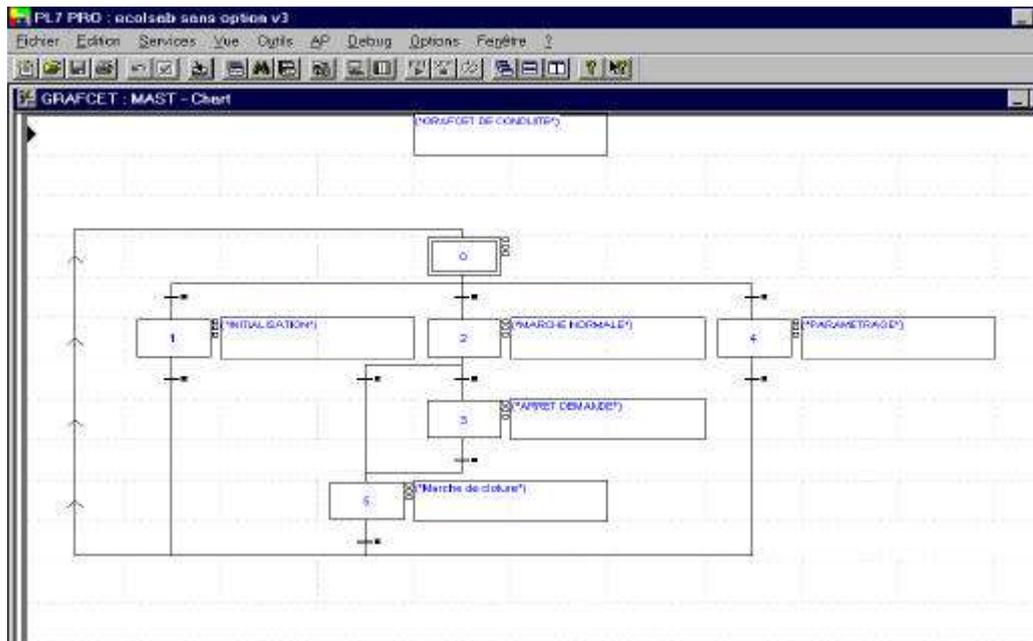
Pour faire une modification "double-cliquer" sur le RUNG. Il passe alors en rouge. Les symboles en LADDER apparaissent en bas de l'écran. Ils sont accessibles par les touches fonctions (F1 à F12). Une fois les modifications terminées, valider le RUNG par un appui sur la touche "Entrée". Il repasse en noir.

Les modifications sont possibles en mode connecté.

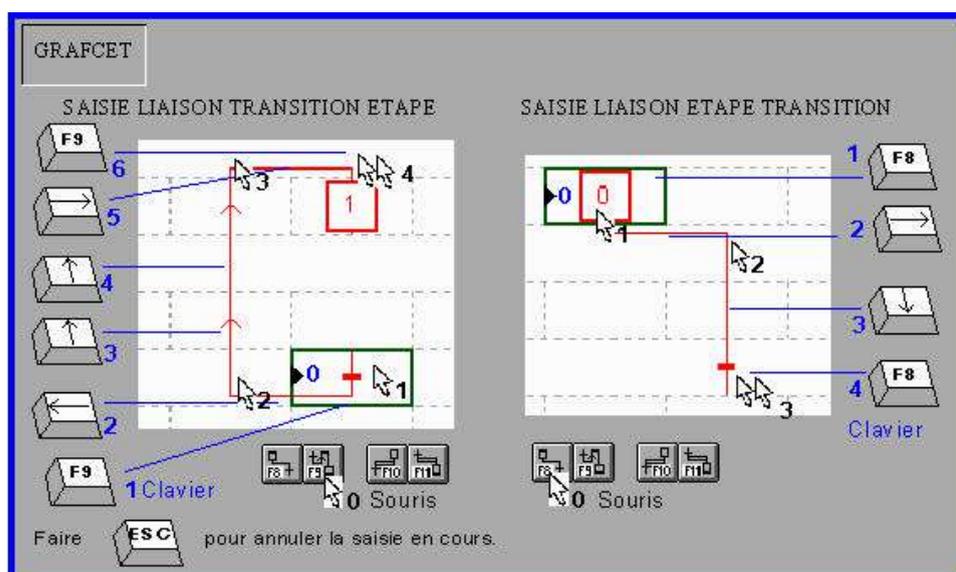
Traitement séquentiel

1) Modifier un grafcet

Ouvrir les pages grafcet en "double-cliquant" sur "GR7".

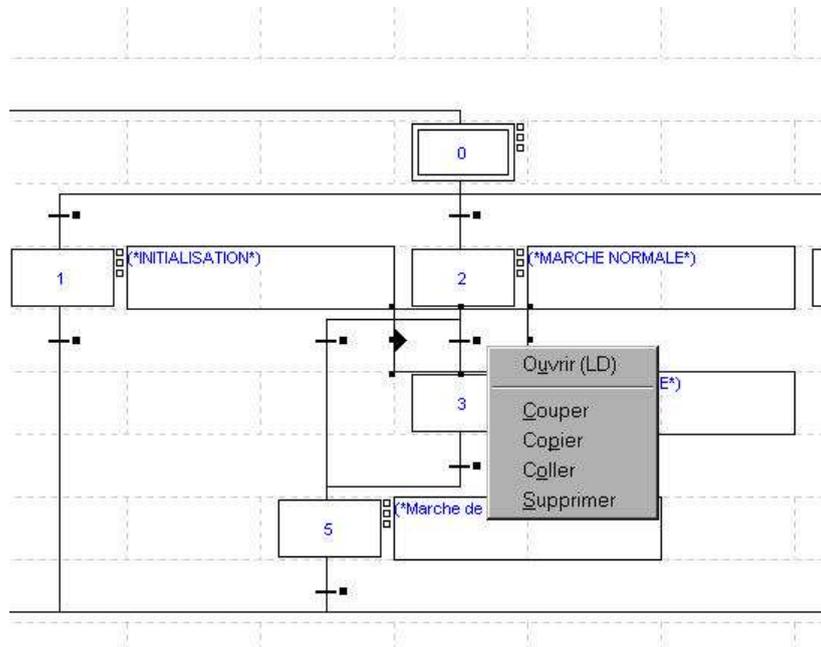


Les symboles de construction apparaissent en bas de l'écran. Ils sont accessibles par les touches fonctions (F1 à F12). Positionner le curseur sur le grafcet à modifier, "double-cliquer", le grafcet devient rouge. Effectuer la modification et valider par la touche "Entrée". Le grafcet redevient noir.



2) Modifier une transition

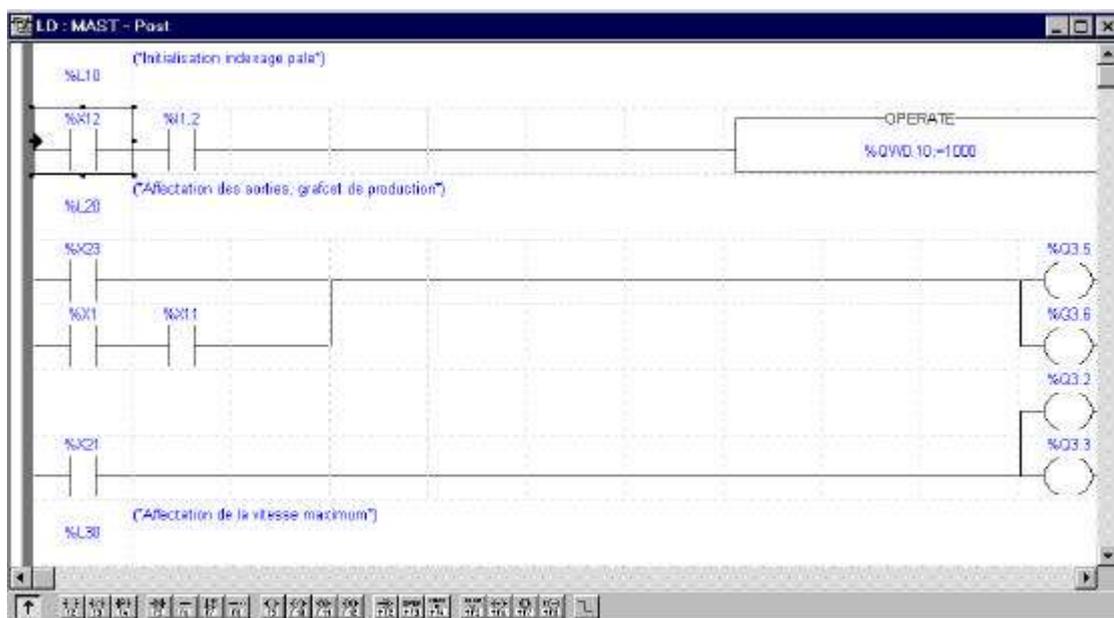
Positionner le curseur sur la transition à modifier, cliquer à droite et choisir "Ouvrir".



Les transitions peuvent être programmées en Ladder (LD), en structuré (ST), ou en instruction list (IL). Les symboles Ladder apparaissent de nouveau en bas de l'écran. Ils sont également accessibles par les touches fonctions (F1 à F12).

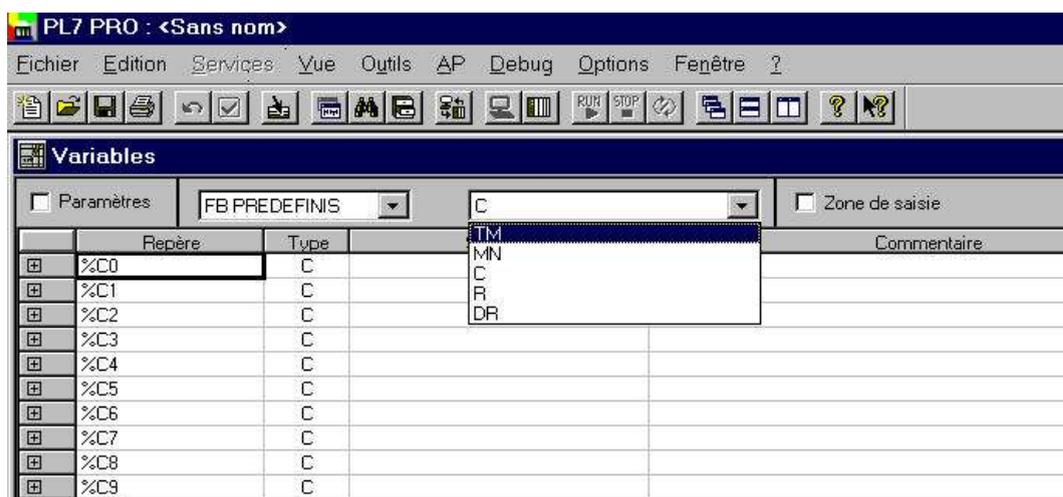
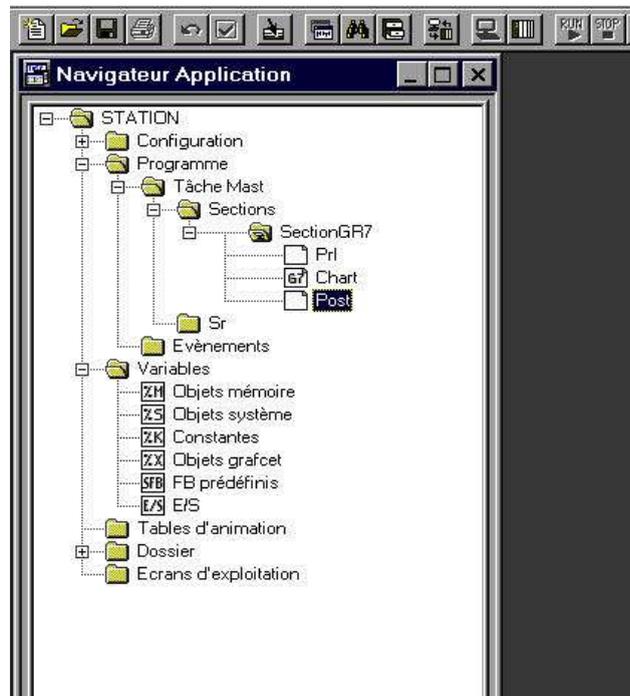
Traitement postérieur

Pour ouvrir les pages du traitement postérieur, "double-cliquer" sur "POST". Le postérieur peut être programmée en Ladder (LD), en structuré (ST), ou en instruction list (IL). Les symboles Ladder apparaissent en bas de l'écran et sont accessibles par les touches fonctions (F1 à F12).



Paramétrage d'un bloc fonction

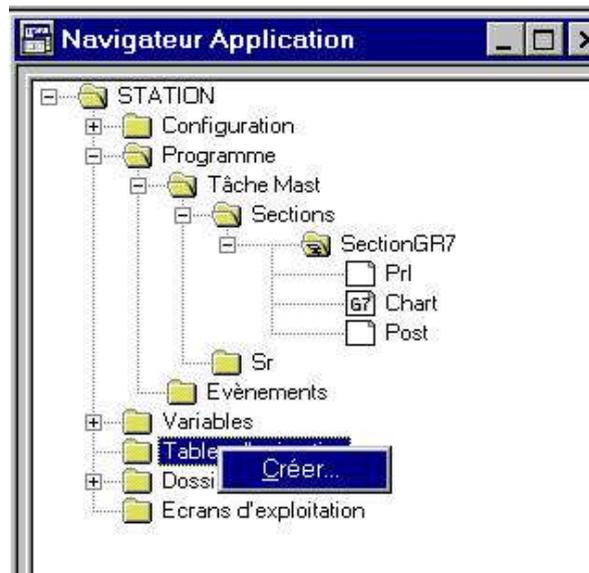
Pour modifier les paramètres d'un bloc fonction (tempo, compteur,...), cliquer sur « FB prédéfinis » dans dossier variables, puis choisir le type de fonction dans le menu déroulant.



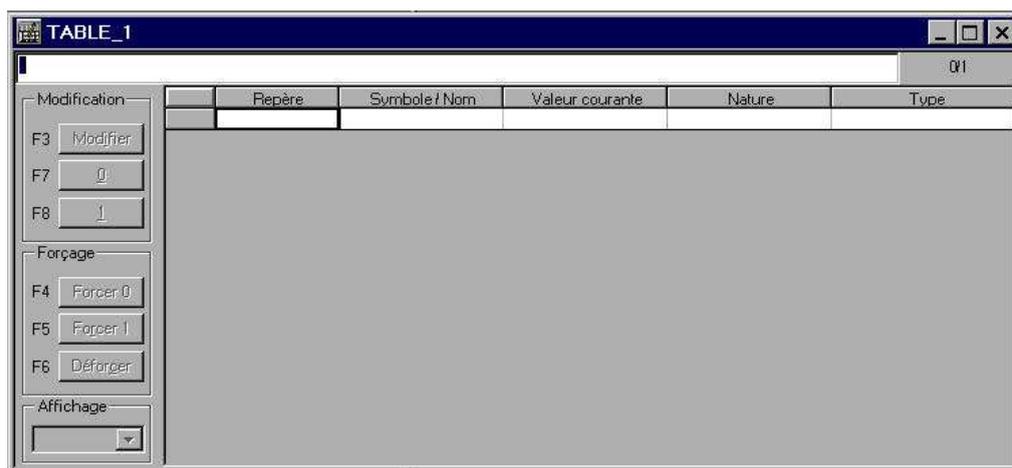
Il est possible de modifier la valeur de présélection en mode connecté, mais pour qu'il soit pris en compte il faut passer l'automate en "STOP", faire une initialisation "INIT", puis repasser l'automate en "RUN". Une fois la modification terminée valider et fermer la fenêtre.

Utilisation d'une table d'animation

Pour visualiser l'état d'une ou plusieurs variables en mode connecté, il suffit de créer une table d'animation. Pour cela cliquer à droite sur "table d'animation", et choisir "créer".



Choisir ensuite le nom des variables que vous désirez visualiser. Il est possible de sauvegarder cette table lors de la fermeture de la fenêtre.



A l'aide des touches fonction (F3 à F8), nous avons la possibilité de forcer ou de modifier ces variables.

STATION

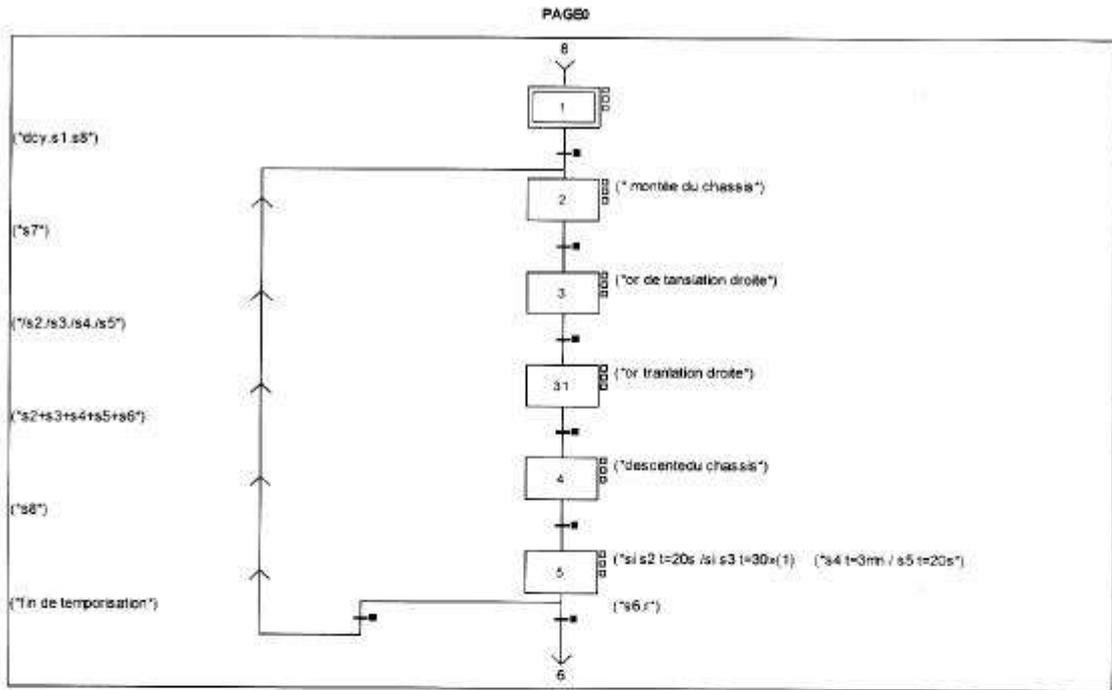
DOSSIER TECHNIQUE

Application :	EXERCICES D'APPLICATION.STX
Concepteur :	ARROUD
Version logiciel :	PL7 V3.3
Projet :	TRETEMENT DE SURFACE
Version application :	0.0
Date de dernière modification :	26/02/2005 11: 22: 52
Automate cible :	TSX 3722 Checksum: 1593 B

STRUCTURE APPLICATION

Tâche	Section	Module	Langage
MAST	SECTIONGR7	PRL CHART CHART - PAGE0 %X5->%X2 CHART - PAGE0 %X1->%X2 CHART - PAGE0 %X2->%X3 CHART - PAGE0 %X3->%X31 CHART - PAGE0 %X31->%X4 CHART - PAGE0 %X4->%X5 CHART - PAGE0 %X5->%X6 CHART - PAGE1 %X6->%X7 CHART - PAGE1 %X7->%X8 CHART - PAGE1 %X8->%X1 POST	LANGAGE À CONTACTS (LD) GRAFCET LANGAGE À CONTACTS (LD) LANGAGE À CONTACTS (LD)

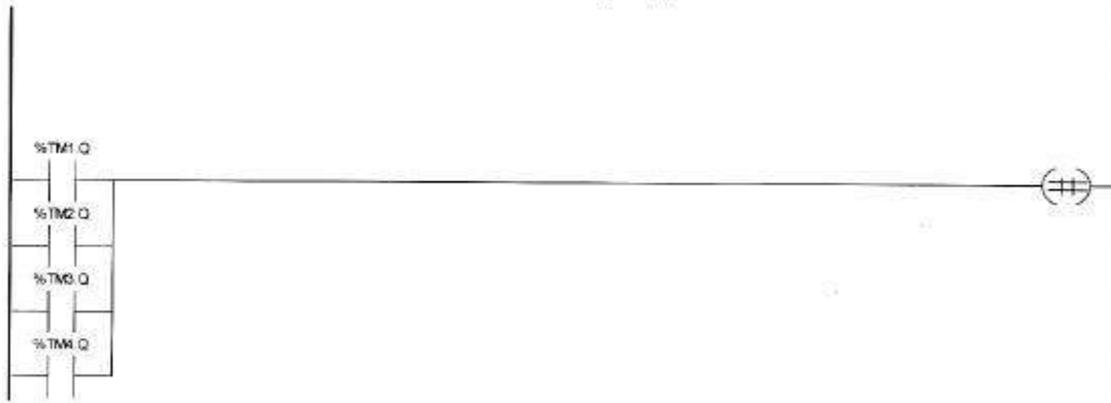
CHART - PAGE 0



Liste des renvois du commentaire de la page :

(1) : (*si s2 t=20s /si s3 t=30s si s4 t=3mm /si s5 t=20s*)

CHART - PAGE0 %X(5)->%X(2)



Liste de Variables utilisées dans le rung :

REPERE
%TM1.Q
%TM2.Q
%TM3.Q
%TM4.Q

SYMBOLE

COMMENTAIRE

MAST-PRL

Condition de validation : Aucune
Commentaire :



Liste de Variables utilisées dans le rung :

REPERE
%S0
%S21
%S1

SYMBOLE
S0
S21
On

COMMENTAIRE
demarrage a froid
initialisation du grafcet
demarrage a chau

CHART - PAGE0

CHART - PAGE0 %X(1)->%X(2)



Liste de Variables utilisées dans le rung :

REPÈRE
%I.0
%I.1
%I.8

SYMBÔLE
Dcy
Fc1
Fc8

COMMENTAIRE
depart cycle
fin de course %s1
fin de course s8

CHART - PAGE0 %X(2)->%X(3)



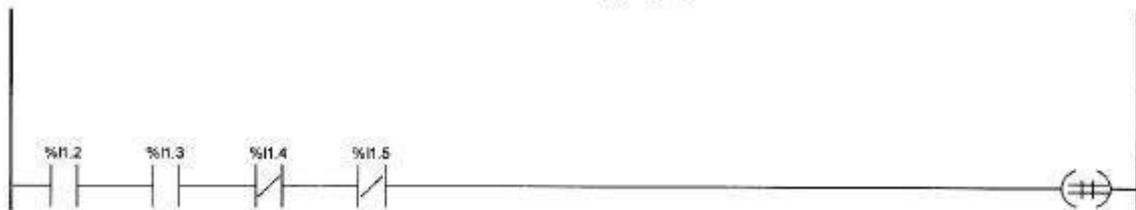
Liste de Variables utilisées dans le rung :

REPÈRE
%I.7

SYMBÔLE
Fc7

COMMENTAIRE
fin de course s7

CHART - PAGE0 %X(3)->%X(31)



Liste de Variables utilisées dans le rung :

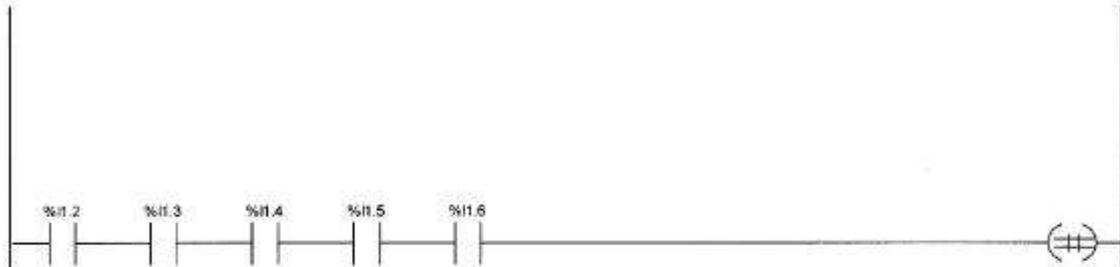
REPÈRE
%I.2
%I.3
%I.4
%I.5

SYMBÔLE
Fc2
Fc3
Fc4
Fc5

COMMENTAIRE
fin de course s2
fin de course s3
fin de course s4
fin de course s5

CHART - PAGE0

CHART - PAGE0 %X(31)->%X(4)



Liste de Variables utilisées dans le rung :

REFERE	SYMBOLE	COMMENTAIRE
%I1.2	Fc2	fin de course s2
%I1.3	Fc3	fin de course s3
%I1.4	Fc4	fin de course s4
%I1.5	Fc5	fin de course s5
%I1.6	Fc6	fin de course s6

CHART - PAGE0 %X(4)->%X(5)

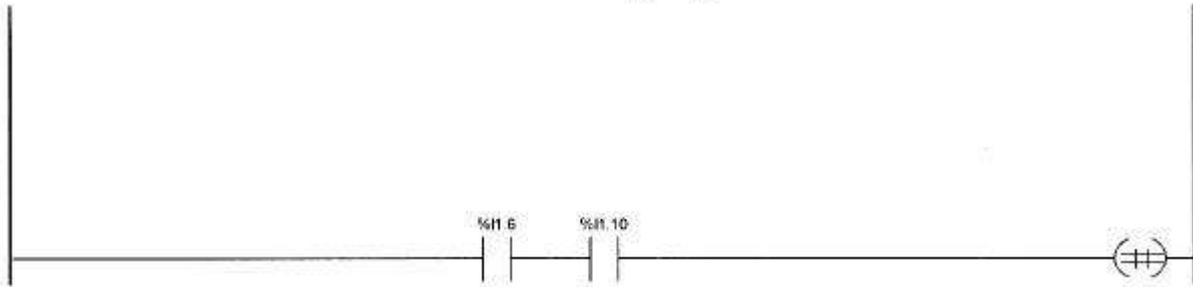


Liste de Variables utilisées dans le rung :

REFERE	SYMBOLE	COMMENTAIRE
%I1.8	Fc8	fin de course s8

CHART - PAGE0

CHART - PAGE0 %X(5)->%X(6)



Liste de Variables utilisées dans le LAD.

REPERE
%I1.6
%I1.10

SYMBOLE
Fc6
Bp

COMMENTAIRE
fin de course s6
bouton poussoir

CHART - PAGE 1

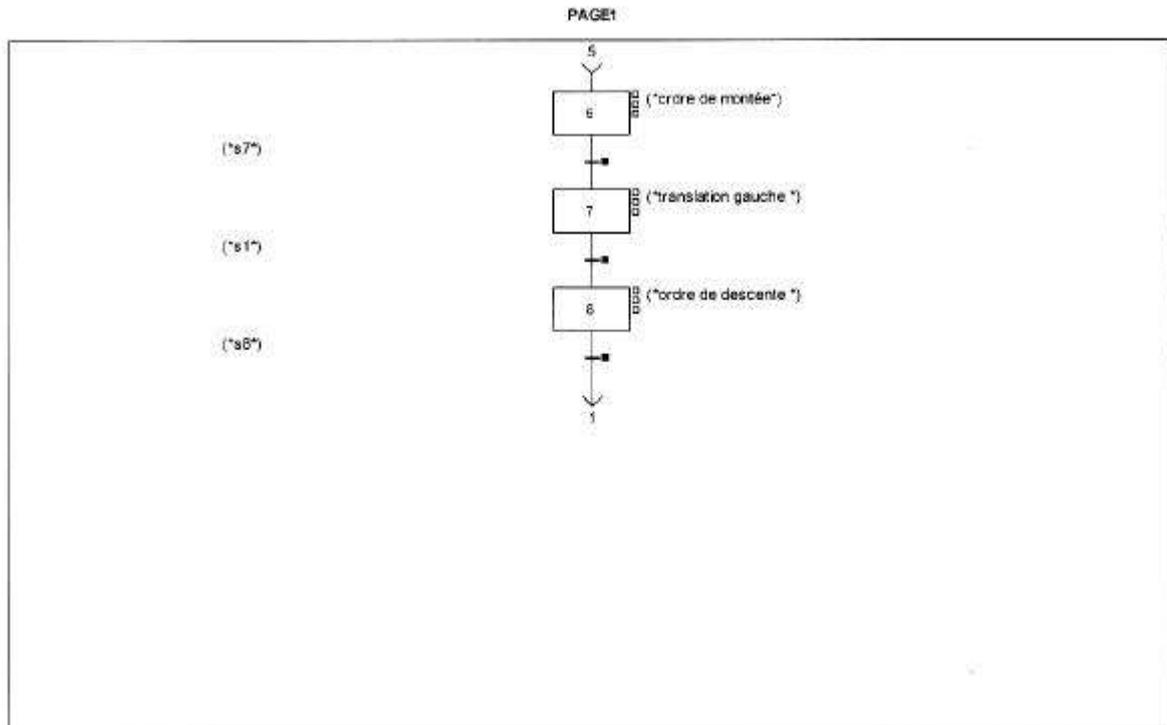


CHART - PAGE1 %X(8)->%X(7)



Liste de Variables utilisées dans le rung :

REPÈRE
%I1.7

SYMBÔLE
Fc7

COMMENTAIRE
fin de course s7