

Table des matières : L'automate

Table des matières : L'automate	1
La programmation automate	2
Présentation de l'automate.....	2
Câblage automate.....	5
A retenir: Cause la destruction de l'automate !!!	11
La sécurité de câblage avec l'automate	13
Grafset à convertir sous PL7 micro	15
Programmation sous PL7_2.....	16
Configuration.....	16
Programmation	16
Transfert,.....	17
La phase test	17
Annexe.....	17
Le Bloc tempo sous PL7-2	18
Programmation sous PL7 Micro	19
L'adressage TSX Micro.....	21
Les Bits systèmes.....	22
Additif aux temporisations	23
Compteur sur Automate sous PL7_2.....	25
Entrée et sorties analogiques sur TSX17	26
Programmation de la tâche maître sur TSX17.....	27
L'écriture Ladder sous Mitsubishi	29
Utilisation de MEDOC sur Mitsubishi	32
La méthode des Mots.....	34
Document constructeur	35
Fonctionnement d'un automate	36
Des capteurs aux actionneurs schéma de principe.....	36

La programmation automate

Notre gamme

- Schneider série TSX 17, 37 (Micro), 47 avec les afficheurs XBT Magelis ou Proface
 - Logiciel : PI7 Micro ou PI7 Pro
- Siemens série S7-200
 - Logiciel : Step7
- Mitsubishi FXON, Série A1
 - Medoc

Info : Schneider et Siemens sont les 2 plus grands fabricants d'automates dans le monde.

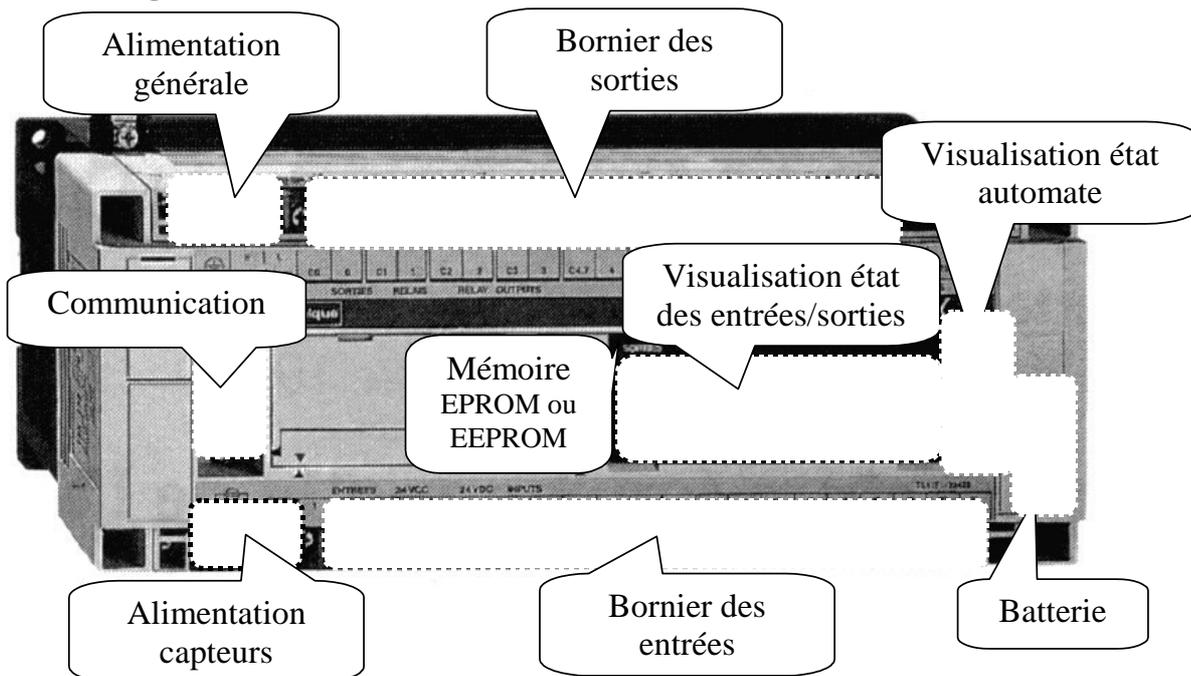
Présentation de l'automate

Définition : Un Automate Programmable Industriel (API) est une machine électronique programmable par un personnel non informaticien et destiné à piloter en ambiance industrielle et en temps réel des procédés ou parties opératives.

Traiter les informations entrantes pour émettre des ordres de sorties en fonction d'un programme.

Un automate programmable est adaptable à un maximum d'applications, 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

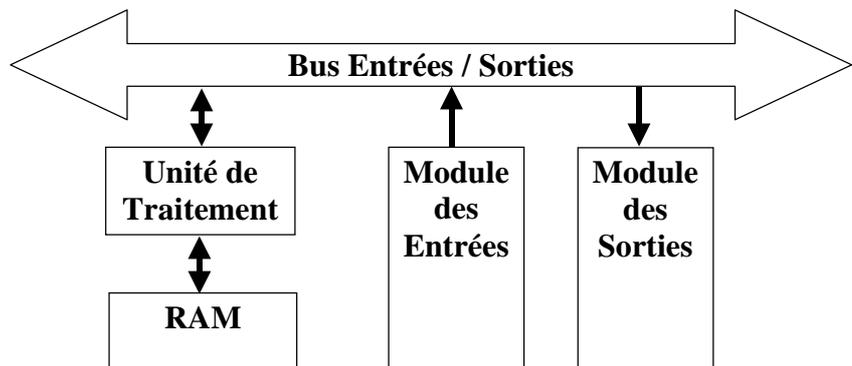
Prise en compte des informations du module d'entrées et écriture de leur valeur dans RAM (zone DONNEE).

•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 :

•Compact ou modulaire	•Sauvegarde (EPROM, EEPROM, pile, ...)
•Tension d'alimentation	•Nombre d'entrées / sorties
•Taille mémoire	•Modules complémentaires (analogique, communication,..)
•Temps de scrutation	•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 figée, 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	1 an 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.

Câblage automate

Les Modules Entrées - Sorties

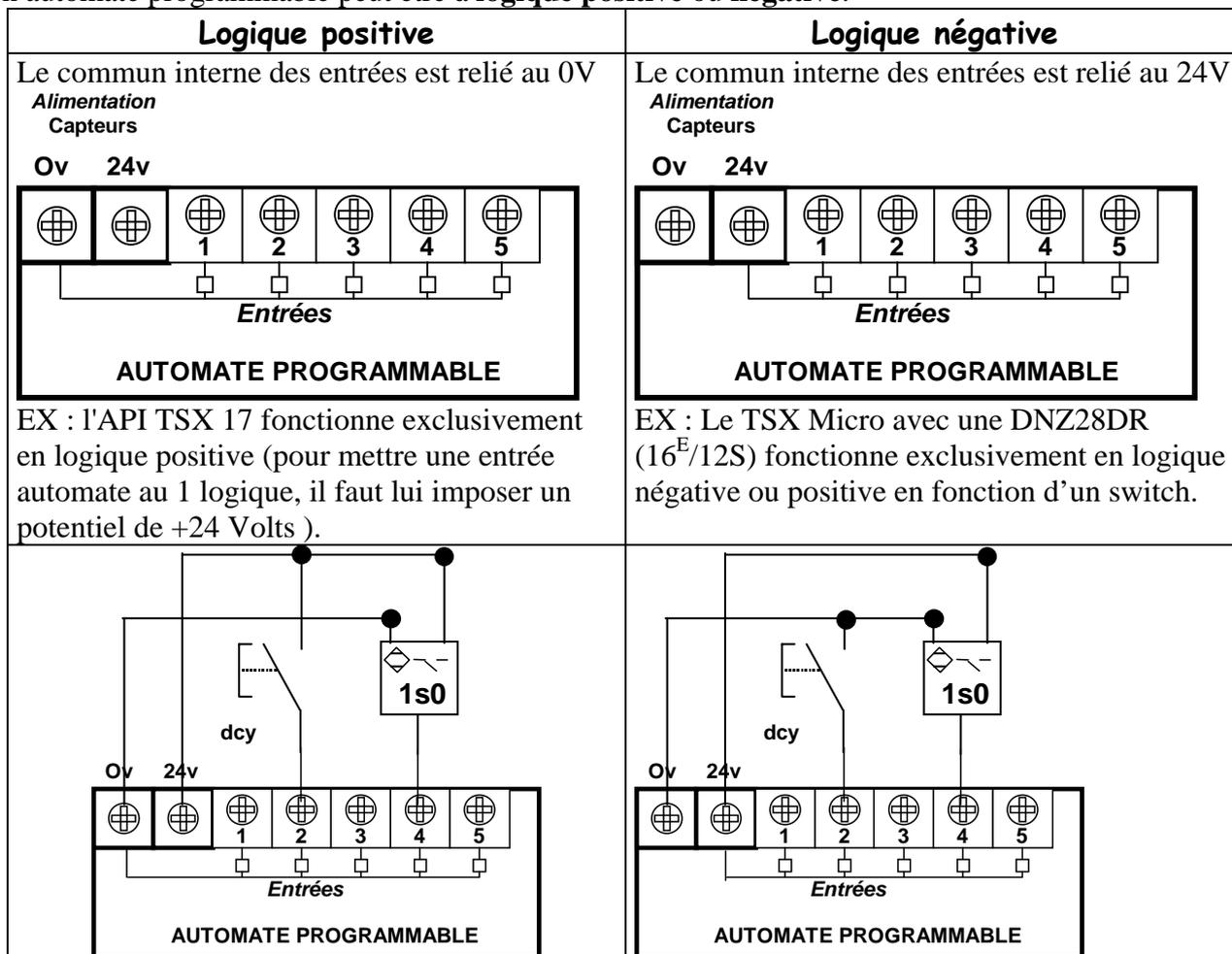
Module d'extension d'Entrées/Sorties TOR Module réseau : communication entre automate	Module d'extension d'Entrées Analogiques 0-10V Module d'extension de Sorties Analogiques 0-10V
--	---

Branchement des Entrées TOR

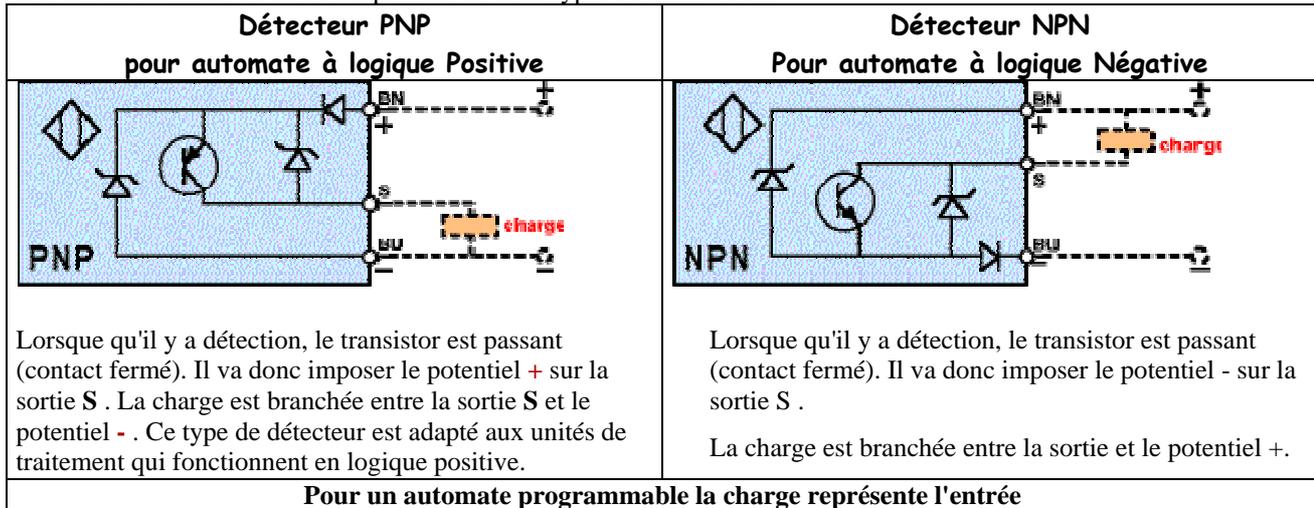
Le principe de raccordement consiste à envoyer un signal électrique vers l'entrée choisie sur l'automate dès que l'information est présente.

L'alimentation électrique peut être fourni par l'automate (en général 24V continu) ou par une source extérieure. (Dans nos TP l'alimentation est tjrs fournis par l'automate)

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



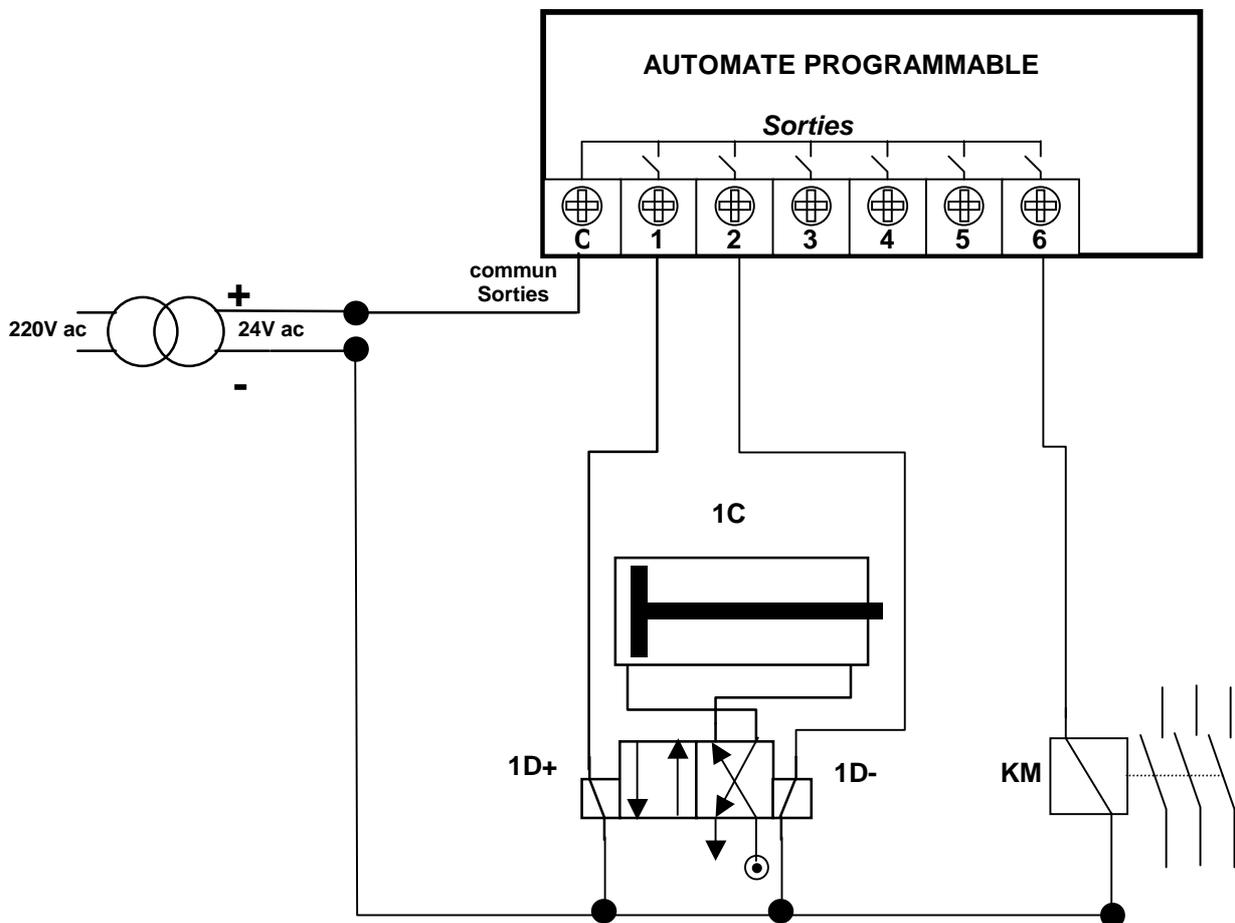
Les détecteurs 3 fils ou électronique sont de deux types **PNP** ou **NPN**.



Branchement des sorties

Le principe de raccordement consiste à envoyer un signal électrique vers le pré actionneur 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.



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> <ul style="list-style-type: none"> ⇒ Valider les entrées correspondant à l'état initial (position) de la Partie Opérative. ⇒ Valider les entrées correspondant aux conditions de marche du cycle. ⇒ Vérifier l'évolution des grafjets (étapes actives). ⇒ Vérifier les ordres émis (Leds de sorties). ⇒ Modifier l'état des entrées en fonction des ordres émis (état transitoire de la P.O.). ⇒ Modifier l'état des entrées en fonction des ordres émis (état final de la P.O.). ⇒ <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> <ul style="list-style-type: none"> ⇒ Positionner la P.O. dans sa position initiale. ⇒ Valider les conditions de marche du cycle. ⇒ Vérifier l'évolution des grafjets et le comportement de la P.O. ⇒ ... <p>Toutes les évolutions du GEMMA et des grafjets doivent être vérifiées.</p>

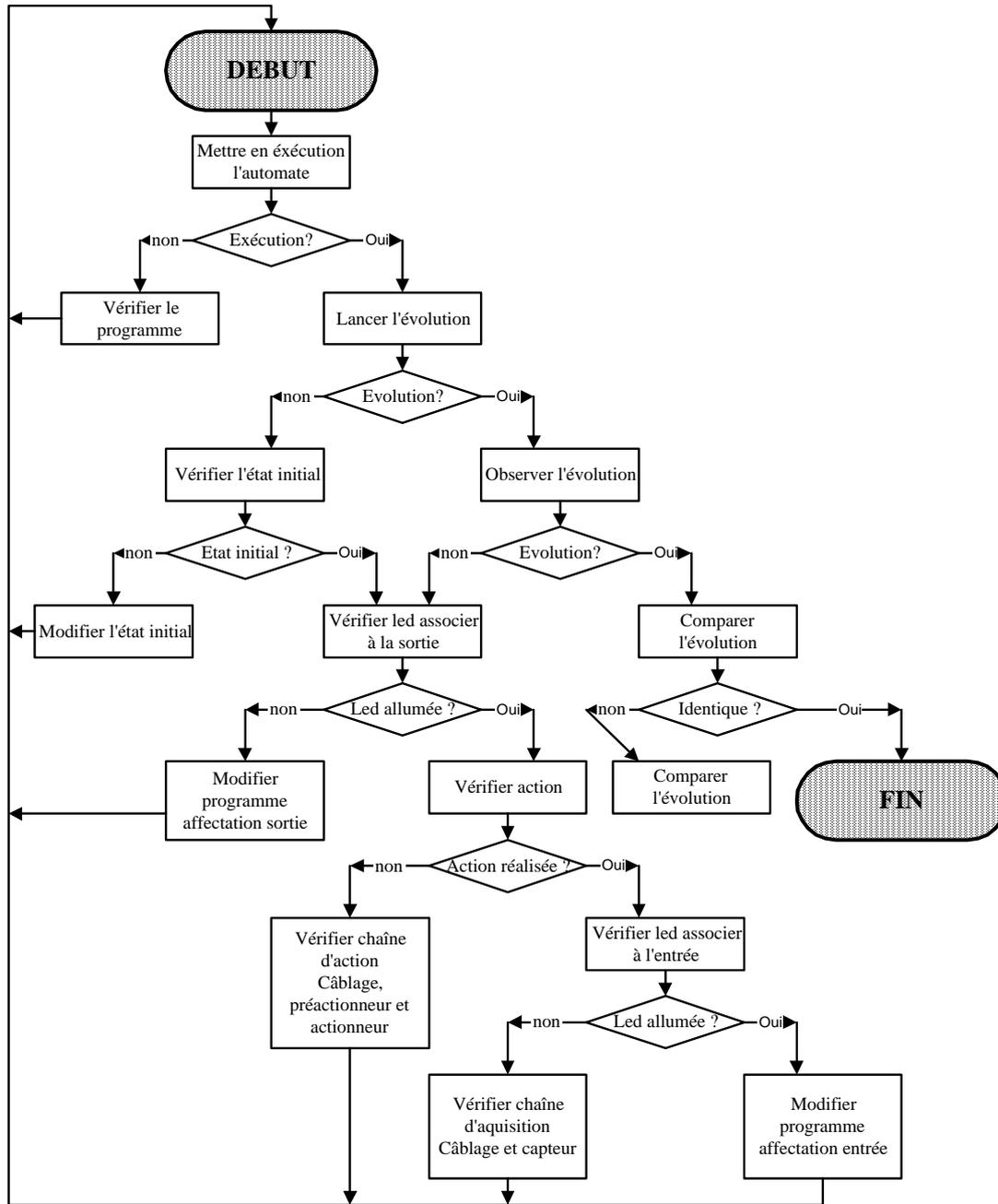
Recherche des dysfonctionnements

Causes de dysfonctionnements

Un dysfonctionnement peut avoir pour origine :

- un composant mécanique défaillant (pré actionneur, 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...).

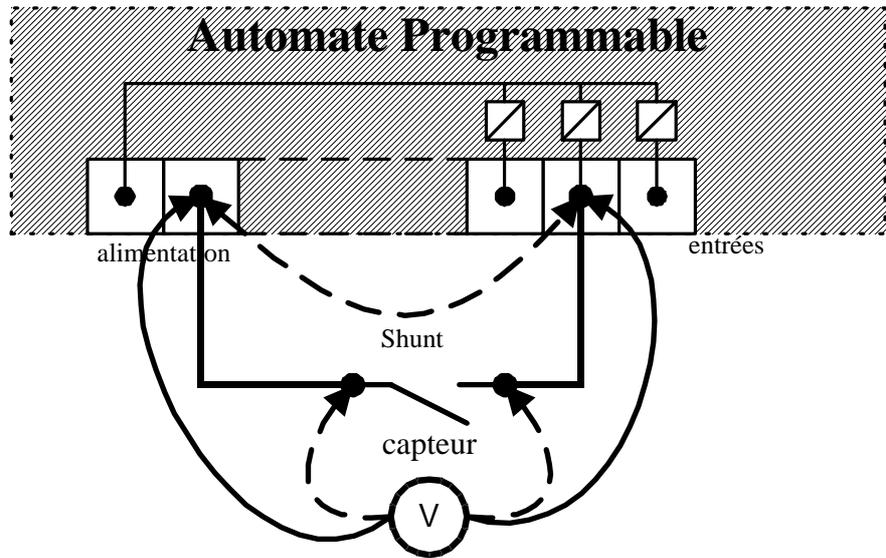
Méthode de recherche des causes de dysfonctionnement



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).

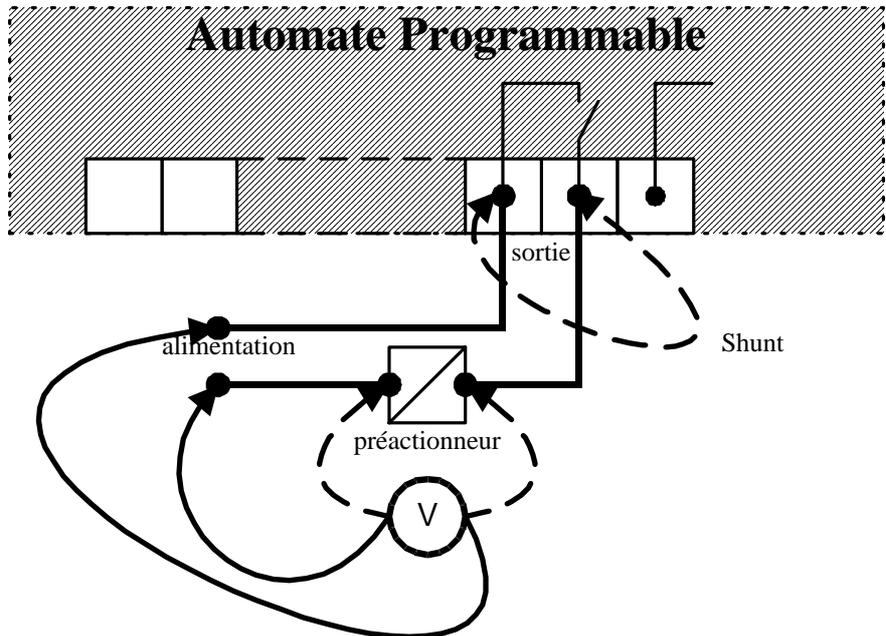
- ⇒ Vérifier l'alimentation des entrées à l'aide d'un voltmètre.
- ⇒ Pour vérifier le capteur et son câblage, tester aux différents points indiqués à l'aide d'un ohmmètre, contact du capteur ouvert, contact du capteur fermé.
- ⇒ Pour vérifier l'interface d'entrée court-circuiter le capteur par un shunt,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



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).

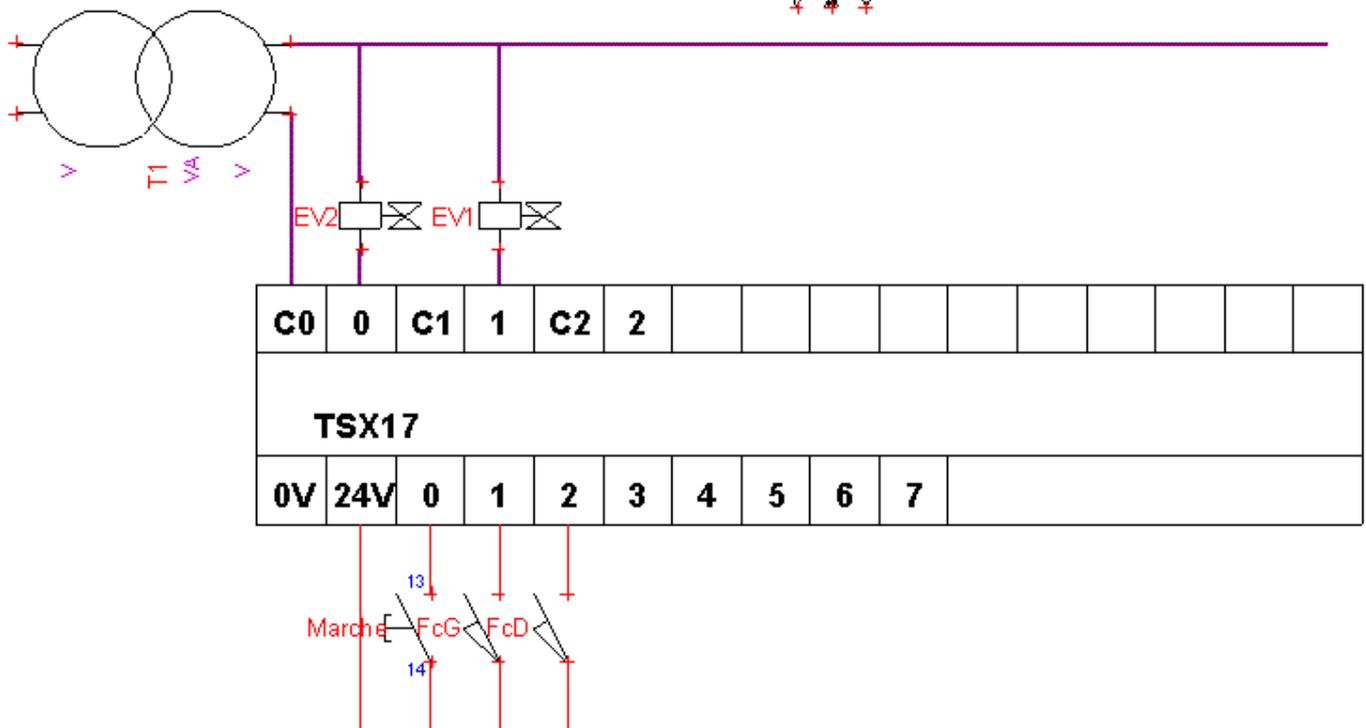
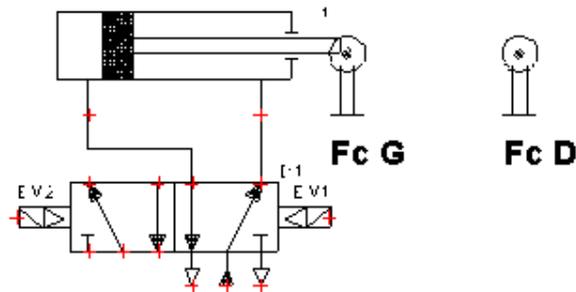
- ⇒ Vérifier que U alimentation existe à l'aide du voltmètre.
- ⇒ Forcer à l'aide du shunt la sortie automate. Si le préactionneur fonctionne, c'est le module de sortie qui est défectueux. Sinon vérifier le préactionneur et son câblage.
- ⇒ Pour vérifier le câblage tester aux différents points de connexion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



A retenir: Cause la destruction de l'automate !!!

- La tension au niveau des Entrées est fournie par l'automate. C'est une tension continue 24V=
- La tension des Sorties vient de l'extérieure. Cela peut être tout type de tension.
- Ne jamais mélanger les 2 sources de tension (Automate 24V= et Extérieure)

Exemple: pour 1 vérin



Note1: Seul les capteurs PNP utilisent le 0Volt de l'automate. Sans capteur PNP le 0 Volt n'est pas branché.

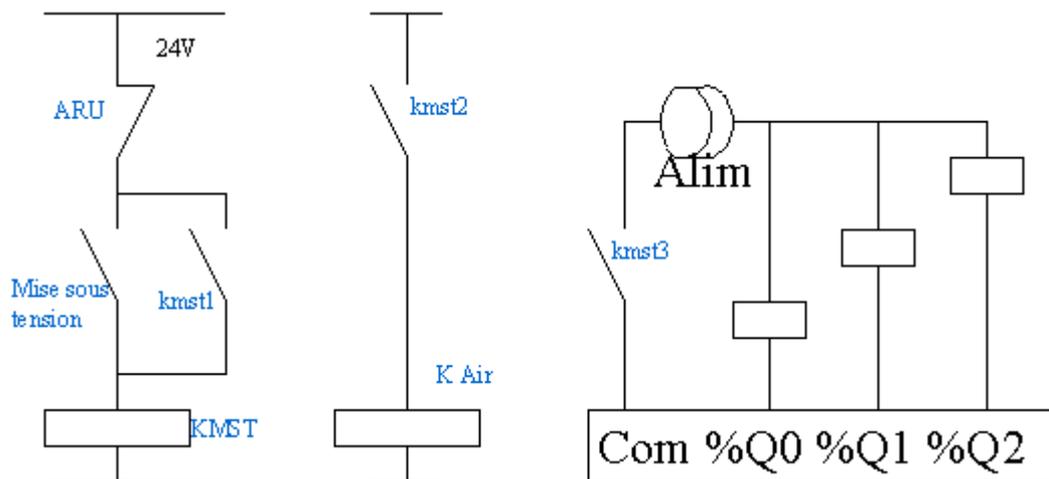
Note2 : Les bobines sont alimentés dans notre cas en 24VAC (Cela dépend du choix des bobines)

Note3 : Un (ou des) fusibles de protection sont nécessaires.

La sécurité de câblage avec l'automate

Le câblage de Kmst est classique, il contient 4 éléments minimum:

- Un BP ARU avec un contact NF.
- Un BP de mise sous tension NO ou de réarmement.
- Un contact d'auto maintien NO minimum.
- Un relais KMST (La tension du relais peut être quelconque.)



Objectif du KMST

Doit couper toute la partie puissance:

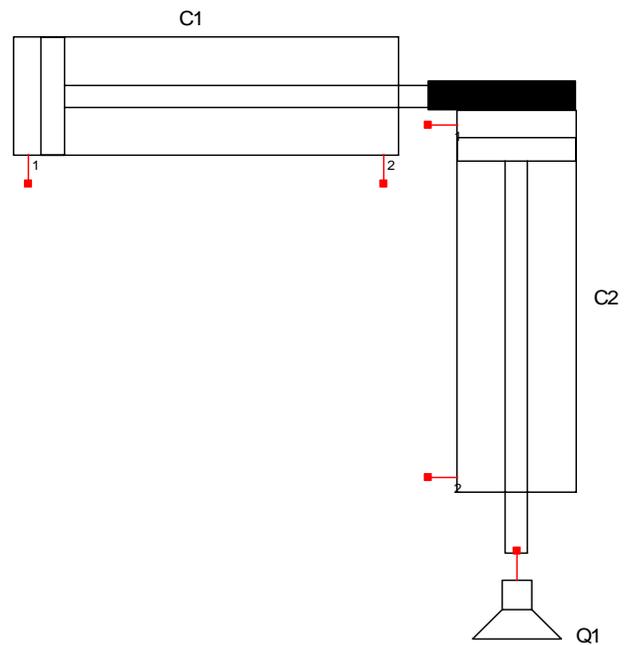
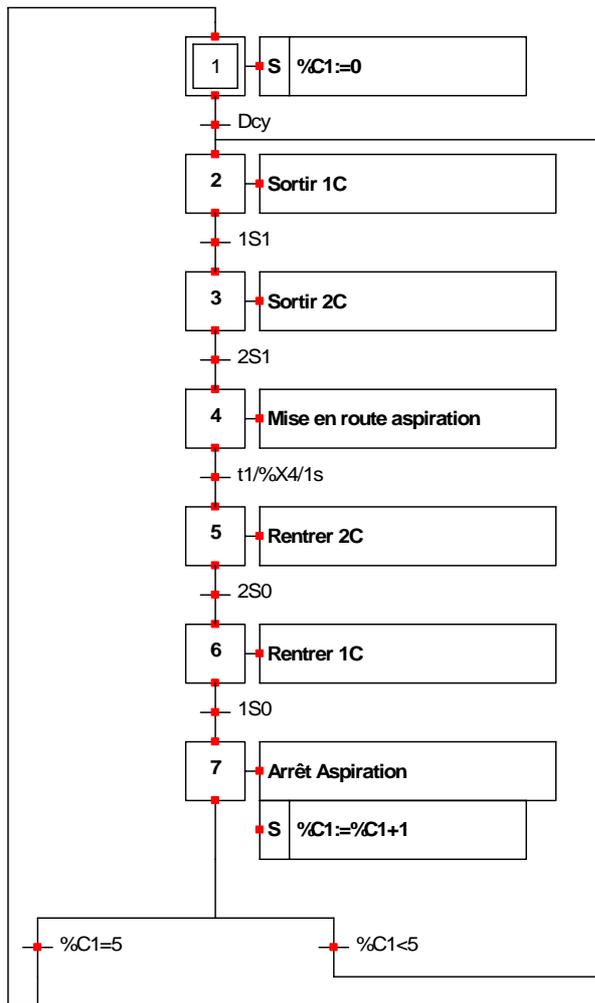
- La pression générale.
- Les moteurs.
- Les distributeurs.
- Les sorties automates.

Ne doit pas couper

- L'automate.
- L'alimentation des capteurs.
- Doit le cas échéant activer des actionneurs de sécurité (freins, alarme)

Note: Si plusieurs sécurités sont nécessaires, comme les capteurs fins de course ou d'autres ARU autour de la machine, ils sont câblés en série avec l'ARU du schéma initiale. Ces capteurs sont en NF.

Grafcet à convertir sous PL7 micro



Réaliser le grafcet partie commande
 Sachant que C1 et C2 sont associé à des distributeurs r
 Que la ventouse est monostable.
 Les 2 vérins ont 2 capteurs fin de course.
 La ventouse ne possède pas de présostat.

Programmation sous PL7_2

Lancement du programme

Sous DOS taper TE (Télémécanique)

Vous devez alors choisir votre automate, si celui ci n'apparaît pas dans le menu, aller le chercher à l'aide de la commande **nouveau** (fichier)

Dans cette partie il est aussi possible de faire les sauvegardes sur disquettes, menu Import / Export

Ce menu permet aussi la sauvegarde sur disquette de votre futur programme, menu **exporter** et **importer**. La touche **Tabulation** -> permet de passer d'une fenetre à l'autre, votre fichier se trouve dans le dossier **Appli**. Valider le nom de votre fichier (une étoile doit apparaitre devant le nom), choisir le lecteur de destination, puis valider en utilisant les touches **Tab** et **entrée**.

Une fois pl7-2 lancer, démarrer par le menu Configuration de l'automate. Attention à la mémoire et au langage utilisé. (Grafcet)

A chaque page la ligne du bas représente les fonctions sélectionnable par les touches F1 à F9.

Pour revenir d'un cran en arrière (ou d'un sous menu) utiliser soit la touche **entrée** ou **F9** pour la fonction valide. La touche **Fin** permet aussi de sortir mais sans prendre en compte les dernières modifications. Il faut procéder par étapes pour créer votre programme.

Configuration

- Mémoire: Dans la configuration vous devez définir la mémoire disponible sur l'automate.
- Langage: Définir si vous voulez utiliser du grafcet ou non.
- Les cartes d'extension: En tsx47 par exemple l'automate possède des cartes (communication RS232C, entrées, sorties,...), il faut donc informer l'ordinateur de la présence de ces cartes et de leurs positions en donnant la référence de chaque carte. (la référence peut être obtenue à l'aide du catalog F9)

Tjrs valider vos modifications par la touche entrée ou F9.

Programmation

en grafcet, 3 modules permettent de rentrer votre programme.

PRL

Contient les informations de traitement préliminaire, le traitement de l'arrêt d'urgence, des calculs de nombre de pièce, les organigrammes,...

Chart:

Contient votre grafcet avec les transitions.

L'écriture se fait en mode **Modif** avec les touches F1 à F9

Pour effacer un dessin la touche **Supprim**, sauf pour les divergences ou il faut repasser en mode dessin de divergence.

Pour écrire les transitions 2 fois de suite 'la touche **page down** (fleche barré vers le bas) ou **zoom** sur la console 407', pour revenir au grafcet valider par **entrée** et puis la touche 'page up' (Les transitions n'apparaissent pas sur le dessin du grafcet)

Pour les divergences utiliser les touches de fonctions.

Une fois votre grafcet fini, en mode modif pour sortir appuyer sur '**entrée**', l'écran doit afficher en haut à gauche **OK au lieu de NOK**.

Si vous le désirez vous pouvez entrer des actions en zoomant sur les actions, mais préférer le traitement post.

POST

Ici vous écrivez vos actions. Pour créer une nouvelle page d'action, la touche **inser**. Ensuite attribuer un numéro de **label** à la page.

Attention une sortie automate sauf cas exceptionnelle ne peut et ne doit apparaitre qu'une seule fois.

Pour supprimer une page utiliser la touche suppr.

en Ladder

Il faut tout écrire en Mast.

Transfert,

dés que votre programme est fini il faut le sauvegarder sur le disque dur, donner un nom 'archiv' et sauvegarde terminal->disque dur.

Ensuite le transfert vers l'automate peut être fait, pour cela passer en mémoire TSX dans le menu général (F1) et transférer 'terminal->tsx'

La phase test

peut démarrer. **INITialiser** l'automate, et passer en **Run**. Visualiser en temps réel votre grafct en passant en mise au point. Les cases actives du grafct passent en noir. Tester en premier temps votre grafct sans faire de câblage, il suffit de relier les entrées Ix,x au 24V de l'automate, les étapes doivent évoluer. Ensuite câbler et tester.

Vous pouvez aussi vous aider du mode réglage pour forcer des variables à 1 ou à 0 (pour faire bouger des actionneurs manuellement par exemple)

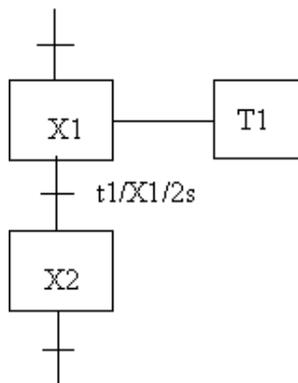
Annexe

- Le logiciel ne garde pas en mémoire vos modifications: Vous ne validez pas correctement vos pages par la touches F9 ou Entrée
- Pas d'accès au menu chart, post, prl: Vous avez mal configurer votre automate.
- Ix,y ou Ox,y sont remplacés par un point: Vos cartes d'entrées sorties sont mal configurés, ou vous faites erreur sur la valeur de x ou de y.
- Votre grafct ne démarre pas, les étapes initiales ne se norcissent pas: Votre grafct n'est pas OK.

Le Bloc tempo sous PL7-2

Auteur: Roizot Sébastien

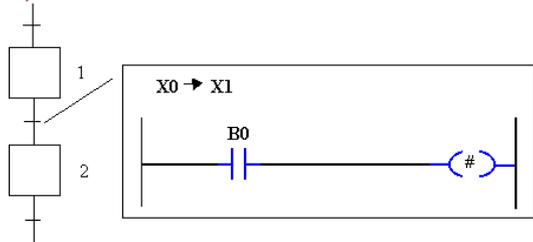
Principe :



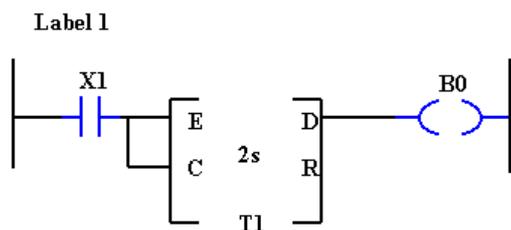
L'action ne s'écrit pas en théorie dans le grafcet.

Programmation:

1)- Le CHART



2)- le POST



3)- Réglage de la tempo

1 CONFIGURATION **ENTER**
 3 MOTS ET BLOCS **ENTER**
 0 TEMPORISATEUR **F1** « content »
 Sélectionner T1 dans la liste des temporisateurs
 T1 : Preset **F1** taper 2 **ENTER**
Timebase **F2** choisir 1s **ENTER**

Fonctionnement:

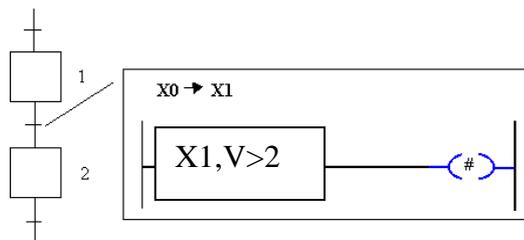
Lorsque le grafcet se trouve en X1 la tempo est lancée.

La temporisation terminée, la sortie D passe à 1 et donc le bit B0 aussi.

La transition X1-X2 est vraie, le grafcet peut donc évoluer et passer à l'étape X2.

Une autre solution toute simple:

Uniquement pour des temporisations de base de temps la seconde, un bloc Compar en transition du grafcet et c'est tout.



PI7micro : %X1.T > 20

Exercice d'application :

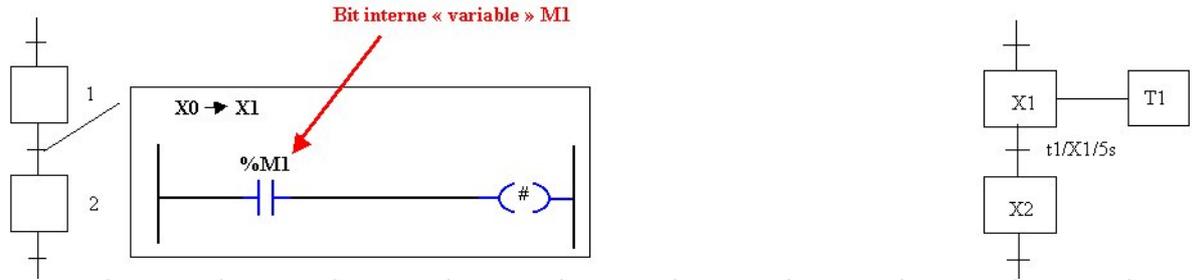
On désire réaliser le grafcet d'un clignotant. L'appuie sur dcy je lance le clignotement d'un voyant tant que l'on n'appuie pas sur le BP arrêt.

Le clignotement du voyant est caractérisé par un allumage de 2s et une extinction de 1s.

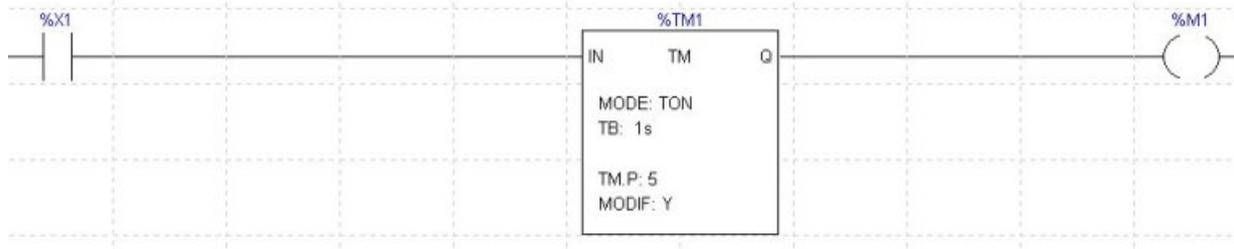
Programmation sous PL7 Micro

Programmation

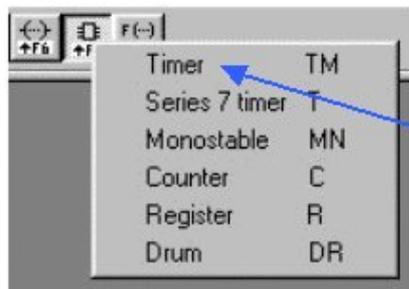
a)- Chart:



b)- Post:



c)- Insertion d'un bloc tempo " Timer "



Cliquer sur l'icône
Choisir Timer



Cocher la case paramètres

d)- Réglage des paramètres du bloc tempo

Repère	Type	Symbole
%TM0	TM	
%TM1	TM	
%TM2	TM	
%TM3	TM	
%TM4	TM	
%TM5	TM	
%TM6	TM	
%TM7	TM	
%TM8	TM	
%TM9	TM	
%TM10	TM	
%TM11	TM	
%TM12	TM	
%TM13	TM	
%TM14	TM	
%TM15	TM	
%TM16	TM	

Double cliquez sur FB prédéfinis pour accéder aux paramètres de la tempo

Repère	Type	Symbole	Preset	Mode	TB	Req
%TM0	TM		9999	TON	1 mn	✓
%TM1	TM		9999	TON	1 mn	✓
%TM2	TM		9999	TON	1 mn	✓
%TM3	TM		9999	TON	1 mn	✓
%TM4	TM		9999	TON	1 mn	✓
%TM5	TM		9999	TON	1 mn	✓
%TM6	TM		9999	TON	1 mn	✓
%TM7	TM		9999	TON	1 mn	✓
%TM8	TM		9999	TON	1 mn	✓
%TM9	TM		9999	TON	1 mn	✓
%TM10	TM		9999	TON	1 mn	✓
%TM11	TM		9999	TON	1 mn	✓
%TM12	TM		9999	TON	1 mn	✓
%TM13	TM		9999	TON	1 mn	✓
%TM14	TM		9999	TON	1 mn	✓
%TM15	TM		9999	TON	1 mn	✓
%TM16	TM		9999	TON	1 mn	✓
%TM17	TM		9999	TON	1 mn	✓
%TM18	TM		9999	TON	1 mn	✓
%TM19	TM		9999	TON	1 mn	✓
%TM20	TM		9999	TON	1 mn	✓

Repère	Type	Symbole	Preset	Mode	TB	Req
%TM0	TM		9999	TON	1 mn	✓
%TM1	TM		5	TON	1 s	✓
%TM2	TM		9999	TON	1 mn	✓

Indiquer la valeur de votre tempo : $5 * TB = 1s$

Choisir l'unité de la tempo dans notre cas 1s

3)- Fonctionnement:

Lorsque le grafcet se trouve en X1 la temporisation est lancée.

La temporisation terminée, la sortie Q du bloc tempo passe à 1 et donc le bit interne M1 aussi.

La transition X1-X2 est vraie, le grafcet peut donc évoluer et passer à l'étape X2.

L'adressage TSX Micro

L'adressage normalisé est le suivant :

Pour les Entrées et les Sorties

%I ou %Q	X ou W ou D	x	i
symbole type d'objet	Format	Position dans le bac	N° Voie
%I: Entrée	X: booléen (peut être omis)	0: Correspond au processeur	
%Q: Sortie	W: Mot	1 à 127: Position des cartes.	
	D: Double Mot		

Exemple:

- %I1.5 voie d'entrée N°5 du module en position 1 (La 1er carte après le processeur)
- %IW8.0 : Voie d'entrée analogique du module en position 8

Pour l'adressage des Mots

%M ou %K ou %S	F ou W ou D ou F	i
Type d'objet	Format	Numéro
<ul style="list-style-type: none"> • M: Interne • K: Constant • S: Système 	<ul style="list-style-type: none"> • B: Octect • W: Mot • D: Double mot • F: Flottant 	

Attention le mot %MD0 et [%MW1,%MW0] sont les mêmes en mémoire.
Idem pour %MD2 et [%MW3,%MW2], etc.

Note par rapport aux autre automates

- Pour le TSX17 sans extension l'automate est considéré comme le module 0, les entrées et sorties s'écrivent : IO,i et OO,i. Si l'on ajoute une extension entrée analogique par exemple : IW1,i
- Pour les Mitsubishi les entrées sorties sont à lire sur l'automate : X400,... et Y430,...
- Pour le TSX 47 (comme pour le TSX Micro & Premium, Mitsubishi A1), l'automate est modulable. Si le rack 1 comporte des 16 entrées, celle-ci s'écrivent : I1,0 à I1,15.
- Au niveau de la programmation, on doit aller dans le menu configuration, pour informer le logiciel des cartes installées dans chaque position.

Les Bits systèmes

Ces bits peuvent être testés dans le programme utilisateur afin de détecter tout événement de fonctionnement devant entraîner une procédure particulière de traitement. Certains d'entre eux doivent être remis dans leur état initial ou normal par programme. Cependant, les bits système qui sont remis dans leur état initial ou normal par le système ne doivent pas l'être par programme ou par le terminal.

Bits	Fonction	Désignation système
%S0	Démarrage à froid.	Normalement à l'état 0, est mis à l'état 1 par : reprise secteur avec perte des données (défaut batterie), programme utilisateur, terminal, changement de cartouche, appui sur le bouton de RESET. Ce bit est mis à 1 durant le premier cycle complet. Il est remis à 0 avant le cycle suivant. Fonctionnement: voir intercalaire A, chapitre 1.4.
%S1	Reprise à chaud.	Normalement à l'état 0, est mis à l'état 1 par : reprise secteur avec sauvegarde des données, . programme utilisateur, . terminal. Il est remis à 0 par le système à la fin du premier cycle complet et avant la mise à jour des sorties. Fonctionnement: voir intercalaire A, chapitre 1.4.
%S4	10ms	Base de temps Bits dont le changement d'état est cadencé par une horloge interne.
%S5	100ms	Base de temps
%S6	1s	Base de temps
%S7	1min	Base de temps
%S8	Test du câblage	Normalement à l'état 1, ce bit est utilisé pour le test du câblage, lorsque l'automate TSX 37 est dans l'état "non configuré" . état 1 : les sorties sont forcées à 0, . état 0: les sorties peuvent être modifiées par un terminal de réglage.
%S9	Mise en position de repli des sorties sur tous les bus	Normalement à l'état 0. Peut être mis à l'état 1 par programme ou par le terminal: . état 1 : provoque le forçage en position de repli des sorties de l'automate (bus X, FIPIO, AS-i...), . état 0: les sorties sont mises à jour normalement.
%S10	Défaut E/S	Normalement à l'état 1. Est mis à l'état 0 quand un défaut d'E/S d'un module en rack ou d'un module déporté (FIPIO) (configuration non conforme, défaut d'échange, défaut matériel) est détecté. Le bit %S 10 est remis à 1 dès la disparition du défaut.

%S11 1 =débordement chien de garde,

%S13 1 =premier cycle après mise en RUN

%S21 1 = initialisation du Grafcet, %S22 1 =remise à zéro du Grafcet, %S23 1 =pré positionnement et gel du Grafcet

%S30 1 = activation de la tâche maître

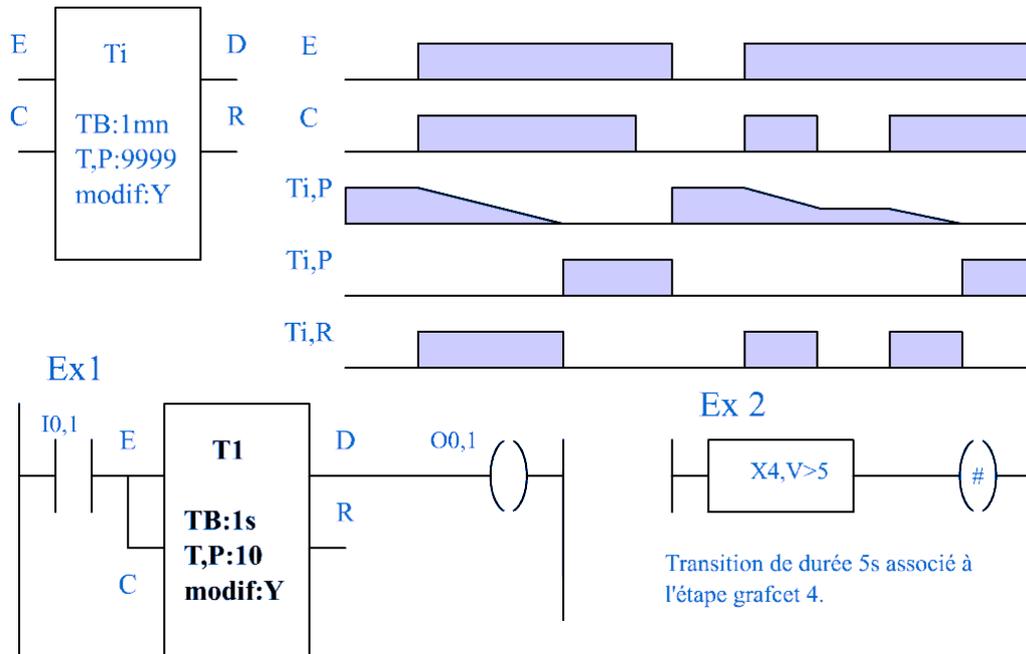
%S31 1 = activation de la tâche rapide

Note : Utiliser l'aide du logiciel pour plus d'information.

Additif aux temporisations

Les paramètres complexes

- TB: Base de temps 1mn, 1s, 100s, 10ms
- Ti,V: Valeur courante qui décroît de Ti,P vers 0
- Ti,P: Valeur de préselection qui donne la durée de la tempo multiplié par TB
- Modif: Sur oui, permet la modification de Ti,P en mode réglage.
- E: Enable: Doit être à 1 pour l'écoulement de la tempo
- C:Contrôle: Sur 0 gèle la valeur courante, à 1 pour l'écoulement.
- D: Done: Tempo écoulé (bit équivalent Ti,D)
- R: Running: A 1 si la tempo est entrain de s'écouler.



1. On active la sortie O0,1 lorsque I0,1 est présent depuis 10 secondes.
2. Chaque étape possède une temporisation de base la seconde, qui permet d'écrire des transitions.

Ecriture simplifié

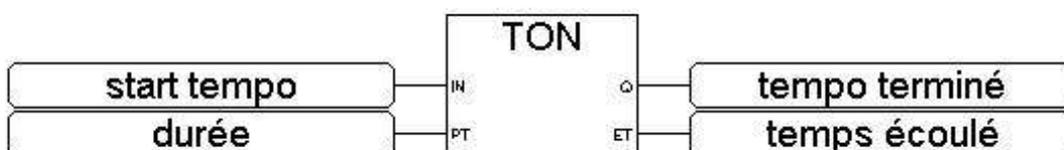
Le schéma de l'exemple 2 est l'écriture simplifiée. **Il n'y a que ça à écrire.**

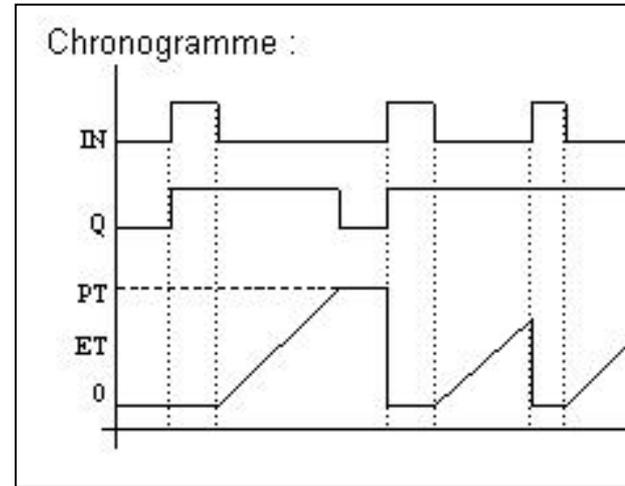
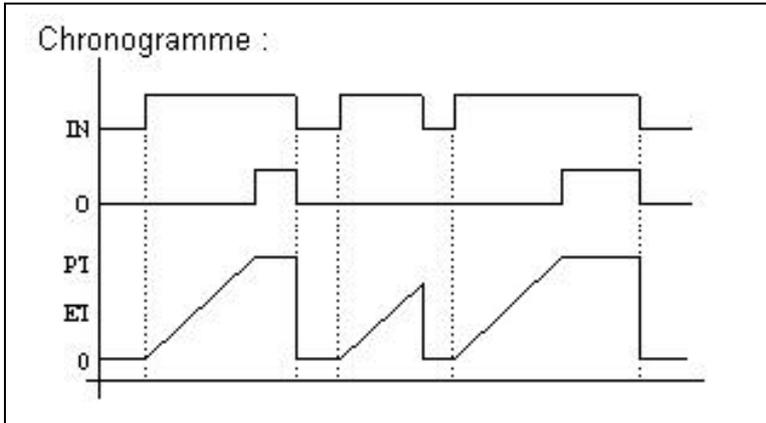
Sur TSX Micro l'écriture change un petit peu pour le bloc comparaison: %X4.T > 50 (valeur en millisecondes)

Exemple à traiter:

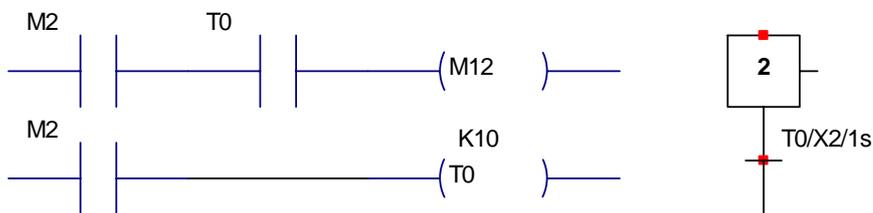
1. Créer une transition de grafcet de 2,5 secondes.
2. Créer un ordre conditionnel.

Les temporisations du type TON et TOF sur TSX micro





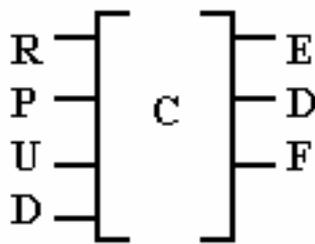
Ecriture sur Mitsubishi. :



L'étape 2 lance une temporisation de 1 seconde (base de temps milliseconde), d'où la condition d'évolution M12.

Compteur sur Automate sous PL7_2

La valeur du retard, la base de temps, le type de tempo se programme dans le menu configuration.



Paramètres: Du compteur Ci (C0 à C31)

Ci,V: Valeur courante: Incrémenté ou décrémente en fonction des entrées U et D.

Ci,P: Valeur de présélection: Valeur à atteindre

Modif: A 1 autorise la modification en mode réglage.

R: Reset: Remise à 0.

P: Preset: Mise à la valeur de départ soit Ci,V:=Ci,P

U: Up: Incrémente Ci,V sur front montant.

D: Down: Décrémente Ci,V sur front montant.

E: Empty: Passe à 1 lorsque le compteur passe de 0 à 9999 après un D.

D: Done: Passe à 1 lorsque la valeur est atteinte.

F: Full: Passe à 1 lorsque Ci,V passe de 9999 à 0 après un U.

Résumé :

Il existe sur les TSX 17 32 compteurs (C0 à C31). R (reset) met le compteur et les sorties à 0. P (preset) met le compteur à la valeur finale et la sortie D (done) à 1 (sauf si R=1). U (up) incrémente le compteur, D (down) le décrémente. La sortie F (full) vaut 1 lors du passage du compteur (par U) de 9999 à 0, E (empty) lors du passage (par D) de 0 à 9999. Si U=D=1, le compteur est inchangé.

La valeur de présélection (Ci,P, entre 0 et 9999) se définit en "zoomant" sur le compteur.

Exemple à traiter:

Graficet avec un comptage et une divergence en OU.

Entrée et sorties analogiques sur TSX17

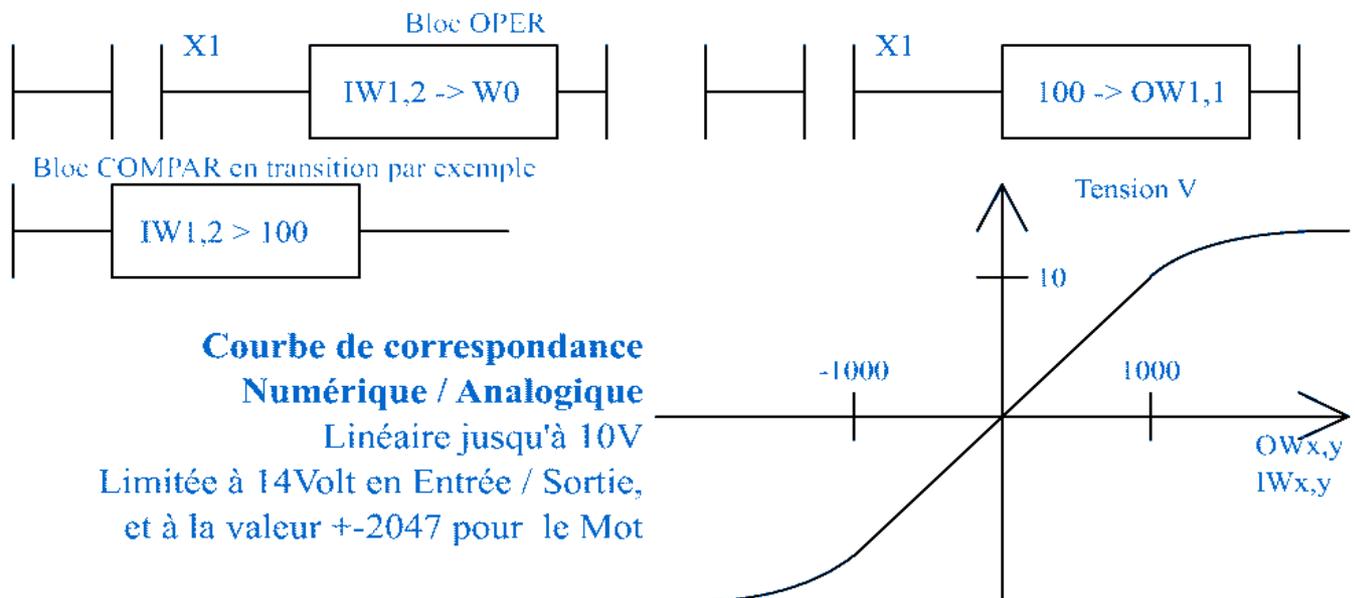
Module d'entrées:

- 4 entrées analogiques (11Bits + signe)
- $IW_{x,y}$ (x la position du module, y le numéro de l'entrée [0,3])
- TSX AEG4110 : Référence de configuration pour pl7 2 (Menu Configuration Entrée / Sortie)
- Tension [-10V, +10V] et Courant 4/20 mA

Module de sorties

- 2 sorties analogiques (11Bits + signe)
- $OW_{x,y}$ (x la position du module, y le numéro de l'entrée [0,1])
- TSX ASG2000 : Référence de configuration pour pl7 2 (Menu Configuration Entrée / Sortie)
- Tension [-10V, +10V] et Courant 4/20 mA

Exemple de programme et courbe de correspondance:



Exercice: Créer un programme de sécurité qui empêche de transmettre une valeur analogique supérieur à la valeur 1100 (Soit 11 Volt).

Note: Différence par rapport au TSX micro

- Affectation : $\%QW_{x,y}$ et $\%IW_{x,y}$
- Limite du mot de la valeur analogique: [-10000, +10000] qui correspond à [-10V, +10V]

Programmation de la tâche maître sur TSX17

La tâche maître est par défaut active, la tâche rapide est par défaut active si elle est programmée. La tâche événementielle est activée lors d'apparition de l'événement qui lui a été associé.

Lors de l'arrivée d'un événement ou début de cycle de la tâche rapide, celle-ci arrête l'exécution en cours des tâches moins prioritaires, afin d'exécuter son traitement; la tâche interrompue reprenant la main lorsque les traitements de la tâche prioritaire se terminent.

L'exécution des tâches rapides et événementielles peut être contrôlée par programme à travers l'utilisation des bits système :

- %S30 permet d'activer ou pas la tâche maître MAST
- %S31 permet d'activer ou pas la tâche rapide FAST, (Un Reset de SY19 sur TSX17 en PRL active la tâche rapide, et le Set la désactive)
- %S38 permet d'activer ou pas les tâches événementielles EVTi,

Exemple de traitement multitâche

- . tâche maître cyclique
- . tâche rapide de période 20 ms . tâche événementielle

Tâche maître

Cette tâche qui est la moins prioritaire gère la majeure partie du programme application. La tâche MAST est organisée selon le modèle décrit au sous-chapitre précédent : lecture implicite des entrées, exécution du programme application et écriture implicite des sorties.

Quel que soit le mode de fonctionnement: périodique ou cyclique, la tâche est contrôlée par un chien de garde qui permet de détecter une durée anormale du programme application. En cas de débordement, le bit système %S11 est positionné à 1 et l'application est déclarée en défaut bloquant pour l'automate.

Le bit système %S30 permet de valider ou d'inhiber la tâche maître.

Cette tâche plus prioritaire que la tâche maître MAST est périodique afin de laisser le temps à la tâche moins prioritaire de s'exécuter.

De plus, les traitements qui lui sont associés doivent donc être courts pour ne pas pénaliser la tâche maître. Comme pour la tâche maître, le programme associé se compose de sections et de sous-programmes.

La période de la tâche rapide FAST est fixée en configuration, de 1 à 255 ms. Celle ci peut être définie supérieure à celle de la tâche maître MAST pour s'adapter à des traitements périodiques lents mais prioritaires. Le programme exécuté doit cependant rester court pour éviter le débordement des tâches moins prioritaires.

La tâche rapide est contrôlée par un chien de garde qui permet de détecter une durée anormale du programme application. En cas de débordement, le bit système %S11 est positionné à 1 et l'application est déclarée en défaut bloquant pour l'automate.

Contrôle de la tâche rapide

Le mot système %SW1 contient la valeur de la période, il est initialisé sur reprise à froid par la valeur définie en configuration, il peut être modifié par l'utilisateur par programme ou terminal.

Des bits et mots système, permettent de contrôler l'exécution de cette tâche :

. %S19: signale un débordement de période, il est positionné à 1 par le système, lorsque le temps de cycle devient supérieur à la période de la tâche.

. %S31 : permet de valider ou d'inhiber la tâche rapide, il est mis à 0 par le système sur démarrage à froid de l'application, à la fin du premier cycle de la tâche maître. Il est mis à 1 ou à 0 pour valider ou inhiber la tâche rapide.

Visualisation des temps d'exécution de la tâche rapide

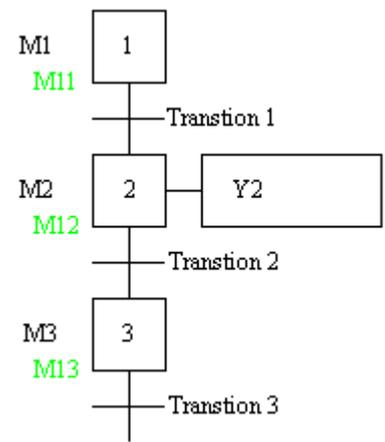
Les mots système suivants permettent d'avoir des informations sur le temps de cycle :

%SW33 contient le temps d'exécution du dernier cycle,
%SW34 contient le temps d'exécution du cycle le plus long,
%SW35 contient le temps d'exécution du cycle le plus court.

L'écriture Ladder sous Mitsubishi

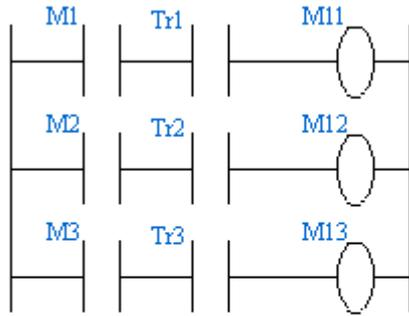
4 éléments de programmation :

- L'initialisation
- Les conditions d'évolutions.
- La structure Grafcet
- Le Traitement Post

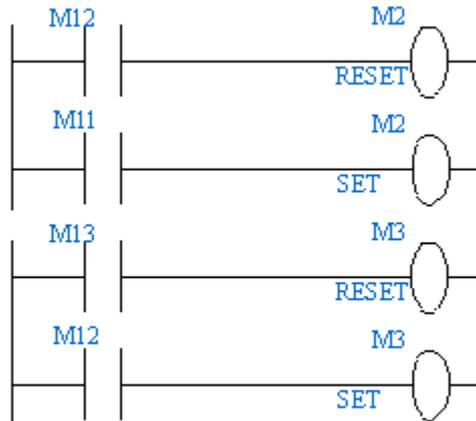


D'après l'extrait du grafcet ci-contre, on en déduit les règles d'écritures:

1) Les Conditions d'évolutions



2) Les règles d'évolutions



3) Le traitement Postérieur



Note:

- Pour l'initialisation, il faut que sur chaque variable d'étape on est une condition de Reset en plus (Sauf pour l'étape initiale ou la condition est sur le Set).
- Sur le FX0N le Bit **M8002** passe à 1 pendant un cycle automate lorsque vous passer manuellement l'automate en Run (switch sur l'automate)
- Pour respecter la règle 5 (Activation et désactivation simultanées Si, au cours du fonctionnement de l'automatisme, une même étape doit être désactivée et activée simultanément, elle reste active.) il faut toujours écrire le Reset de l'étape avant le Set.

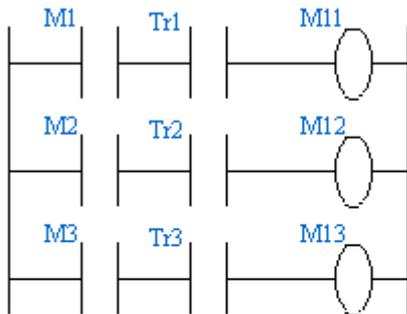
La même chose en monostable :

L'écriture Ladder se présente sous 4 phases:

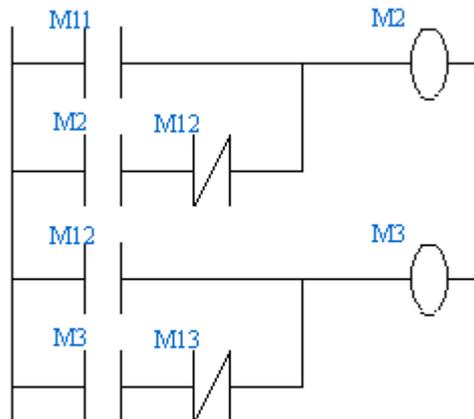
- L'initialisation
- Les conditions d'évolutions.
- La structure Grafcet
- Le Traitement Post

D'après l'extrait du grafcet ci-contre, on en déduit les règles d'écritures:

1) Les Conditions d'évolutions



2) Les règles d'évolutions



3) Le traitement Postérieur



Les références mitsubishi

Les Entrées X0 - X177 (Octal), les sorties Y, les variables internes M0 -M499, les étapes grafcet S, les temporisations T, les compteurs C, les registres de Données 16 bits D, les constantes décimales K, les constantes Hexadécimal H.

Les temporisations:

T0 à T199: 100ms

T200 à T245: 10ms

Les compteurs

C0 à C99

Registre de données D

D0 à D199: Registre 16bits

Utilisable en 32 bits: D1 poids forts, D0 poids faible

Registre d'index V,Z

Regroupement de bit:

K0M0 regroupement de 4 bits (M0-M3)

K1M0 regroupement de 8 bits (M0 -M7)

K2M0 regroupement de 12 bits (M0-M11)

Instruction de saut: CJ (Saut P8 CJ P8)

Les instructions de Bases SET, RST

Front montant PLS, Front descendant PLF

Transfert de zone mémoire MOV

Instruction de comparaison: CMP

Addition: ADD

Rotation: ROR, ROL et avec retenu RCR, RCL (Bit carry:M8022)

Décalage: SFTR, SFTL

Remise à zéro de zone mémoire: ZRST

Utilisation de MEDOC sur Mitsubishi

Début

Lancement du logiciel en tapant medoc. (en cas d'échec taper cd medoc puis medoc)

Touches de base:

Les flèches permettent de choisir la fonction souhaitée.

Enter Permet de valider le menu.

Esc Permet de revenir dans le menu supérieur.

F1 Aide en ligne, à tout moment F1 permet une aide avec les touches à utiliser, indispensable.

Touche Accès rapide: 2 Rentre dans l'éditeur, 5 dans le menu transfert.

Menu principal

Start Edit Transfer Print Files Options Quit

Menu Start

Menu Open New_Proj ListProj PLC_Chang Save Quick Make_Lib

Open: Ouverture d'un ancien programme.

New_Proj: Nouveau projet Choix de l'automate utilisé avec les flèches et la touche tabulation.

ListProj: Liste les projets existants

PLC_Chang: Change l'automate utilisé.

Save: Sauvegarde votre projet en cours sur le disque.

Quick, Make_Lib: Voir documentation.

Menu Edit

Menu Header Name Instr Ladder Param Text Other

Ladder: Permet l'édition du programme.

Voir documentation pour les autres.

Sous Menu Ladder

Find Save Name Copy Move Delete Exchange Test

F2: Permet de rentrer dans l'éditeur et d'en sortir, ensuite vous pouvez vous déplacer de ligne en ligne avec les flèches.

F7: Permet d'éditer les nouvelles instructions et de sortir du mode édition.

1 à 9: (avec ou sans shift) Donne les symboles et [] pour les instructions complexes.

Espace: Correction, Efface l'élément.

F5 et F6 permettent d'insérer des lignes de commentaire.

CTRL H: Réduit l'affichage.

Esc: Quitte l'édition.

Delete: Permet d'effacer des blocs entiers de programme.

Mode debug

- Dans l'éditeur en mode F2.
- F8: Passage en mise au point.
- F7: Forçage des variables. Espace permet le forçage de la variable.

Menu Transfert

PLC GPP Other Run/Stop Setup Eprom A7BDE

PLC: Permet de transférer le programme de l'ordinateur (medoc) vers l'automate (PLC)

Run/Stop: Force l'automate dans un de ces modes (pas tjrs disponible)

Setup: Change la façon de communiquer entre l'automate et l'ordinateur.

Si le transfert ne s'effectue pas, vérifier:

1. que l'automate est allumé.
2. qu'ils sont bien reliés entre eux.
3. que le Setup au niveau du port de communication est bien réglé (com1 ou com2 en fonction du branchement sur l'ordinateur)

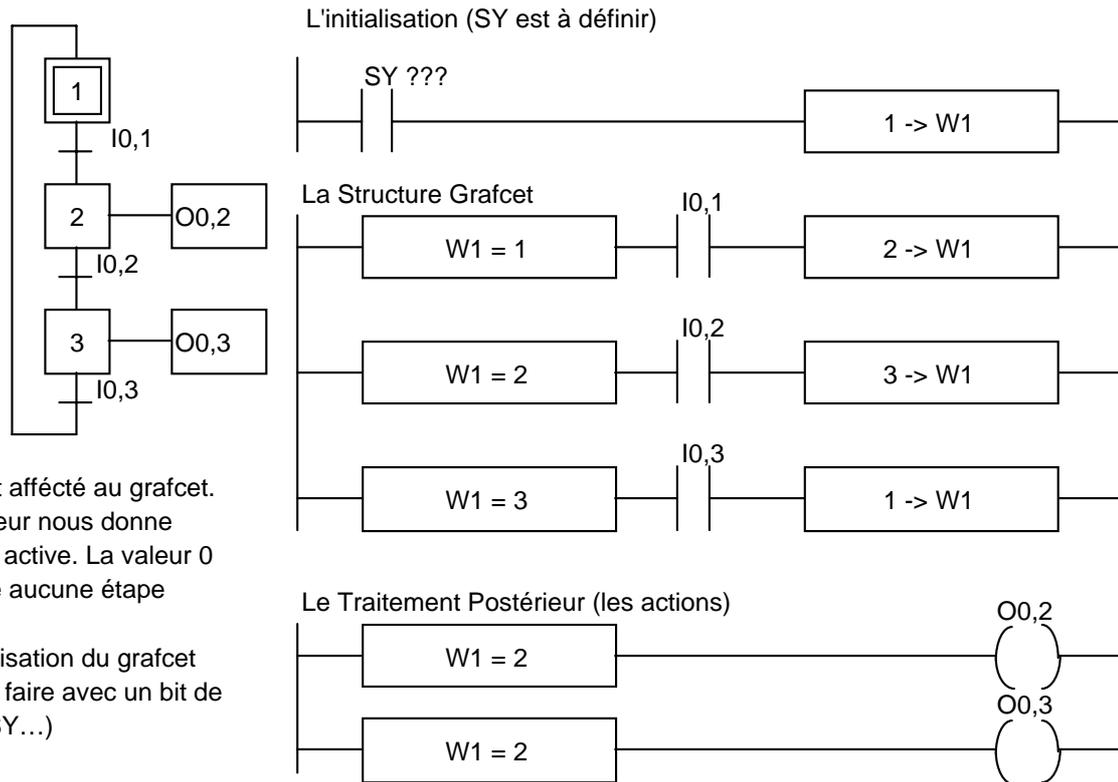
Menu Quit

Permet de quitter

La procédure de Démarrage rapide:

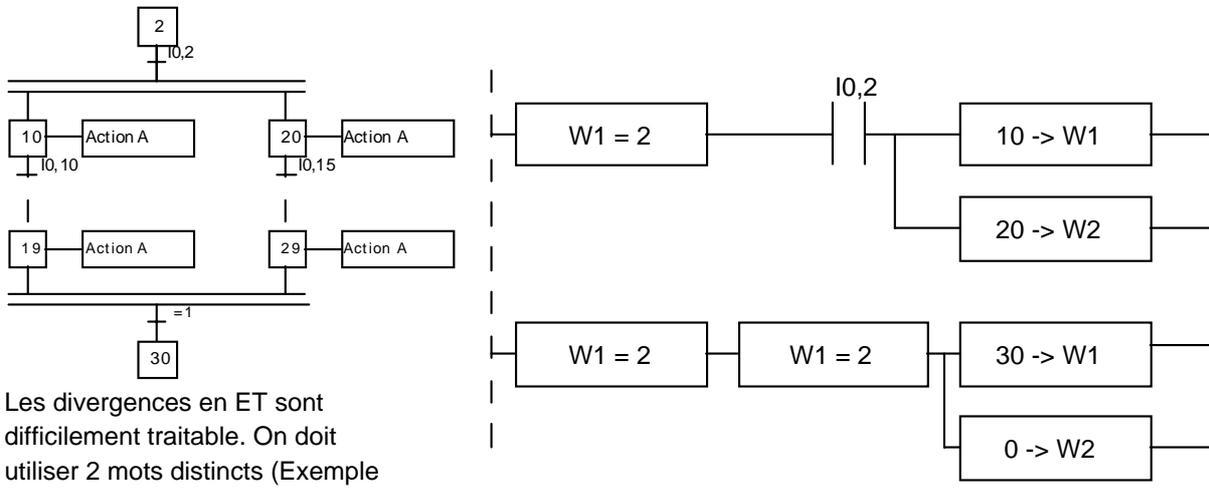
1. Donner un nom à votre projet, Menu New_Proj
 2. Passer dans l'éditeur, menu Ladder (touche 2 pour accès rapide)
 3. Rentrer dans l'éditeur: F2 (alt H pour compresser l'écriture à l'écran) puis F7.
 4. Taper votre programme en Ladder: Touches 1 à 9.
 5. Taper End en fin de programme, à taper directement.
 6. Sortir de l'éditeur, F7 puis F2.
 7. Transférer votre programme, menu Transfer (touche 5 pour accès rapide)
 8. Mettre l'automate en Run
 9. Tester votre programme à partir de l'écriture Ladder. (touche 2)
 10. puis F2 et F8 pour rentrer dans le mode de visualisation en temps réel, le mode Moniteur (équivalent à mémoire TSX mise au point sur PL7_2)
 11. Forcer les variables à 1 si besoin en monitoring (touche F7 puis espace)
- Les affectations des Entrées (X...)/ Sorties (Y...) dépendent des Automates, ainsi que les bits mémoires (M...) et les mots (D...).
- (Les automates disponibles F1, FX0N, FX ,A1S, A1.)

La méthode des Mots



Notes:
 W1 est affecté au grafcet.
 Sa valeur nous donne l'étape active. La valeur 0 signifie aucune étape active.
 L'initialisation du grafcet doit se faire avec un bit de Run (SY...)

Difficulté:
 Si $W1=2$ et $I0,3=1$ lorsque $I0,2$ passe à 1, on passe directement de l'étape 2 à l'étape 1 en 1 temps de cycle. L'action $O0,2$ n'aura pas lieu. Règle de grafcet non respecté. Cela peut provoquer des problèmes sur des grafcets particulier.



Les divergences en ET sont difficilement traitable. On doit utiliser 2 mots distincts (Exemple ci contre avec W1 et W2)

Les divergences en OU ne pose pas de problèmes particulier.

Document constructeur

Schneider automate

TSX 17

TSX Micro

Variateur de vitesse

ATV 08

Mitsubishi

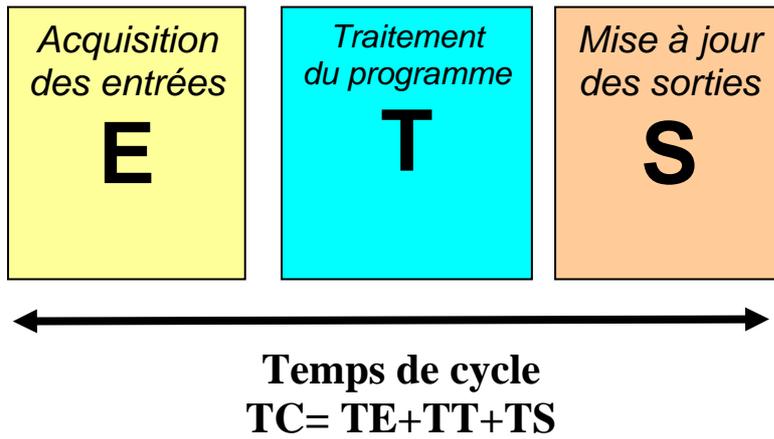
Front montant sur FX0N : Leading and trailing pulse (page 2-20)

Bits système FX0N : PC status (Page 6-2)

Base de temps pour les temporisations : Performance et spécification FX0N (Page 8-4)

Fonctionnement d'un automate

PowerPoint associé : Les API



Des capteurs aux actionneurs schéma de principe

