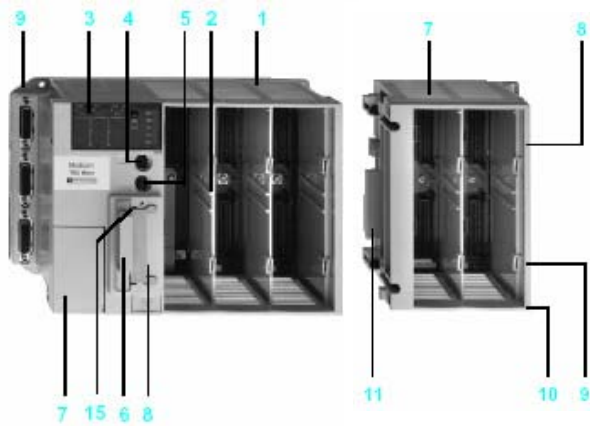


Mise en œuvre des automates TSX Micro

I. Présentation matérielle :

1. Base automate et mini-bac d'extension :

Les automates TSX 37-21/22 et le mini bac d'extension TSX RKZ 02 comprennent :

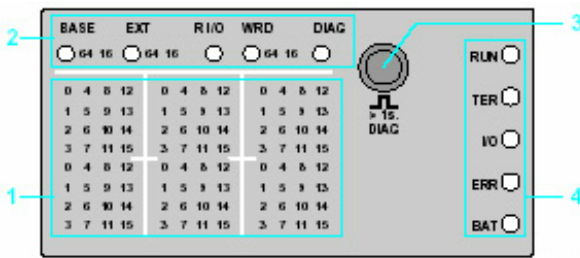


- 1 un bac de base à 3 emplacements disponibles (positions 1 à 6).
- 2 un emplacement réservé à un module au format standard.
- 3 un bloc de visualisation centralisée.
- 4 une prise terminal repérée TER.
- 5 une prise de dialogue opérateur repérée AUX.
- 6 un emplacement pour une carte d'extension mémoire.
- 7 une trappe d'accès aux bornes d'alimentation.
- 8 un emplacement pour un coupleur de communication.
- 9 des connecteurs pour les fonctions analogiques et comptage intégrées pour TSX 37-22.
- 10 un mini bac d'extension à 2 emplacements disponibles (positions 7 à 10).
- 11 un voyant de présence de tension à 24 V.
- 12 des bornes d'alimentation protégées par un cache amovible, pour le raccordement d'une alimentation auxiliaire à 24 V dans le cas des automates alimentés en ~ 100/240 V.
- 13 une borne de masse.
- 14 des connecteurs de raccordement à l'automate de base.
- 15 un bouton de réinitialisation.

2. Bloc de visualisation :

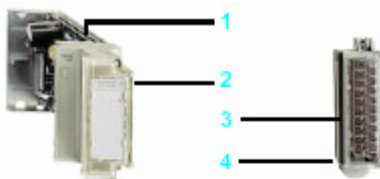
Celui-ci comprend :

1 Trois ensembles de 32 voyants (DEL) représentant les emplacements des modules implantés dans le bac de base ou le mini bac d'extension.

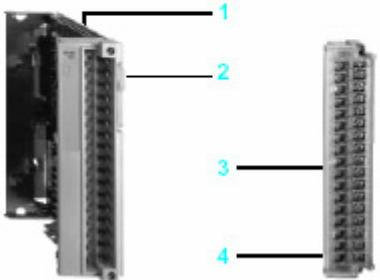


- 2 Une ligne d'information formée de voyants (DEL) signalant les modes de fonctionnement de la visualisation.
- 3 Un bouton-poussoir de commande donnant accès aux différents modes de fonctionnement de la visualisation.
- 4 Cinq voyants (DEL) :
 - RUN, marche/arrêt de l'automate
 - TER, trafic sur la prise terminal
 - I/O, défaut d'entrées/sorties
 - ERR, défaut processeur ou application
 - BAT, défaut ou absence de pile.

3. Modules :



Module d'E/S au demi-format



Module d'E/S au format standard

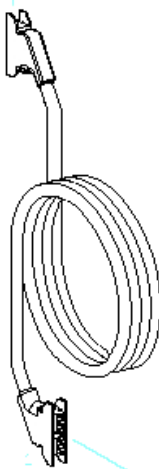
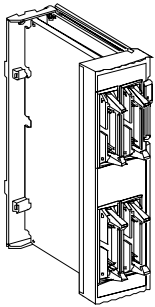
Les modules d'entrées/sorties à raccordement par bornier comprennent :

- 1 un corps métallique rigide.
- 2 un système de verrouillage pour fixation du module dans son emplacement. Ce système est accessible uniquement lorsque le bornier à vis est démonté.
- 3 un bornier à vis débrochable pour le raccordement aux capteurs et préactionneurs
- 4 un volet d'accès aux vis du bornier servant également de support à l'étiquette de repérage.

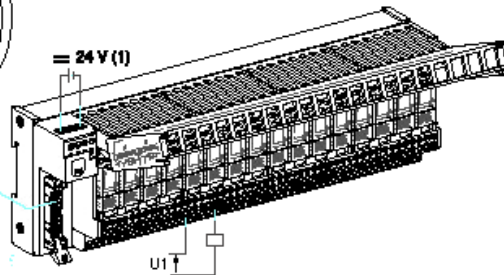


*Les modules peuvent réaliser les fonctions suivantes : Entrées et Sorties TOR, Entrées et Sorties ANA, Comptage, Communication (Ethernet, Modbus, ...), Surveillance d'arrêts d'urgence.
Mettre un cache TSX RAK 01 dans les emplacements vides.*

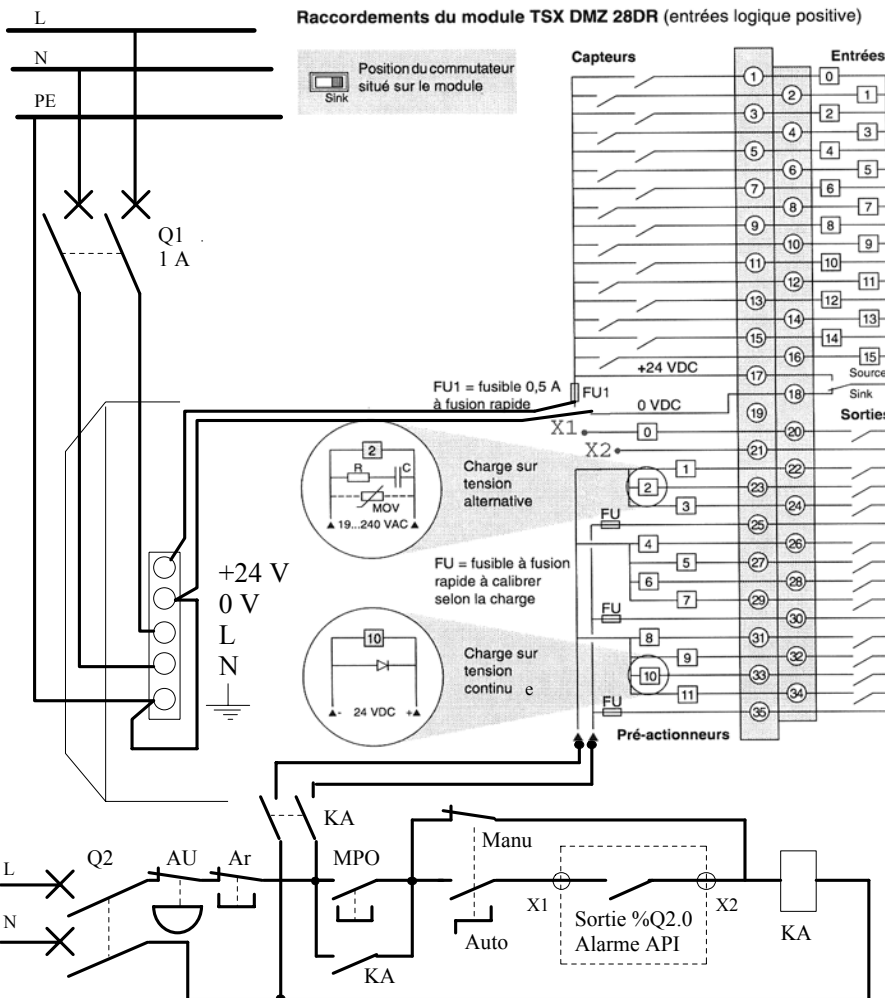
4. Système de connectique TELEFAST :



Pour ce système les modules sont équipés de connecteurs HE10. Il faut alors prévoir des borniers de raccordement déportés et des câbles préfabriqués pour connecteurs haute densité.



5. Raccordement des bases automates et des modules :



Il existe des bases à alimenter en 24 V courant continu. Chaque module nécessite un raccordement spécifique. Toujours se reporter à la documentation constructeur. Placer des fusibles de calibre adapté.

Les modules de sortie à relais doivent être équipés sur chaque pré-actionneurs d'une protection R-C ou d'une diode suivant la nature de la tension d'alimentation du circuit.

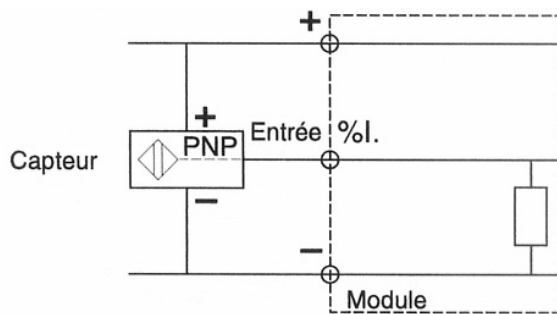
Il est conseillé de réaliser l'asservissement des alimentations :

- Mise sous tension de la base et des entrées, par fermeture de Q1
- Si Q2 fermé et API en RUN mise sous tension des sorties par appui sur MPO. Intérêt ?

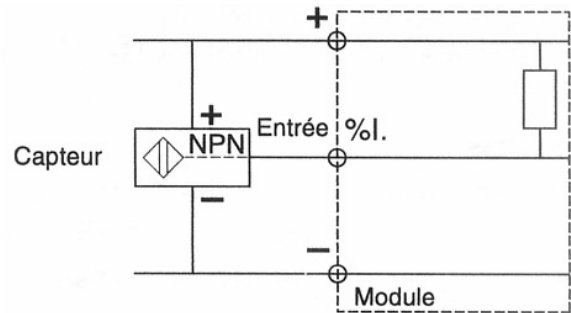
6. Raccordement des capteurs inductifs ou photoélectriques :

- Tous les détecteurs de proximité inductifs ou capacitifs et les détecteurs photoélectriques de type 3 fils PNP, fonctionnant sous une tension de 24V CC, sont compatibles avec toutes les entrées à logique positive. Ceux de type 3 fils NPN avec toutes les entrées à logique négative.

Schémas :

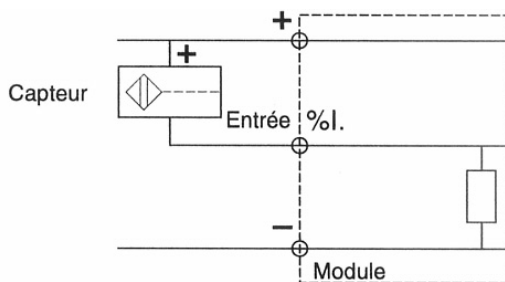


Capteurs 3 fils et entrées logique positive (sink)



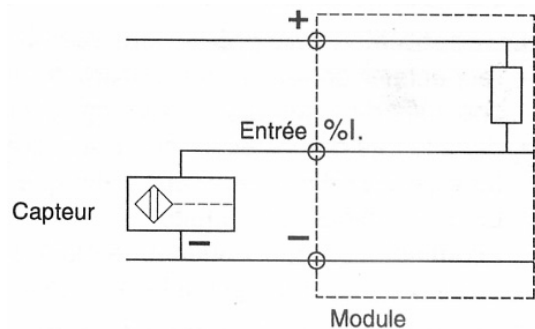
Capteurs 3 fils et entrées logique négative (source)

- Tous les détecteurs de proximité ou autres capteurs de type 2 fils, fonctionnant sous une tension de 24V CC et ayant les caractéristiques ci-dessous sont compatibles avec toutes les entrées 24V CC à logique positive ou négative. Tension de déchet à l'état fermé : $\leq 7V$. Courant commuté minimal : $\leq 2,5 \text{ mA}$. Courant résiduel à l'état ouvert : $\leq 1,5 \text{ mA}$.



Schémas:

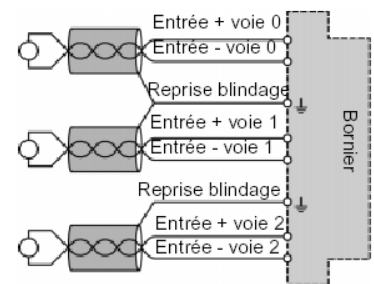
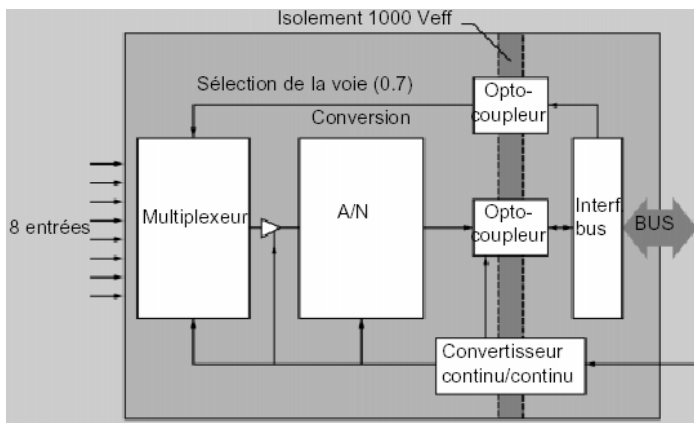
Capteurs 2 fils et entrées logique positive.



Capteurs 2 fils et entrées logique négative

7. Raccordement des entrées et sorties analogiques :

Par conversion analogique/numérique les modules d'entrées analogiques permettent de connaître le niveau d'une tension ou d'un courant image d'une température, d'une pression, d'un débit, d'une vitesse acquise sur un système.



Par conversion numérique/analogique les modules de sorties analogiques transmettent au système un niveau de tension ou de courant image d'une vitesse désirée, d'une température à atteindre etc.

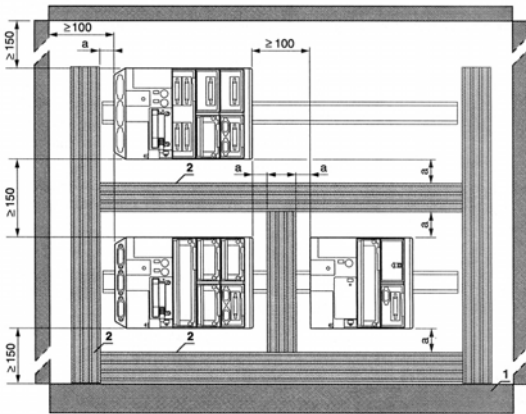
Respecter les règles de précaution pour le branchement de ces modules ou des entrées et sorties intégrées à la base :

Respecter les règles de précaution pour le branchement de ces modules ou des entrées et sorties intégrées à la base :

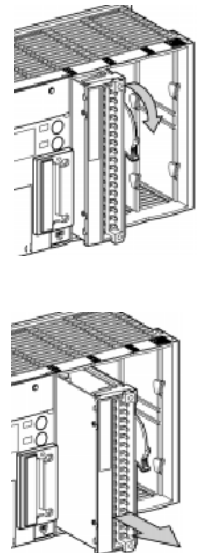
- Raccorder le blindage aux bornes de reprise de terre.
- Eloigner les fils véhiculant des signaux bas niveaux de ceux de puissance.

- Utiliser des paires torsadées blindées.

8. Mise en place et implantation :



Les bases peuvent être montées dans des coffrets sur profilé DIN, platine etc. Leur position doit faciliter le refroidissement par convection naturelle, il faut respecter les espaces recommandés par le constructeur. Il est conseillé d'installer les appareils générateurs de chaleur (transformateurs, modules d'alimentation, contacteurs de puissance, etc.) au-dessus des automates. Il faut prévoir suffisamment d'espace autour des accès à l'automate pour les branchements, basculement des trappes,



passage des câbles etc.

Prendre note des méthodes de mise en place et de démontage des modules et autres accessoires dans la documentation.

Pour la sécurité des personnes, il est obligatoire de relier les masses prévues au conducteur de protection.

II. Mise en œuvre du logiciel PL7 Pro :

Après lancement du logiciel vous devez préciser :

- le processeur et la version de l'API à mettre en œuvre,
- l'utilisation de l'outil Grafcet
- l'installation éventuelle d'une mémoire.

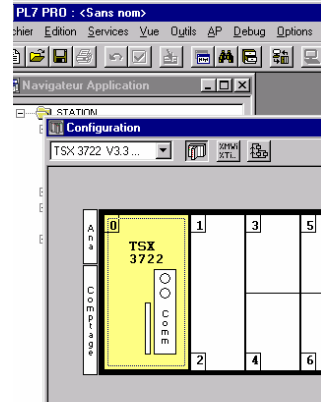
Toutes les configurations ci-dessous doivent être confirmées par appui sur l'icône :



1. Configuration du processeur :

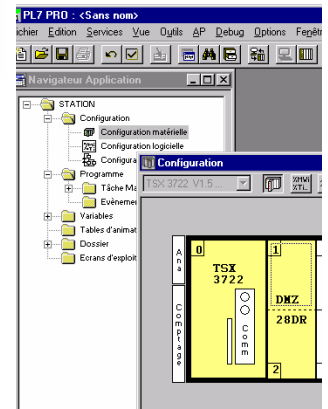
Les déclarations suivantes peuvent être faites à partir du dossier configuration matérielle du navigateur application en double-cliquant sur le processeur de l'API (position 0) :

- L'entrée %I1.8 peut être paramétrée pour commander le passage RUN/STOP de l'automate.
- La sortie %Q2.0 peut être affectée à une fonction d'alarme.
- Démarrage automatique en RUN, cocher la case correspondante.
- Si l'API dispose d'une mémoire EPROM on peut cocher : Sauvegarde / Restitution des mots internes %MWi.
- Sélectionner le type d'exécution de la tâche MAST Cyclique ou Périodique (3 à 255 ms).
- Saisir la valeur du chien de garde de la tâche MAST: 10 à 500 ms maximum (250 ms, valeur par défaut).
- Saisir suivant l'application la valeur de la période de la tâche FAST (2 à 255 ms maximum)...

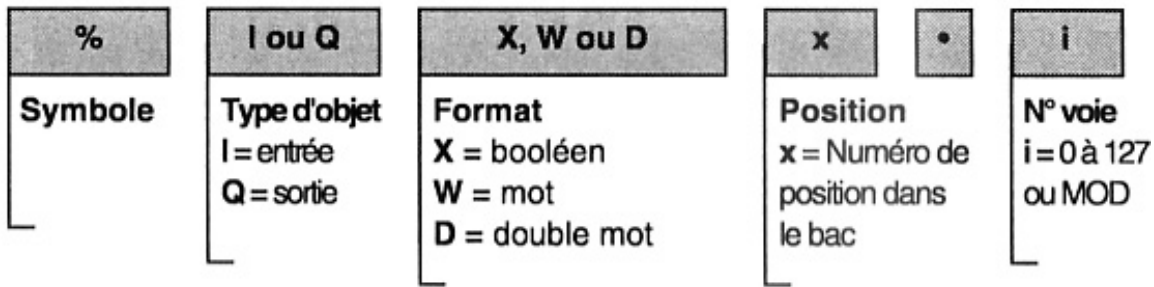


2. Choix et configuration des modules :

- Double-cliquer sur la position du rack à configurer.
- Sélectionner dans le champ Famille le type de module (par exemple Analogique, sauf positions 1 et 2).
- Sélectionner dans le champ Module la référence du module à configurer (par exemple TSX AEZ 801).
- Valider par OK, le module est déclaré dans sa position (celle-ci est tramée et contient la référence du module).
- La configuration de chaque module peut-être modifiée par double-clic sur le module : type d'entrée positive (Sink) ou négative (Source), surveillance défaut d'alimentation, affectation des entrées aux tâches, filtrage des entrées, fonction, repli des sorties etc.



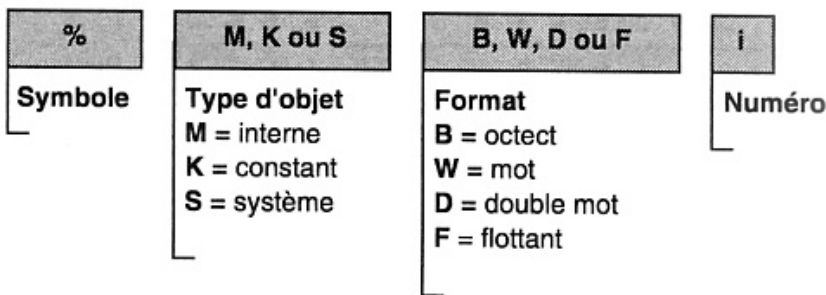
3. Adressage des objets de modules d'entrées sorties :



Exemples d'objets :

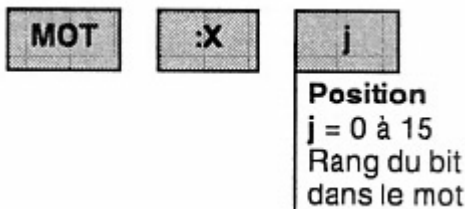
- %I1.6** : Bit de la voie d'entrée n° 6 du module d'entrées TOR placé en position 1 dans le bac.
- %IW3.5** : Mot de la voie d'entrée n° 5 du module d'entrées analogiques placé en position 3.
- %Q2.8** : Bit de la voie de sortie n° 8 du module de sortie TOR placé en position 2.
- %QW4.3** : Mot de la voie de sortie n° 3 du module de sortie analogique placé en position 4.
- %Ix.MOD.ERR** : Bit. Il indique lorsqu'il est à l'état 1 que le module situé en position x est en défaut.
- %Ix.i.ERR** : Bit, à l'état 1 il indique que la voie d'entrée i du module situé en position x est en défaut.

4. Adressage des bits et mots:



Exemples :

- %MB5** : Mot interne variable de 8 bits, numéro 5.
- %KW12** : Mot constant de 16 bits, numéro 12.
- %SW49 à %SW53** : Mots systèmes contenant la date et l'heure courante en BCD.
- %M8** : Bit interne, numéro 8.
- %S6** : Bit système, son état change toutes les secondes.



Exemples :

- %MW10:X4** : Bit n° 4 du mot interne **%MW10** de 16 bits.
- %QW5.1:X10** : Bit n° 10 du mot de la voie 1 du module de sortie analogique placé en position 5.
- %SW60:X13** : Bit du mot système diagnostic automate. A l'état 1 indique le mode Run de l'automate.

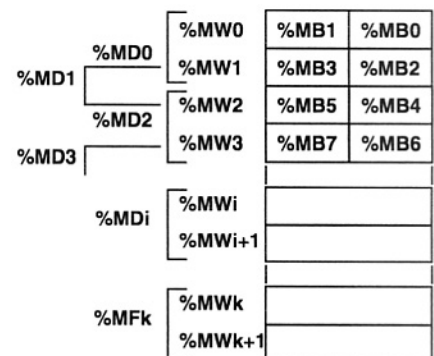
ATTENTION :

Il y a quelque fois nécessité de laisser des écarts suffisants entre objets pour qu'il n'y ait pas interférence quand un recouvrement n'est pas souhaité.

Exemple :

%MW1 correspond au mot de poids faible du double mot **%MD1** et au mot de poids fort du double mot **%MD0**.

Le double mot constant **%KD543** est constitué des mots constants **%KW543** (poids faible) et **%KW544** (poids fort), mais le mot **%KW543** est aussi le poids fort du double mot **%KD542** et **%KW544** est aussi le poids faible du double mot **%KD544**.



5. Autres adressages :

- **%X12** : Bit indiquant l'état de l'étape 12 du GRAFCET. **%X12.T** : mot, temps d'activité de l'étape 12.
- **%TM2.V** : Mot représentant la valeur courante de la temporisation du bloc fonction temporisateur n° 2.
- **%TM3.Q** : Bit de sortie indiquant une fin de temporisation du bloc fonction temporisateur n° 3.
- **%C5.P** : Mot représentant la valeur de présélection du bloc fonction compteur-décompteur n° 5.
- **%C6.F** : Bit de sortie indiquant un débordement du bloc fonction compteur-décompteur n° 6.

6. Editeur de variables :

Dans l'éditeur de variables du navigateur application il est possible de :

- Symboliser les différents objets de l'application (colonne symbole). Leur identification dans la programmation en sera facilité. Repère **%Q2.4** symbole **Mot_vent**.
- Paramétrer les blocs fonctions prédéfinis temporisateurs, compteurs, etc.
- Saisir les valeurs des mots constants (colonne valeur) après avoir coché la case paramètres et choisi la base d'affichage (décimal, binaire, hexadécimal, flottant, message).

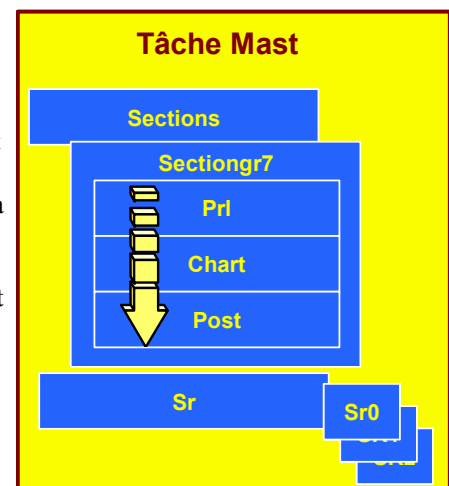
VARIABLES							
<input checked="" type="checkbox"/> Paramètres	FB PREDEFINIS	TM	<input type="checkbox"/> Zone de saisie				
Repère	Symbole	Preset	Mode	TB	Reg	Commentaire	
%TM0	Temps_d_accélération	200	TP	1 s	<input checked="" type="checkbox"/>	Butée maximum à ne pas dépasser	
%TM1	Temps_de_décélération	1000	TOF	1 s	<input checked="" type="checkbox"/>		
%TM2	Surveillance_fin_de_course	50	TP	100 ms	<input checked="" type="checkbox"/>		
%TM3		9999	TON	1 mn	<input checked="" type="checkbox"/>		
%TM4		9999	TON	1 mn	<input checked="" type="checkbox"/>		

7. Editeur de programme :

La structure logicielle d'une application peut-être **monotâche** elle est alors associée à une seule tâche utilisateur avec d'éventuels sous-programmes : la tâche **MAST**.

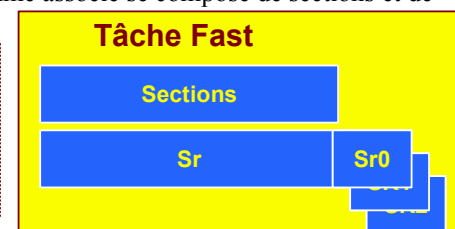
Exemple de structure monotâche avec Grafcet.

- La flèche indique l'ordre d'exécution des 3 modules Prl, chart et Post.
- **Prl** : Traitement préliminaire. Le traitement préliminaire est scruté dans sa totalité du haut vers le bas. Exécuté avant les traitements séquentiel et postérieur, il permet de traiter tous les événements ayant une influence sur ces derniers : gestion des reprises secteur et réinitialisations, remise à zéro ou prépositionnement des graphes. C'est donc uniquement dans le traitement préliminaire qu'il faut agir sur les bits associés aux étapes (mise à 0 ou à 1 des bits étapes %Xi ou %Xi.j par les instructions SET et RESET).
- **Chart** : Pages Grafcet (8 pages). Le Grafcet permet de représenter graphiquement et de façon structurée le fonctionnement d'un automatisme séquentiel.
- **Post** : Traitement postérieur. Il est exécuté après le Grafcet. C'est ici qu'il est recommandé de programmer l'écriture des sorties %Qi.x, pour être certain qu'elle n'a été effectuée qu'une fois pour chacune d'elles.
- **Sr** : Sous-programme. Les modules sous-programmes se programment comme les précédents en tant qu'entités séparées en : langage à contacts, liste d'instructions ou littéral structuré. Les appels aux sous-programmes s'effectuent dans les sections ou depuis un autre sous-programme (8 niveaux d'imbrications maximum). Les sous-programmes sont aussi liés à une tâche, un même sous-programme ne peut pas être appelé depuis plusieurs tâches.

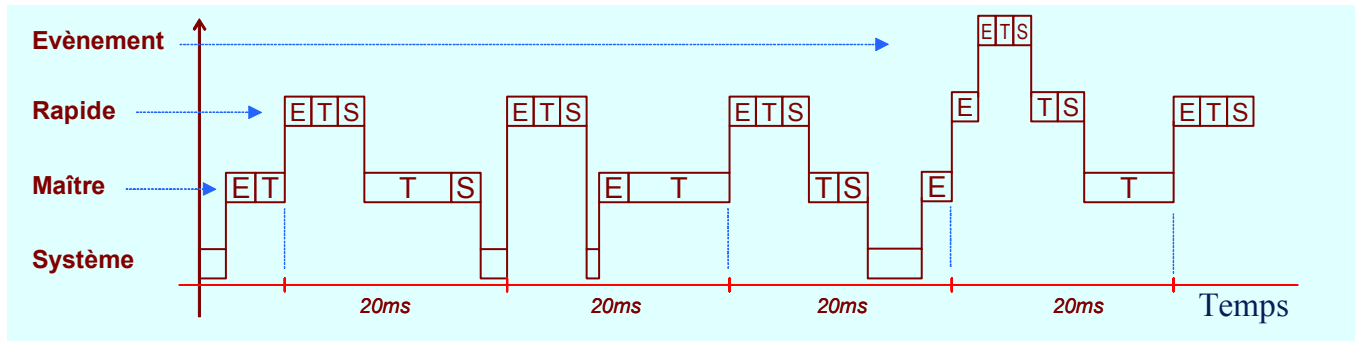


Pour une structure **multitâche** deux autres tâches peuvent être rajoutées à la tâche maître :

- La tâche rapide (**Fast**) : 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.
- Les tâches événementielles (**Evt**). Les traitements événementiels permettent de réduire le temps de réaction du logiciel sur des événements de commande en provenance de certains modules métiers. Ces traitements sont exécutés en priorité sur toutes les autres tâches. Elles conviennent donc aux traitements demandant des délais de réactions très courts par rapport à l'arrivée de l'événement.



Exemple de séquençement d'un traitement multitâche:



Légende :

E : acquisition des entrées ; T : traitement du programme ; S : mise à jour des sorties.

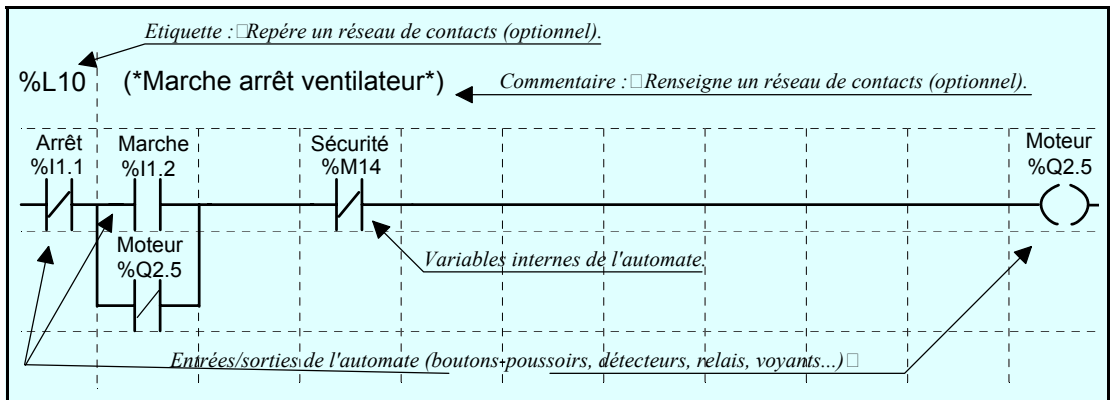
On remarque la périodicité de la tâche rapide de 20 ms et la haute priorité de la tâche événementielle sur toutes les autres tâches. Attention au temps de prise en compte du changement d'une entrée

8. Description des langages PL7 :

Le logiciel de programmation PL7 permet de programmer un automate programmable conformément au standard CEI 1131-3.p à l'aide des langages suivants :

- **LD, langage à contacts ou Ladder** : Un programme écrit en langage à contacts se compose d'une suite de réseaux exécutés

séquentiellement par l'automate. Dessiné entre deux barres de potentiel, un réseau est un ensemble d'éléments graphiques comme ceux de la figure ci-contre et également des fonctions d'automatismes (temporisateurs, compteurs...), des opérations arithmétiques, logiques et spécifiques.



- **IL, langage liste d'instructions** : Un programme écrit en langage liste d'instructions se compose d'une suite d'instructions exécutées séquentiellement par l'automate. Chaque instruction est composée d'un code instruction et d'un opérande. Le code instruction détermine l'opération à exécuter. Il existe 2 types de codes instructions **le test**, dans laquelle figurent les conditions nécessaires à une action (ex : LD, AND, OR...), et l'**action** qui sanctionne le résultat consécutif à un enchaînement de test. (ex : ST, STN, R, ...). Une instruction agit sur un opérande. Cet opérande peut être : une entrée/sortie de l'automate (boutons-poussoirs, détecteurs, relais, voyants...), une fonctions d'automatisme (temporisateurs, compteurs...), une opération arithmétique et logique ou une opération de transfert, une variable interne de l'automate.

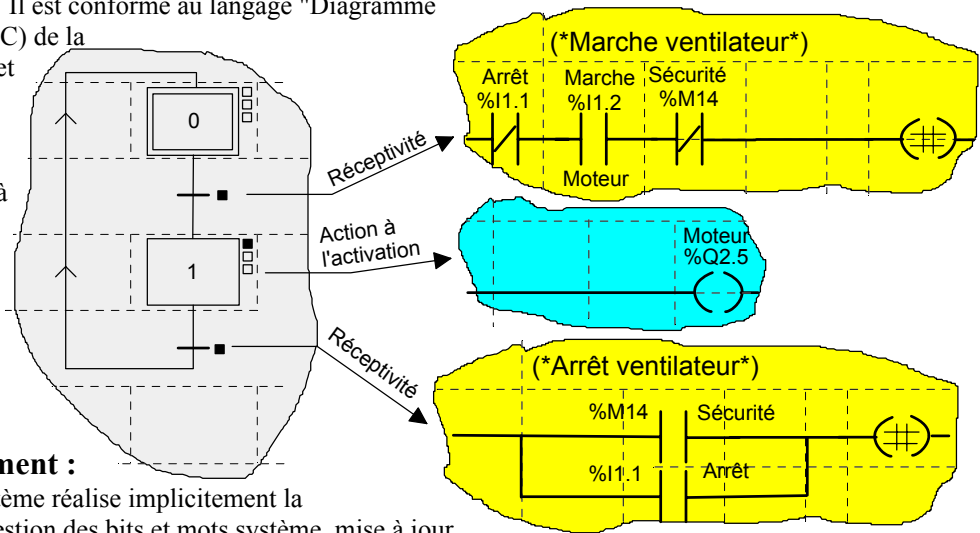
! (*Marche arrêt ventilateur*)	
LDN	%I1.1
AND(%I1.2
OR	%Q2.5
)	
ANDN	%M14
ST	%Q2.5

```
! %L10: (*Marche arrêt ventilateur*)
IF %I1.1 AND(%I1.2 OR %Q2.5)AND NOT %M14 THEN
SET %Q2.5;
END IF;
```

- **ST, langage littéral structuré** : C'est un langage évolué de type algorithmique particulièrement adapté à la programmation de fonctions arithmétiques complexes, manipulations de tableaux et gestions de messages. Il permet la réalisation de

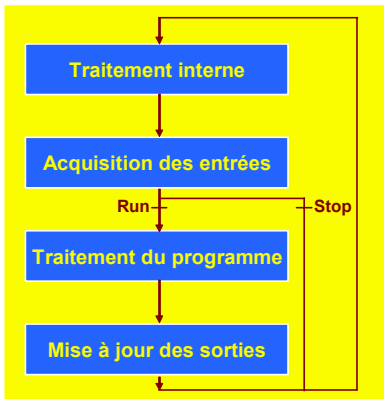
programmes par écriture de lignes de programmation, constituées de caractères alphanumériques. Une section de programme littéral est organisée en phrases. Une phrase littérale est l'équivalent d'un réseau de contacts en langage à contacts.

▪ **Chart, le langage Grafset** : Il est conforme au langage "Diagramme fonctionnel en séquence" (SFC) de la norme IEC 1131-3. Le Grafset permet de représenter graphiquement et de façon structurée le fonctionnement d'un automate séquentiel à l'aide d'étapes et transitions comportant éventuellement des divergences et convergences ET, OU.



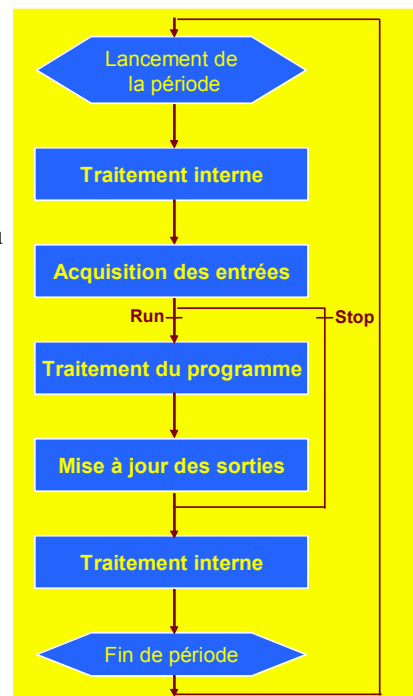
III. Modes de fonctionnement :

▪ **Traitement interne** : Le système réalise implicitement la surveillance de l'automate (gestion des bits et mots système, mise à jour des valeurs courantes de l'horodateur, mise à jour des voyants d'état, détection des passages RUN/STOP..) et le traitement des requêtes en provenance du terminal (modifications et animation).



Exécution cyclique

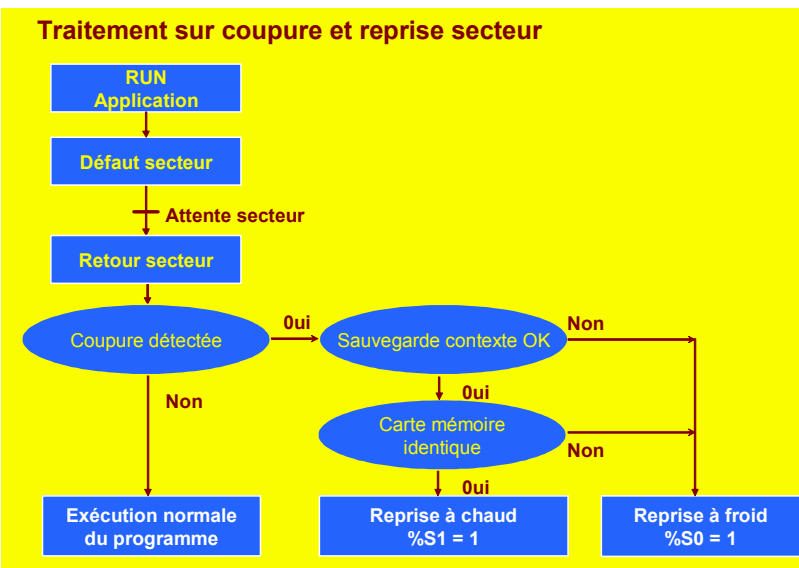
- **Acquisition des entrées** : Ecriture en mémoire de l'état des informations présentes sur les entrées des modules TOR et métier associées à la tâche.
- **Traitement du programme** : Exécution du programme application, écrit par l'utilisateur.
- **Mise à jour des sorties** : Ecriture des bits ou des mots de sorties associés aux modules TOR et métier associés à la tâche selon l'état défini par le programme application avec ou sans Grafset. Les fonctions de régulation doivent être programmées dans une tâche périodique.



Exécution périodique

IV. Traitement sur coupure et reprise secteur :

L'illustration ci-dessous présente les différentes reprises secteurs détectées par le système. Si la durée de la coupure est inférieure au temps de filtrage de l'alimentation (environ 10 ms pour les alimentations alternatives ou à 1 ms pour les alimentations continues), celle-ci n'est pas vue par le programme qui s'exécute

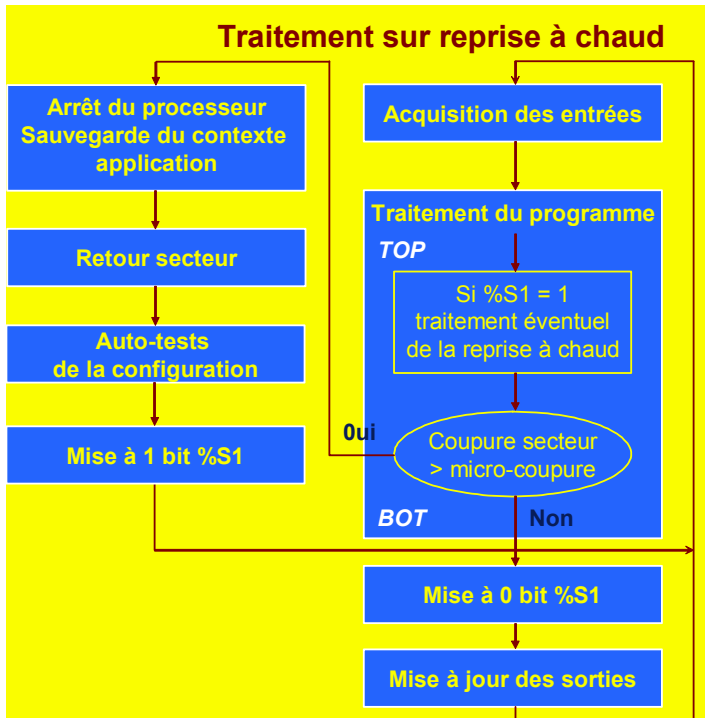


normalement.

Lors de la coupure secteur, le système mémorise le contexte application et l'heure de la coupure. Il positionne toutes les sorties à l'état repli (état défini par configuration).

A la reprise secteur, le contexte sauvegardé est comparé à celui en cours; ce qui définit le type de démarrage à exécuter. Si le contexte application a changé (perte du contexte système ou nouvelle application), l'automate effectue l'initialisation de l'application : démarrage à froid. Si le contexte application est identique, l'automate effectue une reprise sans initialisation des données : reprise à chaud.

- **Traitement par programme de la reprise à chaud** : Une reprise à chaud peut être provoquée par :

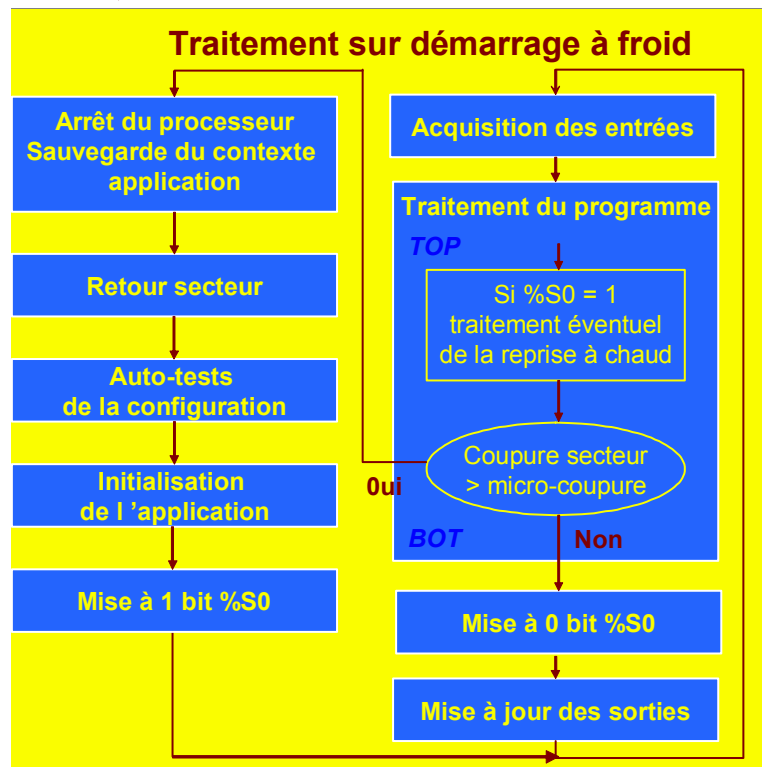


Une reprise à chaud peut être provoquée par : Une reprise secteur sans perte du contexte, mise à 1 par programme du bit système %S1 depuis PL7 par terminal, action sur bouton RESET du module alimentation du rack 0. L'exécution du programme reprend à partir de l'élément où a eu lieu la coupure secteur, sans mise à jour des sorties. A la fin du cycle de reprise, le système effectue l'initialisation des files de messages et d'événements, l'envoi des paramètres de configuration à tous les modules d'entrées/sorties TOR et métiers, la désactivation de la tâche rapide et des traitements événementiels (jusqu'à la fin du premier cycle de la tâche maître). Le système effectue un cycle de reprise dans lequel il reprend en compte l'ensemble des modules d'entrées, relance la tâche maître avec les bits %S1 (reprise à chaud) et %S13 (premier cycle en RUN) positionné à 1, remet à l'état 0 les bits %S1 et %S13 en à la fin de ce premier cycle de la tâche maître, réactive la tâche rapide et les traitements événementiels à la fin de ce premier cycle de la tâche maître. En cas de reprise à chaud, si vous désirez un traitement particulier vis-à-vis de l'application, vous devez écrire le programme correspondant sur test de %S1 à 1 en début de programme de la tâche maître.

- **Traitement par programme de la**

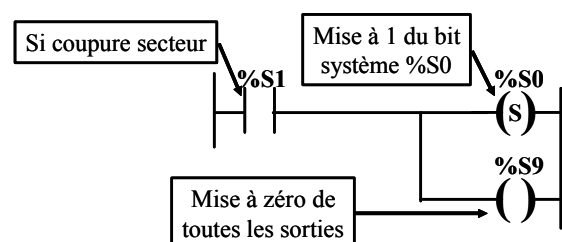
reprise à froid : Les différentes causes possibles d'un démarrage à froid sont : Chargement d'une application, action sur bouton RESET du processeur, action sur bouton RESET du processeur suite à un défaut bloquant, manipulation du préhenseur ou insertion/extraction d'une carte mémoire PCMCIA, initialisation depuis PL7 Junior ou PL7 Pro, forçage du bit système %S0, reprise après une coupure de l'alimentation avec perte du contexte.

L'exécution du programme reprend en début de cycle. Le système effectue : la remise à zéro des bits, de l'image des E/S, et des mots internes (si l'option RAZ des %MW sur reprise à froid est cochée dans l'écran configuration du processeur). Si l'option RAZ des %MW n'est pas active et si des mots internes %MWi sont sauvegardés dans la mémoire interne Flash EPROM (TSX 37), ceux-ci sont restitués lors d'un démarrage à froid, l'initialisation des bits et mots système, l'initialisation des blocs fonction à partir des données de configuration, la désactivation des tâches, autres que la tâche maître, jusqu'à la fin du premier cycle de la tâche maître, le positionnement du Graftec sur les étapes initiales, l'annulation des forçages, des files de message et d'événements, l'envoi des paramètres de configuration à tous les modules d'entrées/sorties TOR et modules métier.



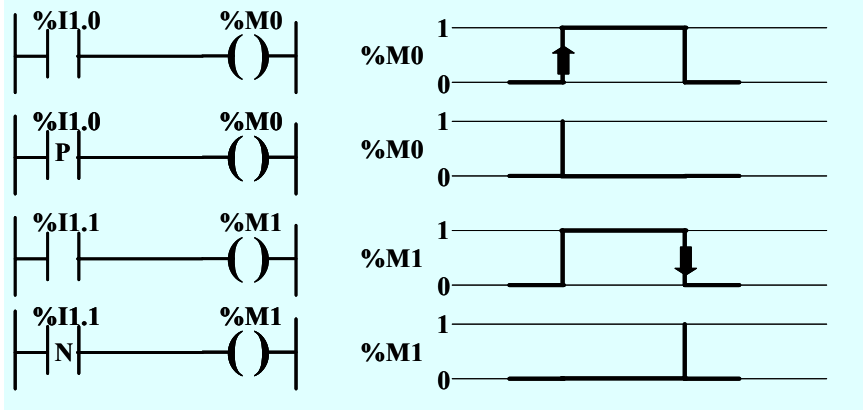
Pour ce premier cycle de reprise le système effectue : la relance de la tâche maître avec les bits %S0 (reprise à chaud) et %S13 (premier cycle en RUN) positionné à 1, le mot %SW10 (détection d'une reprise à froid lors du premier tour d'une tâche) est mis à 0, la remise à l'état 0 des bits %S0 et %S13, et la remise à 1 de chaque bit du mot %SW10 à la fin de ce premier cycle de la tâche maître, l'activation la tâche rapide et les traitements événementiels à la fin de ce premier cycle de la tâche maître.

- **Traitement de la reprise secteur** : En cas de reprise à chaud, si l'on souhaite réinitialiser toutes les variables lors du retour du secteur, le bit %S0 doit être positionné à 1 par programme par le bit %S1. Réserver une section de programme en préliminaire nommée Reprise-Secteur.



V. Fronts montants et descendants :

Seules les variables %I, %Q et %M peuvent être testées sous la forme de fronts montants ou descendants. Dans tous les autres cas, il est faut associer à la variable un bit interne.



VI. Bloc fonction

Quel que soit le type de bloc fonction utilisé, il doit obligatoirement être relié en entrée à la barre de potentiel gauche, en direct ou à travers d'autres éléments graphiques.

Sorties "en l'air" : il n'est pas nécessaire de relier à d'autres éléments graphiques les sorties des blocs fonction.

Sorties testables : les sorties des blocs fonction sont accessibles à l'utilisateur sous forme d'objet bit.

Les variables internes de blocs et les sorties graphiques sont des objets exploitables à distance depuis une autre partie du programme.

Les entrées non câblées des blocs fonction standard sont mises à 0.

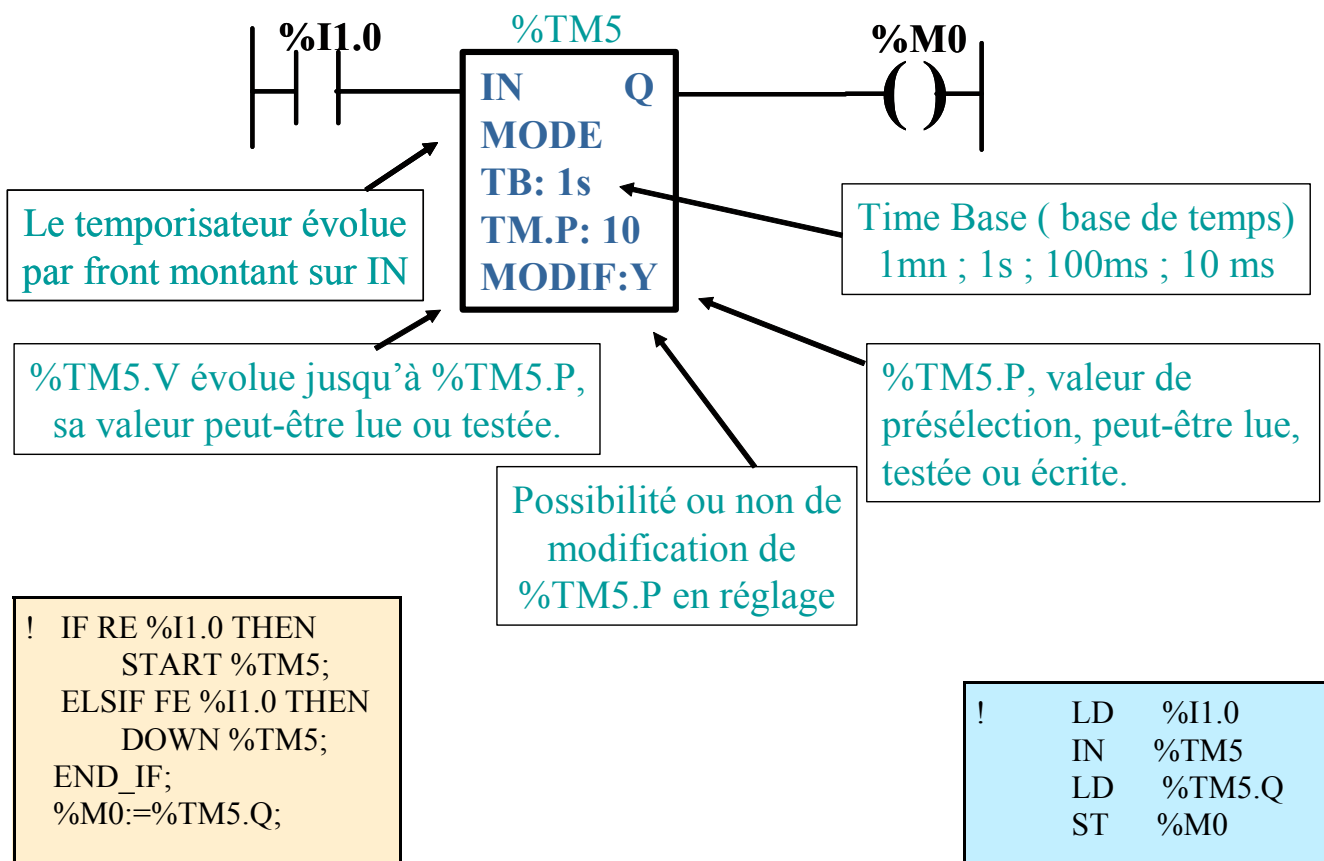
VII. Bloc fonction temporisateur :

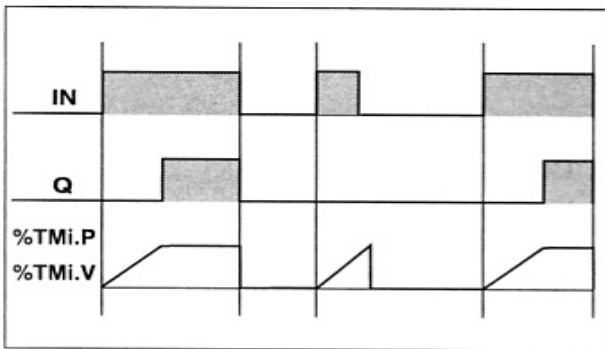
Le temporisateur possède 3 modes de fonctionnement :

- **TON** : permet de gérer des retards à l'enclenchement,
- **TOF** : permet de gérer des retards au déclenchement,
- **TP** : permet d'élaborer une impulsion de durée précise (monostable).

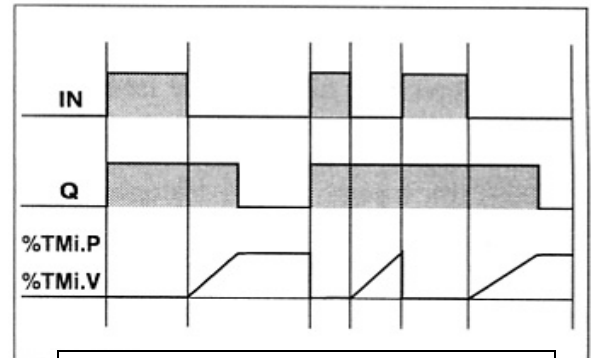
La configuration consiste à déterminer les paramètres suivants :

- **Mode** : TON, TOF ou TP ; **TB** : 1 min, 1s, 100 ms ou 10 ms ; **%TMi.P** de 0 à 9999 ; **MODIF** : Y ou N

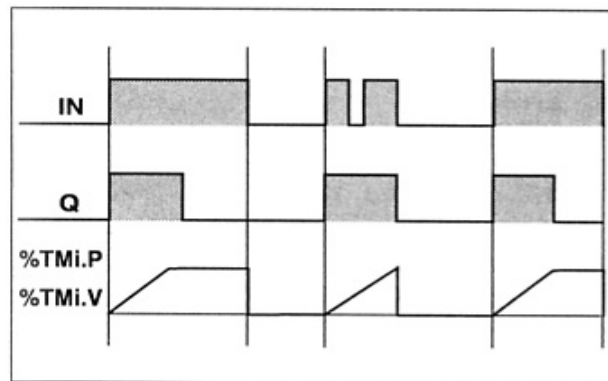




Retard à l'enclenchement ou retardé à l'action : mode TON



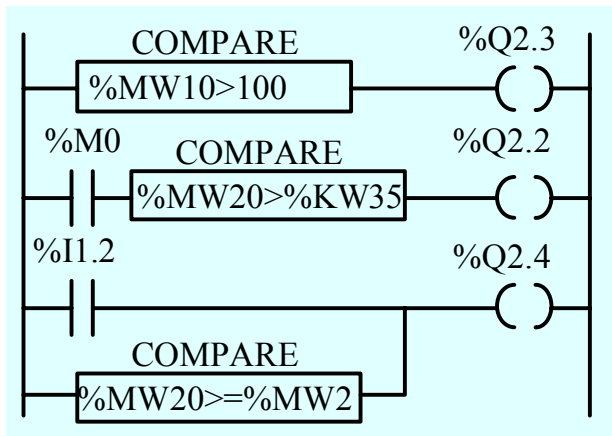
Retard au déclenchement ou retardé au relâchement : mode TOF



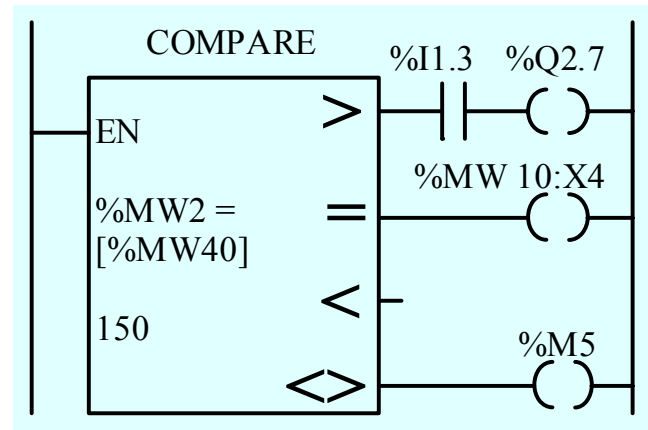
Impulsion, monostable : mode TP

VIII. Blocs comparaison :

Permet la comparaison de 2 opérands. Suivant le résultat la sortie correspondante passe à 1.



Comparaison horizontale



Comparaison verticale

```
! LD [%MW10>100]
ST %Q2.3
LD %M0
AND [%MW20>%KW35]
ST %Q2.2
LD %I1.2
OR [%MW20>=%MW2]
ST %Q2.4
```

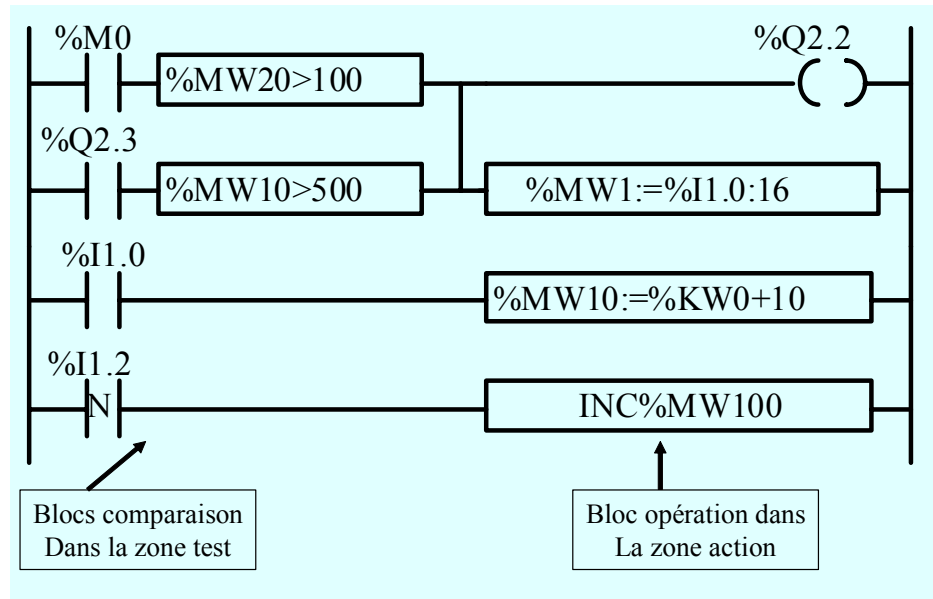
```
! IF %I1.3 THEN
    %Q2.7:=%MW2[%MW4]>150;
END_IF;
%MW10:X4:=%MW2[%MW4]=150;
%M5:=%MW2[%MW4]<>150;
```

IX. Bloc opération :

Les blocs opération se positionnent dans la zone action, ils réalisent les opérations arithmétiques, logiques..., font appel à la syntaxe du langage littéral structuré.

L'affectation se fait vers l'opérande placée à gauche. Les opérandes peuvent être des tableaux de bits %X0:10 (bits d'étapes), des mots simples %SW10 (mot système), ou doubles mots %KD5 (mots constants).

Les instructions peuvent être arithmétiques +, -, *, /, SQRT (racine carrée) ou logiques AND, OR, XOR ...



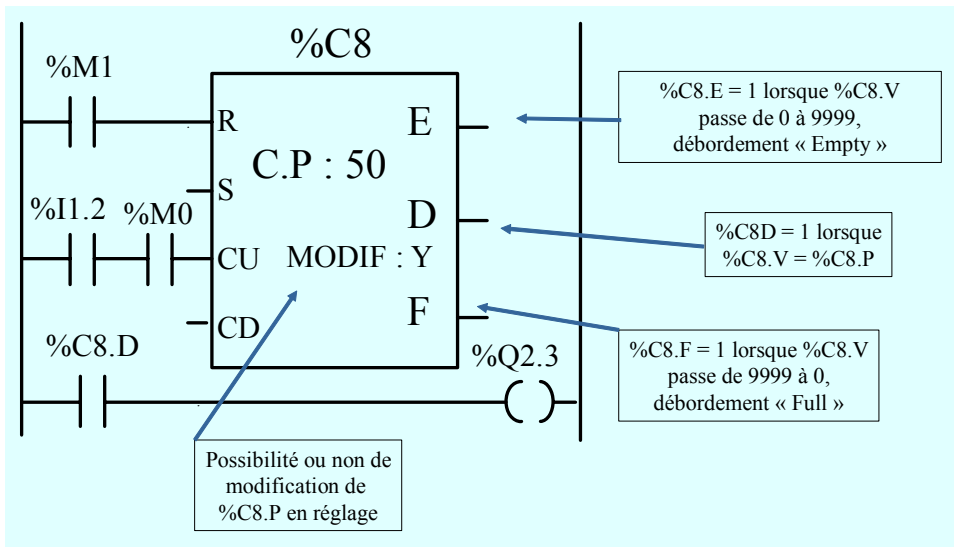
Exemples : %M10:8 := %I1.0:8 ; La valeur de chacune des 8 entrées à partir de %I1.0 jusqu'à %I1.7 est placée dans chacun des 8 bits à partir de %M10 jusqu'à %M17. Si %MW12 := %I1.0:8 ce sont les 8 bits de poids faible du mot %MW12 qui contiennent l'état des entrées.

X. Bloc

compteur-décompteur :

Ce bloc permet le comptage et le décomptage d'événements.

%Ci.V est le mot représentant la valeur courante qui est incrémentée ou décrétementée en fonction des entrées CU et CD, il peut



```
! LD %M1
R %C8
LD %I1.2
AND %M0
CU %C8
LD %C8.D
ST %Q2.3
```

```
! IF %M1 THEN
  RESET %C8;
END_IF;
IF %I1.2 AND %M0
THEN
  UP %C8;
END_IF;
%Q2.3 := %C8.D;
```

être lu et testé.

XI. Chargement, essai :

Le programme est transféré du PC vers l'API par un câble de type TSX CPU.

- Vérifier sur le bloc de visualisation de l'API le bon fonctionnement des entrées programmées, même en STOP.
- En RUN vous pouvez vérifier le bon fonctionnement des sorties à blanc, puissance coupée.
- En mode connecté on peut observer l'évolution du fonctionnement, des tables d'animation permettent de vérifier les valeurs des objets du système.

La mise au point d'un automatisme peut être très longue.

Bibliographie : Doc Schneider