

Automates Modicon Nano Terminal FTX 117

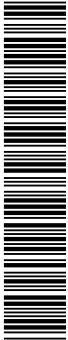
Manuel de mise en œuvre
et de programmation

Edition Novembre 2003

05



* W 9 1 3 2 9 2 5 0 0 1 0 4 A *



* T L X D M 0 7 1 1 7 F *



W913292500104A 05



TLXDM07117F

Schneider Electric Industries SAS

Headquarters

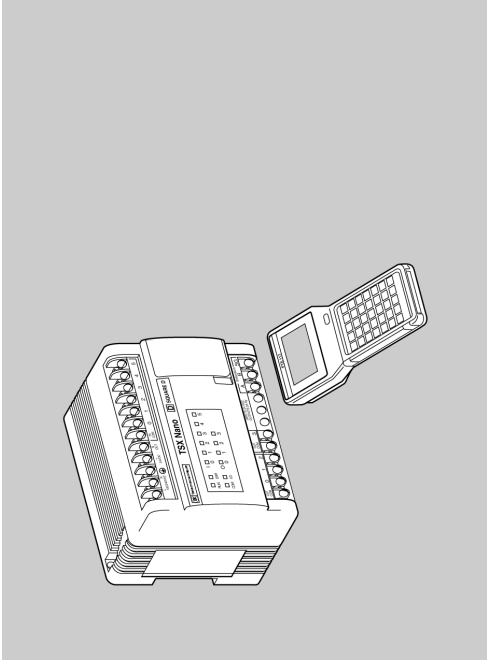
89, bd Franklin Roosevelt
F - 92506 Rueil Malmaison Cedex

<http://www.schneider-electric.com>

En raison de l'évolution des normes et du matériel,
les caractéristiques indiquées par les textes et
les images de ce document ne nous engagent
qu'après confirmation par nos services.

Printed in Italy

Novembre 2003



Telemecanique

© **Copyright Schneider Automation SAS 2003.**

Toute reproduction de cet ouvrage est interdite. Toute copie ou reproduction, même partielle, par quelque procédé que ce soit, photographique, magnétique ou autre de même que toute transcription totale ou partielle lisible sur machine électronique est interdite.

Les produits matériels et services présentés dans ce document sont à tout moment susceptibles d'évolution quant à leurs caractéristiques de présentation, de fonctionnement ou d'utilisation.

La garantie **Schneider Automation SAS** se limite à la conformité des produits de ces marques **Square D** et **Telemecanique**, ainsi qu'à leurs caractéristiques de fonctionnement annoncées. **Schneider Automation SAS** ne peut, en aucun cas, être tenue responsable ni de l'usage effectué ni des conséquences qui en découlent.

Les produits matériels présentés dans ce document sont associés pour leur mise en œuvre et leur exploitation à des logiciels **PL7®** décrits dans les manuels **TLX D•••** correspondants.

PL7 est une marque déposée de Schneider Automation SAS.
Windows est une marque déposée de Microsoft Corporation.

1 Généralités

La présente documentation s'adresse à des personnes qualifiées sur le plan technique pour mettre en oeuvre, exploiter et maintenir les produits qui y sont décrits. Elle contient les informations nécessaires et suffisantes à l'utilisation conforme des produits. Toutefois, une utilisation "avancée" de nos produits peut vous conduire à vous adresser à l'agence la plus proche pour obtenir les renseignements complémentaires.

Le contenu de la documentation n'est pas contractuel et ne peut en aucun cas étendre ou restreindre les clauses de garantie contractuelles.

2 Qualification des personnes

Seules des **personnes qualifiées** sont autorisées à mettre en oeuvre, exploiter ou maintenir les produits. L'intervention d'une personne non qualifiée ou le non-respect des consignes de sécurité contenues dans ce document ou apposées sur les équipements, peut mettre en cause la sécurité des personnes et/ou la sûreté du matériel de façon irrémédiable. Sont appelées "**personnes qualifiées**", les personnes suivantes :

- au niveau de la conception d'une application, les personnels de bureau d'études familiarisés avec les concepts de sécurité de l'automatisme (par exemple, un ingénieur d'études, ...),
- au niveau de la mise en oeuvre des équipements, les personnes familiarisées avec l'installation, le raccordement et la mise en service des équipements d'automatisme (par exemple, un monteur ou un câbleur pendant la phase d'installation, un technicien de mise en service, ...),
- au niveau de l'exploitation, les personnes initiées à l'utilisation et à la conduite des équipements d'automatisme (par exemple, un opérateur, ...),
- au niveau de la maintenance préventive ou corrective, les personnes formées et habilitées à régler ou à réparer les équipements d'automatisme (par exemple, un technicien de mise en service, un technicien de S.A.V, ...).

3 Avertissements

Les avertissements servent à prévenir les risques particuliers encourus par les personnels et/ou le matériel. De par leur importance, ils sont donc signalés dans la documentation et sur les produits par une marque d'avertissement :

Danger ou Attention

signifie que la non application de la consigne ou la non prise en compte de l'avertissement conduit ou peut conduire à des lésions corporelles graves, pouvant entraîner la mort ou/et à des dommages importants du matériel.

Avertissement ou Important ou !

indique une consigne particulière dont la non-application peut conduire à des lésions corporelles légères ou/et à des dommages matériel.

Note ou Remarque

met en exergue une information importante relative au produit, à sa manipulation ou à sa documentation d'accompagnement.

4 Conformité d'utilisation

Les produits décrits dans la présente documentation **sont conformes aux Directives Européennes (*)** auxquelles ils sont soumis (marquage CE). Toutefois, ils ne peuvent être utilisés de manière correcte, que dans les applications pour lesquelles ils sont prévus dans les différentes documentations et en liaison avec des produits tiers agréés.

En règle générale, si toutes les prescriptions de manipulation, de transport et de stockage et si toutes les consignes d'installation, d'exploitation et de maintenance sont respectées, les produits seront utilisés d'une manière correcte, sans danger pour les personnes ou les matériels.

(*) Directives DCEM et DBT concernant la Compatibilité Electromagnétique et la Basse Tension.

5 Installation et mise en oeuvre des équipements

Il est important de respecter les règles suivantes, lors de l'installation et de la mise en service des équipements. De plus, si l'installation contient des liaisons numériques, il est impératif de respecter les règles élémentaires de câblage, présentées dans le guide utilisateur, **référéncé TSX DG GND** et le manuel **TSX DR NET**, intercalaire C.

- respecter scrupuleusement les consignes de sécurité, contenues dans la documentation ou sur les équipements à installer et mettre en oeuvre.
- le type d'un équipement définit la manière dont celui-ci doit être installé :
 - un équipement encastrable (par exemple, un pupitre d'exploitation ou un contrôleur de cellules) doit être encastré,
 - un équipement incorporable (par exemple, un automate programmable) doit être placé dans une armoire ou un coffret,
 - un équipement "de table" ou portable (par exemple, un terminal de programmation) doit rester avec son boîtier fermé,
- si l'équipement est connecté à demeure, il sera nécessaire d'intégrer dans son installation électrique, un dispositif de sectionnement de l'alimentation et un coupe circuit de protection sur surintensité et de défaut d'isolement. Si ce n'est pas le cas, la prise secteur sera mise à la terre et facilement accessible. **Dans tous les cas, l'équipement doit être raccordé à la masse de protection PE par des fils vert/jaune (NFC 15 100).**
- pour permettre de détecter une tension dangereuse, les circuits BT (bien que Basse Tension) doivent être obligatoirement raccordés à la terre de protection.
- avant de mettre sous tension un équipement, il est nécessaire de vérifier que sa tension nominale est réglée en conformité avec la tension d'alimentation du réseau.
- si l'équipement est alimenté en 24 ou en 48 V continu, il y a lieu de protéger les circuits basse tension. N'utiliser que des alimentations conformes aux normes en vigueur.
- vérifier que les tensions d'alimentation restent à l'intérieur des plages de tolérance définies dans les caractéristiques techniques des équipements.
- toutes les dispositions doivent être prises pour qu'une reprise secteur (immédiate, à chaud ou à froid) n'entraîne pas d'état dangereux pour les personnes ou pour l'installation.
- les dispositifs d'arrêt d'urgence doivent rester efficaces dans tous les modes de fonctionnement de l'équipement, même anormal (par exemple, coupure d'un fil). Le réarmement de ces dispositifs ne doit pas entraîner des redémarrages non contrôlés ou indéfinis.
- les câbles véhiculant des signaux doivent être placés de telle manière que les fonctions d'automatismes ne soient pas perturbées par des influences capacitatives, inductives, électromagnétiques, ...
- les équipements d'automatisme et leurs dispositifs de commande doivent être installés de façon à être protégés contre des manœuvres inopinées.

- afin d'éviter qu'un manque de signaux n'engendre des états indéfinis dans l'équipement d'automatisme, les mesures de sécurité adéquates doivent être prises pour les entrées et les sorties.

6 Fonctionnement des équipements

La sûreté de fonctionnement d'un dispositif représente son aptitude à éviter l'apparition de défaillances et à minimiser leurs effets lorsqu'elles se sont produites.

Un système est dit de sécurité totale si l'apparition de défaillances ne conduit **jamais** à une situation dangereuse.

Un défaut interne à un système de commande sera dit de type :

- passif, s'il se traduit par un circuit de sortie ouvert (aucun ordre n'est donné aux actionneurs).
- actif, s'il se traduit par un circuit de sortie fermé (un ordre est envoyé aux actionneurs).

Du point de vue de la sécurité, un défaut d'un type donné sera dangereux ou non selon la nature de la commande effectuée en fonctionnement normal. Un défaut passif est dangereux si la commande normale est une opération d'alarme; un défaut actif est dangereux s'il maintient ou active une commande non désirée.

Il est important de noter la différence fondamentale de comportement d'un relais électromécanique et d'un composant électronique (par exemple un transistor) :

- la probabilité est grande, environ 90 cas sur 100, pour que la défaillance d'un relais conduise à un circuit ouvert (circuit de commande hors tension).
- la probabilité est de l'ordre de 50 cas sur 100, pour que la défaillance d'un transistor conduise soit à un circuit ouvert, soit à un circuit fermé.

C'est pourquoi il est important de bien mesurer la nature et la conséquence des défauts lorsque l'on aborde une automatisation à partir de produits électroniques tels que les automates programmables, y compris dans le cas d'utilisation sur ceux-ci de modules de sorties à relais.

Le concepteur du système devra se prémunir, **par des dispositifs extérieurs à l'automate programmable**, contre les défauts actifs internes à cet automate, non signalés et jugés dangereux dans l'application. Leur traitement peut faire appel à des solutions de technologies variées telles que mécanique, électromécanique, pneumatique, hydraulique (par exemple, câblage direct du détecteur de fin de course et des arrêts d'urgence sur la bobine du contacteur de commande d'un mouvement).

Pour se prémunir contre les défauts dangereux susceptibles d'intervenir au niveau des circuits de sortie et des pré-actionneurs, on pourra mettre à profit des principes généraux mettant en oeuvre la grande capacité de traitement de l'automate, comme par exemple " le contrôle par les entrées de la bonne exécution des ordres demandés par le programme".

7 Caractéristiques électriques et thermiques

Le détail des caractéristiques électriques et thermiques des équipements figure dans les documents techniques associées (manuels de mise en oeuvre, instructions de service).

8 Conditions d'environnement

Les équipements, tels les automates TSX Nano, répondent aux exigences du traitement "TC" (1). Pour des installations en atelier de production industrielle ou en ambiance correspondant au traitement "TH" (2), ces équipements doivent être incorporés dans des enveloppes de protection minimale IP54, prescrit par les normes IEC 664 et NF C 20 040.

Les automates TSX Nano, qui présentent par eux mêmes un indice de protection IP20, peuvent donc être installés sans enveloppe dans des locaux à accès réservé ne dépassant pas le degré de pollution 2 (salle de contrôle ne comportant ni machine, ni activité de production de poussières).

(1) Traitement "TC" : traitement tout climat.

(2) Traitement "TH" : traitement pour ambiances chaudes et humides.

9 Maintenance préventive ou corrective

Disponibilité

La disponibilité d'un système représente son aptitude, sous les aspects combinés de sa fiabilité, de sa maintenabilité et de sa logistique de maintenance, à être en état d'accomplir une fonction requise, à un instant donné et sur un intervalle de temps déterminé.

La disponibilité est donc propre à chaque application puisqu'elle est la combinaison de :

- l'architecture du système automatique,
- la fiabilité et la maintenabilité : caractéristiques intrinsèques des matériels (automates, capteurs, machine, etc...),
- la logistique de maintenance : caractéristique intrinsèque à l'utilisateur de l'automatisme (structure des logiciels, signalisation des défauts, process, pièces de rechange sur place, formation du personnel).

Conduite à tenir pour le dépannage

- les réparations sur un équipement d'automatisme ne doivent être effectuées que par du personnel qualifié (technicien S.A.V ou technicien agréé par Schneider Automation SA). Lors de remplacement de pièces ou de composants, n'utiliser que des pièces d'origine.
- avant d'intervenir sur un équipement (par exemple ouvrir un boîtier), couper dans tous les cas son alimentation (débrancher sa prise de courant ou ouvrir le dispositif de sectionnement de son alimentation).
- avant d'intervenir sur site sur un équipement "mécanique", couper son alimentation de puissance et verrouiller mécaniquement les pièces susceptibles de mouvements.
- avant d'effectuer une opération sur l'automate, de modifier une connexion,... vérifier dans la documentation si cette opération doit s'effectuer hors tension ou s'il est possible de l'effectuer sous tension. Suivre rigoureusement les consignes données par la documentation.
- sur des sorties à logique positive ou des entrées à logique négative, prendre toutes les précautions pour ne pas qu'un fil déconnecté vienne en contact avec la masse mécanique (risque de commande intempestive).

Chapitre	Page
1 Présentation générale	1/1
1.1 L'automate dans une structure d'automatisme	1/1
1.2 Les automates TSX Nano	1/2
1.2-1 Présentation	1/2
1.2-2 Rappel des références catalogue	1/4
1.2-3 Principales fonctionnalités des automates TSX Nano	1/5
1.3 Le cycle automate	1/8
1.3-1 Exécution normale (cyclique)	1/8
1.3-2 Exécution périodique	1/9
1.4 L'extension d'entrées/sorties	1/11
1.5 L'adressage des entrées/sorties	1/12
1.6 Les entrées/sorties spécifiques	1/13
1.7 Compléments sur les entrées/sorties	1/15
1.7-1 Les entrées à filtrage programmable	1/15
1.7-2 Les sorties statiques protégées sur TSX 07...12	1/17
1.8 Les points de réglage analogique	1/18
1.9 La visualisation de l'état automate et des entrées/sorties	1/19
1.10 Les extensions automates	1/21
1.11 Les modules analogiques	1/23
1.11-1 Présentation	1/23
1.11-2 Rappel des références catalogue	1/24

Chapitre	Page
2 Encombrements/Montage/Implantation	2/1
2.1 Encombrements	2/1
2.2 Montage	2/2
2.3 Règles d'implantation	2/3
3 Raccordements	3/1
3.1 Précautions et règles de câblage des entrées/sorties	3/1
3.1-1 Précautions et règles générales	3/1
3.1-2 Précautions particulières de raccordement des entrées à faible immunité	3/2
3.2 Raccordement des alimentations	3/4
3.3 Raccordement des entrées TOR	3/5
3.3-1 Raccordement des entrées 24 VDC	3/5
3.3-2 Raccordement des entrées 115 VAC	3/7
3.4 Raccordement des sorties TOR	3/7
3.4-1 Raccordement des sorties relais	3/7
3.4-2 Raccordement des sorties relais sur automate TSX 07 •1 1648	3/9
3.4-3 Raccordement des sorties transistors, logique négative	3/9
3.4-4 Raccordement des sorties transistors, logique positive	3/11
3.5 Raccordement d'une extension d'entrées/sorties	3/13
3.6 Raccordement des extensions automates	3/14
3.7 Raccordement des entrées analogiques	3/15
3.7-1 Raccordement avec l'entrée 0 du TSX Nano câblée en sink	3/15
3.7-2 Raccordement avec l'entrée 0 du TSX Nano câblée en source	3/15

Chapitre	Page
3.8 Raccordement des sorties analogiques	3/16
3.8-1 Raccordement avec la sortie 0 Source du TSX Nano	3/16
3.8-2 Raccordement avec la sortie 0 Sink du TSX Nano	3/16
4 Fonctions spécifiques	4/1
4.1 Entrée RUN/STOP	4/1
4.2 Sortie SECURITE	4/1
4.3 Entrées mémorisation d'état	4/2
4.4 Entrées/sorties liées au comptage rapide	4/3
4.4-1 Utilisation en compteur rapide	4/4
4.4-2 Utilisation en fréquencemètre	4/5
4.4-3 Utilisation en compteur/décompteur	4/6
4.5 Sortie PULSE : génération train d'impulsions	4/7
4.6 Sortie PWM : modulation de largeur d'impulsions	4/8
5 Caractéristiques/Conditions de service	5/1
5.1 Caractéristiques des alimentations	5/1
5.2 Caractéristiques des entrées TOR 24VDC et 115VAC	5/2
5.3 Caractéristiques des sorties TOR transistors 24VDC	5/3
5.4 Caractéristiques des sorties TOR relais	5/4
5.5 Caractéristiques des E/S analogiques	5/5

Chapitre	Page
5.6 Conditions de service	5/7
5.6-1 Normes	5/7
5.6-2 Environnement, conditions normales de service	5/7
6 Mise en service	6/1
6.1 Procédure de première mise sous tension	6/1
6.2 Vérification du raccordement des entrées/sorties	6/3
7 Compléments	7/1
7.1 Coupures et reprises secteur	7/1
7.2 Initialisation automate	7/3
7.3 Sauvegarde du programme et des données	7/3

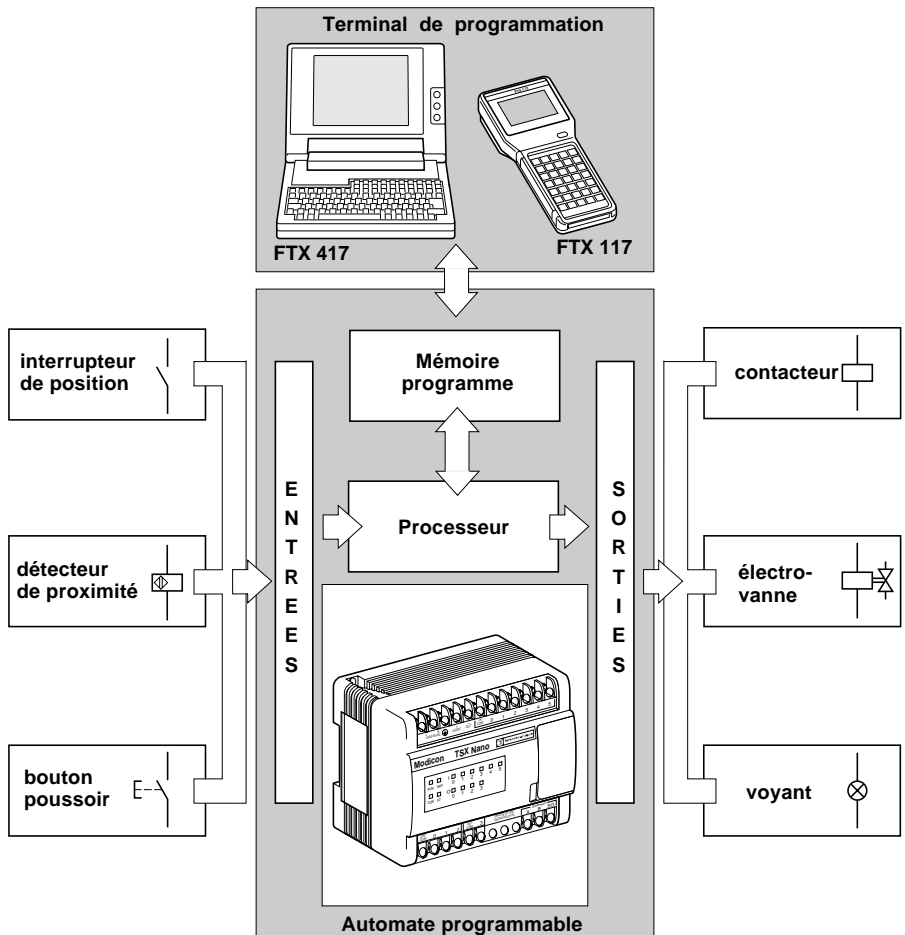
1.1 L'automate dans une structure d'automatisme

Un automate programmable se décompose en quatre sous-ensembles principaux :

- les entrées,
- les sorties,
- la mémoire où sont stockées les instructions du programme utilisateur,
- le processeur qui lit les informations d'entrées et commande les sorties en fonction des instructions du programme utilisateur.

Le terminal de programmation est l'outil utilisé pour :

- créer et transférer en mémoire le programme utilisateur,
- mettre au point le programme utilisateur et la mise en marche de l'automatisme,
- assurer le diagnostic de l'installation.

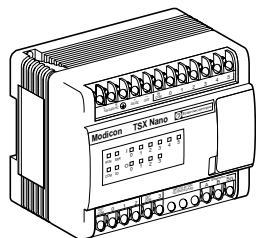


1.2 Les automates TSX Nano

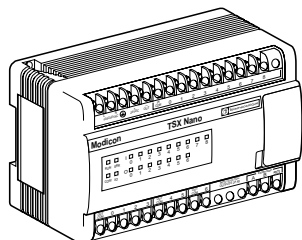
1.2-1 Présentation

Les automates TSX Nano sont disponibles en trois tailles d'entrées/sorties (10, 16 et 24 entrées/sorties). Ils sont programmables en langage PL7 réversible (langage liste d'instructions ou langage à contacts). La programmation de ces automates s'effectue à partir :

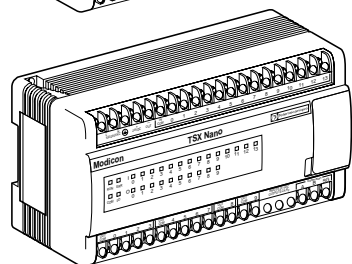
- soit du terminal FTX 117 (langage liste d'instructions),
- soit d'un terminal FTX 417 ou compatible PC (langage à contacts ou langage liste d'instructions).



10 E/S (6E + 4S)



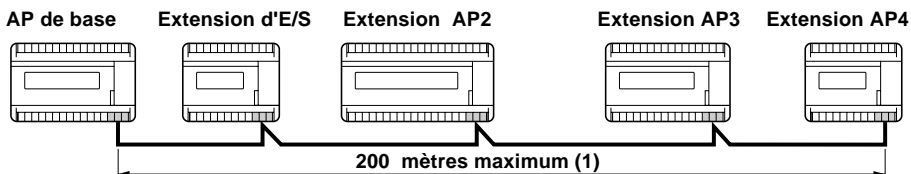
16 E/S (9E + 7S)



24 E/S (14E + 10S)

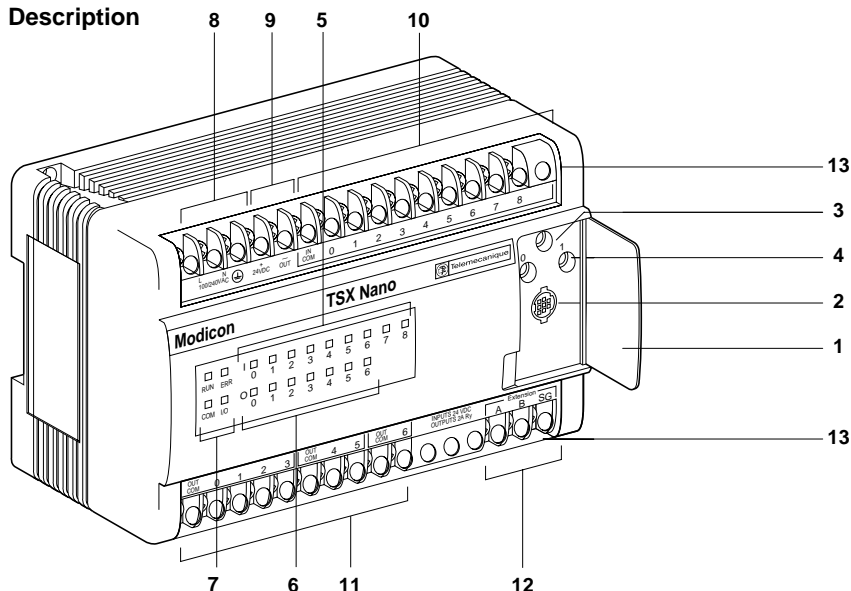
16 E/S (9E + 7S) automate à entrées 115VAC

Chaque automate de base peut être étendu par une **Extension d'E/S**, constituée par l'un des automates TSX Nano. De plus un maximum de trois **Extensions automatiques** avec communication par mots d'échange peuvent être associées à l'automate de base. Seul l'automate de base peut recevoir une extension d'entrées/sorties.



(1) voir chapitre 3.5 - intercalaire A pour le type de câble à utiliser

Description



- 1 Volet d'accès aux éléments 2, 3, et 4.
 - 2 Prise pour raccordement d'un équipement ASCII ou UNI-TELWAY. Protocoles ASCII, UNI-TELWAY Maître ou esclave RS 485.
 - 3 Sélecteur pour codage de la fonction de l'automate :
 - 0 = automate de base, 1 = Extension d'entrées/sorties.
 - 5 = Extension automate n°2, 6 = Extension automate n°3, 7 = Extension automate n°4.
 - 4 Point(s) de réglage analogique :
 - 1 sur automates 10 E/S,
 - 2 sur automates 16 ou 24 E/S,
 - 5 Visualisation de l'état des entrées (1),
 - 6 Visualisation de l'état des sorties (1),
 - 7 Visualisation de l'état automate : RUN, ERR, COM, I/O,
 - 8 Raccordement de l'alimentation secteur,
 - 9 Alimentation capteurs sur modèles alimentés en $\leq 100/240$ V : ≤ 24 V/150 mA. Cette alimentation n'existe pas sur le modèle à entrées ≤ 115 V.
 - 10 Raccordement des entrées.
 - 11 Raccordement des sorties.
 - 12 Raccordement Extension (Extension E/S et/ou Extension automate) ou Modbus esclave.
 - 13 Cache amovible de protection des borniers.
- (1) avec possibilité de visualisation de l'état de 16 bits internes maximum.

1.2-2 Rappel des références catalogue

Nb. E/S	Type alimentation		Type entrées isolées		Type sorties			Références
	110/240V c ☉	24V a æ	24V a	115V c ☉	transistors logique négative	24Va logique positive	relais 24 V a 24/240Vc	
10	●		●				●	TSX 07 20 1028 TSX 07 30 1028
	●		●		●			TSX 07 20 1008 TSX 07 30 1008
6E +		●	●		●			TSX 07 20 1002 TSX 07 30 1002
4S		●	●				●	TSX 07 20 1022 TSX 07 30 1022
		●	●			●		TSX 07 20 1012 TSX 07 30 1012
16	●		●				●	TSX 07 21 1628 TSX 07 31 1628
	●			●			●	TSX 07 21 1648 TSX 07 31 1648
9E +	●		●		●			TSX 07 21 1608 TSX 07 31 1608
7S		●	●		●			TSX 07 21 1602 TSX 07 31 1602
		●	●				●	TSX 07 21 1622 TSX 07 31 1622
		●	●			●		TSX 07 21 1612 TSX 07 31 1612
24	●		●				●	TSX 07 21 2428 TSX 07 31 2428
	●		●		●			TSX 07 21 2408 TSX 07 31 2408
14E +		●	●		●			TSX 07 21 2402 TSX 07 31 2402
10S		●	●				●	TSX 07 21 2422 TSX 07 31 2422
		●	●			●		TSX 07 21 2412 TSX 07 31 2412

Note:

Les modèles TSX 07 3●.... offrent par rapport aux modèles TSX 07 2●.... des fonctions supplémentaires telles que Uni-Telway Maître/Esclave sur prise terminal, automate Modbus esclave sur port d'extension, raccordement de modules d'entrée et sortie analogique, ...

Les modèles TSX 07 3●..02/08/48 (références grisées) seront commercialisées courant 1997.

1.2-3 Principales fonctionnalités des automates TSX Nano

Toutes les entrées/sorties sont par défaut configurées en E/S TOR. Cependant certaines entrées/sorties peuvent être affectées par configuration à des tâches spécifiques (entrée RUN/STOP, entrées mémorisation d'état, entrées/sorties comptage rapide 10 kHz ou comptage/décomptage 1 kHz, sortie SECURITE, sortie train d'impulsions "Pulse", sortie modulation de largeur d'impulsion "PWM").

L'automate TSX Nano est programmable en langage PL7 réversible (langage liste d'instructions ou langage à contacts) permettant l'utilisation de fonctions horodateurs, temporisateurs, compteurs/décompteurs, registres mots, registre bits à décalage, programmeurs cyclique, pas à pas.

Scrutation	Normale (cyclique) ou périodique (2 à 150 ms)	
Temps de cycle	inférieur à 1 ms pour 1000 instructions. (3)	
Temps d'exécution	0,2 µs (minimum) pour une instruction élémentaire booléenne	
Capacité mémoire	Données	256 mots internes, 64 mots constants, 128 bits internes (dont 64 sauvegardés sur coupure secteur)
	Programme	≤ 1000 instructions (RAM sauvegardée et EEPROM)
Sauvegarde	RAM automate : par batterie. Durée de sauvegarde : 30 jours	
Langage	PL7 réversible : langage liste d'instructions ou langage à contacts	
Extension d'E/S	1 (chaque automate peut être utilisé comme extension d'E/S). Distance maximale automate de base ↔ extension : 200 mètres(1)	
Extensions automates	3, associées à l'automate de base avec communication par mots d'échanges. Dans ce cas, seul l'automate de base peut recevoir une extension d'entrées/sorties. Distance maximale automate de base ↔ dernière extension automate : 200 mètres(1)	
Blocs fonctions	Horodateurs sauvegardés(2) 16	
	Temporisateurs	32 base de temps: 1 ms (pour les 2 premiers), 10ms, 100ms, 1s, 1min avec présélection de 0 à 9999.
	Compteurs/décompteurs	16 présélection de 0 à 9999
	Registres mots	4 type FIFO ou LIFO
	Registres bits à décalage	8 16 bits
	Programmateurs cyclique	4 8 pas, 16 bits d'ordre
	Pas à pas	4 256 pas

(1) voir chapitre 3.5 - intercalaire A pour le type de câble à utiliser

(2) disponibles sur tous les automates 16 et 24 E/S : TSX 07 ● 16/24

(3) par défaut, un automate maître (switch position 0) scrute toutes les extensions à 19 200 bds. Pour des raisons de performance, il est conseillé de déconfigurer les extensions non utilisées (voir chapitre A-1.10).

Points de réglage analogique	1 (sur TSX Nano, 10 E/S), 2 (sur TSX Nano, 16 E/S et 24 E/S)
Visualisation de bits internes	Sur un automate de base ou une extension automate, possibilité de visualiser en face avant les états de 8 bits internes (TSX Nano 10 et 16 E/S) ou 16 bits internes (TSX Nano 24 E/S), voir ch 1.9.
Entrées à filtrage programmable (1)	Sur un automate de base ou une extension automate, possibilité de configurer le temps de filtrage des entrées: Pas de filtrage, 3 ms ou 12 ms (voir chapitre A-1.7). La configuration s'effectue par groupe de voies.
Entrées/sorties spécifiques (2)	<p>Par configuration, possibilité d'affecter des fonctions spécifiques à certaines entrées/sorties.</p> <hr/> <p>Entrées RUN/STOP : 1 parmi les 6 premières de l'automate de base ou de l'extension automate (%I0.0 à %I0.5)</p> <hr/> <p>Mémorisation d'état : les 6 premières entrées de l'automate de base ou de l'extension automate (%I0.0 à %I0.5).</p> <hr/> <p>Module d'entrée analogique raccordé sur %I0.0 en fonction fréquencemètre,</p> <hr/> <p>Comptage rapide : 10 kHz, Fréquencemètre : 10 kHz Comptage/décomptage rapide : 1 kHz</p> <hr/> <p>Sorties SECURITE : 1 parmi les 4 premières sorties de l'automate de base ou de l'extension automate (%Q0.0 à %Q0.3)</p> <hr/> <p>PULSE : train d'impulsions (4,9 kHz maximum)</p> <hr/> <p>PWM : modulation de largeur d'impulsions (4,9 kHz maximum)</p> <hr/> <p>Module de sortie analogique raccordé sur %Q0.0 en fonction PWM (modulation largeur d'impulsion),</p> <hr/> <p>Réflexes : 2 (%Q0.1 et %Q0.2), associées au comptage rapide, elles permettent la prise en compte d'informations sans attendre la mise à jour en fin de cycle</p>

Note :

Les sorties PULSE et PWM peuvent être utilisées sur les automates à sorties relais dans la limite de la fréquence de battement des relais (environ 50Hz). Dans ce cas, il y a risque d'atteindre rapidement le nombre de manœuvres autorisées.

Il est donc conseillé d'utiliser principalement ces sorties sur les automates à sorties transistors.

(1) non disponibles sur automate à entrées 115 VAC : TSX 07 •1 1648

(2) Sur l'automate à entrées 115 VAC : TSX 07 •1 1648, les entrées/sorties spécifiques sont limitées à l'entrée RUN/STOP et à la sortie SECURITE.

Synthèse des principales fonctionnalités en fonction du type d'automate

Fonctionnalités	Automates 10 E/S TSX 07 30/20 10..	Automates 16 E/S TSX 07 31/21 1648 31/21 16..		Automates 24 E/S TSX 07 31/21 24..
Entrées RUN/STOP	●	●	●	●
Mémorisation état	●		●	●
Comptage 10 kHz	●		●	●
Analogique/ Fréquencemètre				
Comptage/ Décomptage 1kHz	●		●	●
Sorties SECURITE	●	●	●	●
PULSE	●	●	●	●
Analogique/ PWM	●	●	●	●
Réflexes	●		●	●
Entrées à filtrage programmable	●		●	●
Horodateur		●	●	●
Points de réglage analogique	●	●	●	●
Visualisation de bits internes sur voyants E/S	●	●	●	●
Communication				
prise UNI-TELWAY	●	●	●	●
terminal ASCII	●	●	●	●
port Ext. E/S ou automate	●	●	●	●
extension Modbus esclave	●	●	●	●
Blocs Temporisateurs	●	●	●	●
f o n c - Compteurs/ tions décompteurs	●	●	●	●
Registres mots	●	●	●	●
Registres bits à décalage	●	●	●	●
Programmateurs cycliques	●	●	●	●
Pas à pas	●	●	●	●
Instruc- Grafcet	●	●	●	●
tions (1) Relais Maître	●	●	●	●
Varia- Chaînes de bits	●	●	●	●
bles (1) Tableaux de mots	●	●	●	●
Indexation	●	●	●	●

(1) toutes les autres instructions et variables sont accessibles par les différents types d'automates.

1.3 Le cycle automate

1.3-1 Exécution normale (cyclique)

Par défaut, le cycle automate s'exécute de manière cyclique comme suit :

Traitement interne :

Le système assure implicitement :

- surveillance de l'automate :
 - contrôle de l'exécutabilité de la mémoire programme,
 - gestion du temps (mise à jour des valeurs courantes de l'horodateur),
 - mise à jour des voyants : RUN, I/O, ERR, COM,
 - détection des passages RUN/STOP,
 - surveillance d'autres paramètres système,
- traitement des requêtes en provenance de la prise terminal et du port d'extension.

Acquisition des entrées :

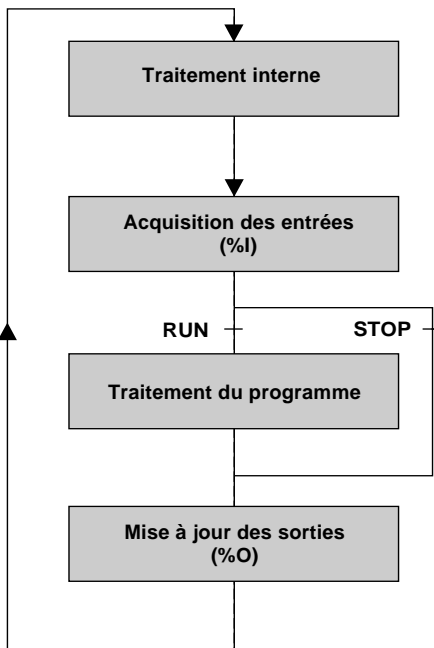
- écriture en mémoire de l'état des informations présentes sur les entrées (%I),

Traitement du programme :

- le programme écrit par l'utilisateur est alors exécuté.

Mise à jour des sorties :

- écriture des bits de sorties (%Q), selon l'état défini par le programme.



Le cycle de fonctionnement

- automate en RUN

Le processeur effectue le traitement interne, l'acquisition des entrées, le traitement du programme et la mise à jour des sorties,

- automate en STOP

Dans ce cas, le processeur effectue seulement le traitement interne, l'acquisition des entrées, et mise à jour des sorties,

Attention

Débordement du temps d'exécution

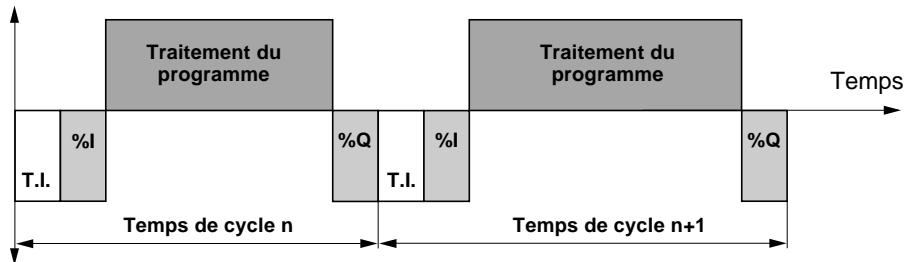
La durée d'exécution du programme utilisateur est contrôlée par l'automate (chien de garde logiciel) et ne doit pas dépasser 150 ms. Dans le cas contraire, un défaut apparaît provoquant l'arrêt immédiat de l'automate (voyants RUN et ERR clignotants).

Les cas de fonctionnement possibles :**1 Temps de cycle \leq temps du chien de garde**

Fonctionnement normal, dès la fin du cycle, le cycle suivant est lancé

2 Temps de cycle \geq temps du chien de garde

L'automate passe en STOP, les voyants RUN et ERR clignotent et le bit système %S11=1.

Diagramme des temps d'une exécution cyclique

Légende:

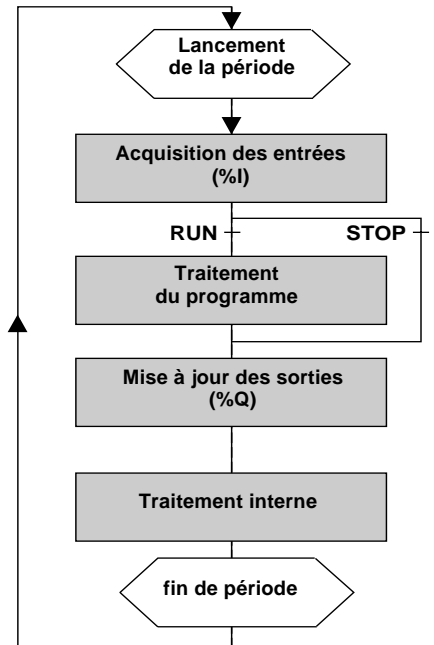
T.I = traitement interne %I = acquisition des entrées %Q = mise à jour des sorties

1.3-2 Exécution périodique

Dans ce cas, l'acquisition des entrées, le traitement du programme et la mise à jour des sorties) est effectuée de façon périodique selon un temps défini par l'utilisateur lors de la configuration (2 à 150 ms).

En début de cycle automate, un temporisateur logiciel est armé à la valeur définie en configuration. Le cycle automate doit être terminé avant l'expiration de ce temporisateur. A l'échéance du temporisateur, on relance le cycle suivant.

Si le temps de cycle dépasse le temps programmé, le bit système %S19 est mis à 1. Son test et sa remise à 0 sont à la charge du programme utilisateur.



Attention

Débordement du temps d'exécution

La durée d'exécution du programme utilisateur est contrôlée par l'automate (chien de garde logiciel) et ne doit pas dépasser 150 ms. Dans le cas contraire, un défaut apparaît provoquant l'arrêt immédiat de l'automate (voyants RUN et ERR clignotants).

Les cas de fonctionnement possibles :

1 Temps de cycle \leq période

Fonctionnement normal, le cycle suivant est lancé dès que la fin de la période programmée est atteinte.

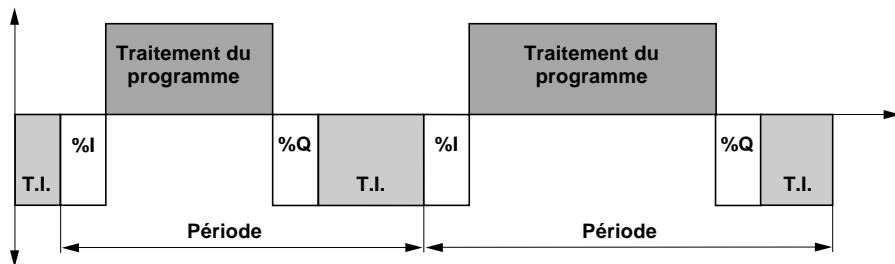
2 Période \leq temps de cycle \leq temps du chien de garde

Le bit système %S19 est mis à l'état 1 par le système et sa mise à l'état 0 est à la charge du programme utilisateur. L'automate reste en RUN.

3 Temps de cycle \geq temps du chien de garde

L'automate passe en STOP, les voyants RUN et ERR clignotent et le bit système %S11=1.

Diagramme des temps d'une exécution périodique

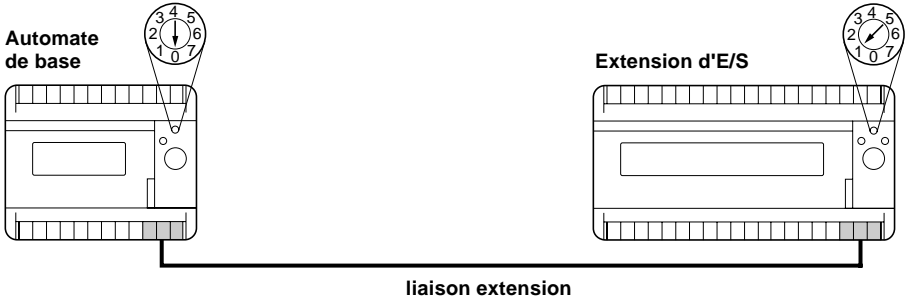


Légende :

T.I. = traitement interne **%I** = acquisition des entrées **%Q** = mise à jour des sorties

1.4 L'extension d'entrées/sorties

Chaque automate de base TSX Nano peut être étendu par une extension d'entrées/sorties. Cette extension est constituée par l'un quelconque des automates de la famille TSX Nano (10, 16 ou 24 E/S), alors utilisé uniquement en extension d'entrées/sorties.



La fonction de chaque automate est définie par positionnement du sélecteur comme indiqué sur la figure ci-dessus :

- sélecteur sur position 0 = automate de base,
- sélecteur sur position 1 = extension d'entrées/sorties.

La liaison extension entre l'automate de base et l'extension d'entrées/sorties s'effectue par un câble paire torsadée/blindée.

La distance maximale entre l'automate de base et l'extension d'entrées/sorties est limitée à 200 mètres.

Référence du câble : TSX CA0 003 de longueur 30 cm.

Pour des distances supérieures, l'utilisateur devra utiliser un câble dont les caractéristiques sont définies chapitre 3.5 - intercalaire A.

Conseil de répartition entre les entrées/sorties de l'automate de base et les entrées/sorties de l'extension d'E/S :

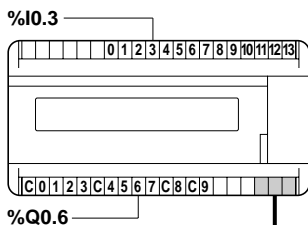
Il est conseillé d'affecter à l'automate de base les entrées/sorties "critiques" (temps de réponse faible ou sécuritives) pour le fonctionnement optimum de l'automatisme.

1.5 L'adressage des entrées/sorties

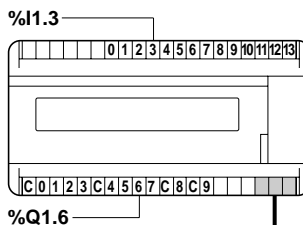
L'adressage d'une entrée/sortie est défini par les caractères suivants :

%	I ou Q	0 ou 1	.	i
symbole	I = entrée Q = sortie	0 = automate de base ou extension automate 1 = extension d'entrées/sorties	point	i = numéro de voie (voir tableau ci-dessous)

Automate de base



Extension d'E/S



Exemples :

%I0.3 signifie : Entrée, voie n°3 de l'automate de base

%Q1.6 signifie : Sortie, voie n°6 de l'automate utilisé en extension d'entrées/sorties

Numéros de voies pour les différents types d'automates

Type d'automate	Nombre d'E/S	Entrées	Sorties
TSX 07 .. 10..	10 (6E + 4S)	i = 0 à 5	i = 0 à 3
TSX 07 .. 16..	16 (9E + 7S)	i = 0 à 8	i = 0 à 6
TSX 07 .. 24..	24 (14E + 10S)	i = 0 à 13	i = 0 à 9

1.6 Les entrées/sorties spécifiques

Présentation

L'ensemble des entrées/sorties est configuré par défaut en E/S TOR. Cependant certaines entrées/sorties d'un automate de base ou d'une extension automate peuvent par configuration, être affectées à des fonctions spécifiques. Une entrée/sortie déjà utilisée pour une fonction, ne peut pas l'être pour une autre.

Entrées spécifiques

Fonctions des entrées		Entrées automate de base ou extension automate						
		%I0.0	%I0.1	%I0.2	%I0.3	%I0.4	%I0.5	%I0.6 à %I0.13
RUN/STOP		●	●	●	●	●	●	-
Mémoire d'état		●	●	●	●	●	●	-
Entrées liées au comptage/ décomptage rapide	Comptage	●	-	-	-	-	-	-
	Présélection comptage	-	●	-	-	-	-	-
	Validation et arrêt comptage/ décomptage	-	-	●	-	-	-	-
	Décomptage	-	-	-	●	-	-	-
Capture		-	-	-	-	●	-	-
Entrée analogique		●	-	-	-	-	-	-

Sorties spécifiques

Fonctions des sorties (1)		Sorties automate de base ou extension automate					
		%Q0.0	%Q0.1	%Q0.2	%Q0.3	%Q0.4	%Q0.5 à %Q0.10
SECURITE		●	●	●	●	-	-
PULSE : train d'impulsions		●	-	-	-	-	-
PWM: modulation de largeur d'impulsion		●	-	-	-	-	-
Sortie analogique		●	-	-	-	-	-
Sorties liées au comptage /dé-comptage rapide	Réflexe 0	-	●	-	-	-	-
	Réflexe 1	-	-	●	-	-	-

(1) Les sorties %Q0.0, %Q0.1, %Q0.2 et %Q0.3 configurées en sorties spécifiques ne doivent pas être utilisées à d'autres fins dans le programme application (par exemple avec les instructions booléennes (ST, STN, S, R) avec les bits d'ordres des blocs programmeur cyclique %DRi) ou comme sorties affectées aux blocs horodateur RTC.

Définition

L'utilisation de ces entrées/sorties est définie au chapitre 4 du présent intercalaire ; la description présentée ci-dessous se limite à la définition de la fonction de chaque entrée/sortie.

- **Entrée RUN/STOP** (voir chapitre 4.1)

Elle permet par l'intermédiaire d'un commutateur externe :

- de lancer l'exécution du programme (RUN),
- d'arrêter l'exécution du programme (STOP).

- **Entrée mémorisation d'état** (voir chapitre 4.3)

Elle permet la mémorisation d'une impulsion de durée inférieure à un cycle afin que celle-ci soit prise en compte au cycle suivant.

- **Entrées liées au comptage rapide** (voir chapitre 4.4)

Entrée comptage

Elle permet la prise en compte d'impulsions de comptage à une fréquence maximale de :

- 5 ou 10 kHz si elle est configurée en compteur rapide ou fréquencesmètre,
- 1 kHz si elle est configurée en compteur/décompteur rapide.

Une des applications est la gestion d'un module d'entrée analogique (voir chap. 1.11).

Entrée présélection comptage

Elle permet :

- soit la présélection à la valeur 0 dans le cas d'une utilisation en compteur rapide,
- soit la présélection à une valeur définie en configuration dans le cas d'utilisation en compteur/décompteur.

Entrée validation et arrêt comptage/décomptage

Elle permet de valider la prise en compte des signaux reçus sur les entrées comptage et décomptage.

Entrée décomptage

Elle permet la prise en compte d'impulsions de décomptage à une fréquence maximale de 1kHz.

Entrée capture

Elle permet d'effectuer une lecture à la volée de la valeur courante

- **Sortie SECURITE** (voir chapitre 4.2)

Normalement à l'état 1, elle passe à l'état 0 sur défaut de l'automate et peut ainsi être utilisée dans les circuits externes de sécurité.

- **Sortie PULSE** (voir chapitre 4.5)

Elle permet la génération d'un signal de période variable mais de rapport cyclique constant égal à 50% de la période.

- **Sortie PWM** (voir chapitre 4.6)

Elle permet la génération d'un signal, de période constante avec possibilité de faire varier le rapport cyclique.

Une des applications est la gestion d'un module de sortie analogique (voir chap. 1.11).

- **Sorties réflexes** (voir chapitre 4.4)

Liées au comptage rapide, elles permettent la prise en compte d'informations sans attendre la mise à jour en fin de cycle.

1.7 Compléments sur les entrées/sorties

1.7-1 Les entrées à filtrage programmable

L'immunité des entrées d'un automate de base ou d'une extension automate peut être modifiée par configuration à l'aide du terminal en agissant sur le temps de filtrage. Les valeurs accessibles en configuration sont les suivantes :

- 12 ms : immunité aux rebonds et signaux parasites,
- 3 ms : immunité aux rebonds et signaux parasites,
- Pas de filtrage : acquisition de signaux courts pour application rapide mais sensibilité aux rebonds et signaux parasites. Dans ce cas, l'utilisation de contacts secs est déconseillée.

Important

Pour chaque type de valeur de filtrage configurable on peut considérer 3 zones délimitées par 2 valeurs: la valeur d'immunité et la valeur de prise en compte. Tout signal de durée inférieure ou égale à l'immunité sera rejeté. Tout signal de durée supérieure ou égale à la valeur de prise en compte sera pris en compte. Tout signal de durée comprise entre ces 2 valeurs pourra soit être rejeté soit pris en compte. Dans le cas où il n'y a pas de filtrage configuré sur une entrée, les valeurs d'immunité et de prise en compte seront fixes pour les entrées %I0.8 à %I0.13 mais dépendront pour les entrées %I0.0 à %I0.7 de la configuration ou non de la fonction de comptage ou de fréquencemètre sur l'entrée %I0.0.

Filtrage configuré	Immunité	Prise en compte
12 ms	10 ms	13 ms
3 ms	2 ms	4 ms
pas de filtrage et %FC non configuré	0,125 ms pour %I0.8 à %I0.13 0,025 ms pour %I0.0 à %I0.7	0,375 ms pour %I0.8 à %I0.13 0,100 ms pour %I0.0 à %I0.7
pas de filtrage et %FC configuré en comptage 5 khz ou fréquencemètre 5 khz	0,125 ms pour %I0.8 à %I0.13 0,025 ms pour %I0.0 à %I0.7	0,375 ms pour %I0.8 à %I0.13 0,100 ms pour %I0.0 à %I0.7
pas de filtrage et %FC configuré en comptage 10 khz ou fréquencemètre 10 khz	0,125 ms pour %I0.8 à %I0.13 0,007 ms pour %I0.0 à %I0.7	0,375 ms pour %I0.8 à %I0.13 0,037 ms pour %I0.0 à %I0.7

Important

Pour le cas où il n'y a pas de filtrage configuré, les valeurs de prise en compte étant inférieures au temps de cycle automate (et donc à la période de scrutation des entrées), pour s'assurer qu'un signal de durée supérieure au temps de prise en compte soit traité, il est nécessaire d'utiliser l'entrée qui gère ce signal en capture d'impulsions.

• Entrées TOR

Par défaut, toutes les entrées d'un automate de base ou d'une extension entrées/sorties ou d'une extension automate sont configurées avec un filtrage de 12 ms. Ce temps peut être modifié par groupe de voies sur un automate de base ou une extension automate.

• Entrées à mémorisation d'état

Chacune des 6 premières entrées (%I0.0 à %I0.5) d'un automate de base ou d'une extension automate peut être configurée individuellement en mémorisation d'état. Ce fonctionnement est à utiliser pour mémoriser toute impulsion de durée inférieure au temps de cycle automate. Dans ce cas, la valeur d'immunité et de prise en compte dépendent aussi de la configuration de la fonction comptage rapide %FC.

Entrées %I0.0 à %I0.5 en mode capture	Immunité	Prise en compte
%FC non configuré	0,025 ms	0,100 ms
%FC configuré en comptage 5 khz ou fréquencemètre 5 khz	0,025 ms	0,100 ms
%FC configuré en comptage 10 khz ou fréquencemètre 10 khz	0,007 ms	0,037 ms

• Entrées comptage rapide

Si en configuration l'utilisation d'un compteur rapide, d'un fréquencemètre ou d'un compteur/décompteur est déclarée, les entrées suivantes sont automatiquement affectées à l'entrée des impulsions de comptage.

- %I0.0 en comptage rapide ou fréquencemètre,
- %I0.0 et %I0.3 en comptage/décomptage.

Utilisation en comptage rapide ou fréquencemètre:

Deux modes de fonctionnement sont configurables: un mode à 10 khz et un mode à 5 khz. L'entrée %I0.0 compte alors les impulsions qui doivent respecter les caractéristiques de durée minimum et d'écart entre elles minimum pour être prises en compte.

Mode	Immunité	Durée min. de l'impulsion	Ecart min. entre impulsions
5 Khz	0,025 ms	0,100 ms	0,100 ms
10 Khz	0,004 ms	0,045 ms	0,045 ms

Utilisation en comptage/décomptage :

Les entrées %I0.0 et %I0.3 comptent et décomptent les impulsions qui doivent respecter des caractéristiques de durée minimum et d'écart entre elles minimum pour être prise en compte. La fréquence maximum est de 1 khz.

%I0.0 et %I0.3	Immunité	Durée min. de l'impulsion	Ecart min. entre impulsions
	0,025 ms	0,100 ms	0,100 ms

Important

- Pendant que l'entrée de comptage %I0.0 est à l'état 1, aucune action de décomptage ne sera traitée sur l'entrée %I0.3.
- Pendant que l'entrée de décomptage %I0.3 est à l'état 1, aucune action de comptage ne sera traitée sur l'entrée %I0.0.

1.7-2 Les sorties statiques protégées sur TSX 07 ...12

Protection contre les surcharges et court-circuits

Les automates TSX 07 ...12 disposent de 4, 7 ou 10 sorties statiques protégées contre les surcharges et court-circuits.

Les sorties statiques 0,5A possèdent un dispositif de protection électronique qui permet, sur toute sortie active, de détecter une surcharge ou un court-circuit au 0V. L'apparition d'un tel défaut provoque :

- la limitation en courant (1A typique) de la sortie concernée,
- la disjonction de toutes les sorties du bloc (AP de base ou extension d'E/S),
- l'activation en fixe du voyant I/O de l'AP de base (si disjonction des sorties de l'AP de base) ou des voyants de l'AP de base et de l'extension d'E/S (si disjonction des sorties de l'extension d'E/S),
- la mise à 0 du bit système Défaut E/S %S10, et le positionnement des bits système %S118, %S119, %SW118:X0 et %SW119:X0 (voir chapitres B 5.1, B 5.2 et B 6.3).

Réarmement des sorties statiques

Lorsqu'un défaut a provoqué la disjonction des sorties de l'automate, il est nécessaire de les réarmer. La disjonction des sorties entraînant une marche dégradée du process piloté par l'automate, il est recommandé de conditionner le réarmement des sorties statiques à une opération manuelle. L'opérateur peut alors, avant d'effectuer le réarmement, prendre toutes dispositions vis-à-vis de l'automatisme et de la sécurité des personnes (par exemple demander le passage en marche manuelle).

Cette programmation conseillée est décrite au chapitre B 6.3.

Note :

Si le process piloté par l'automate le permet et sous la responsabilité de l'utilisateur, il est possible de programmer un réarmement automatique (voir chapitre B 6.3).

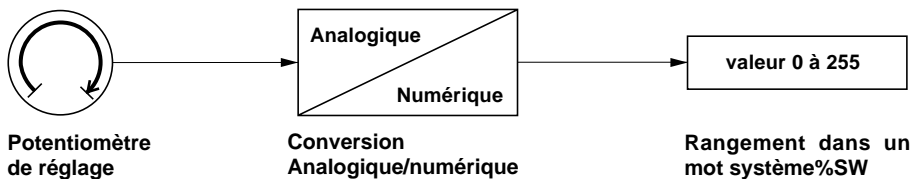
Important

Une initialisation à froid de l'automate (set de %S0) efface le défaut de disjonction et provoque une tentative de réarmement.

1.8 Les points de réglage analogique

Principe

Un convertisseur analogique/numérique convertit la tension aux bornes d'un potentiomètre en une valeur numérique (0 à 255) qui est rangée dans un mot. Cette valeur peut par exemple être utilisée comme valeur de présélection d'un temporisateur qui, exploitée par le programme utilisateur permettra le réglage à vue d'un temporisateur sans avoir recours à un terminal de programmation pour modification de la valeur de présélection. Un exemple de mise en œuvre logicielle est décrit intercalaire B - chapitre 3.2.



Les points de réglage analogique ne sont utilisables que sur les automates de base ou les extensions automates.

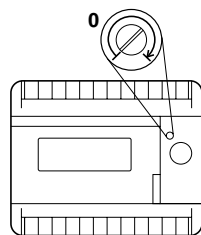
Le nombre de points de réglage analogique est variable selon le type d'automate :

- 1 sur les automates 10 E/S, repéré 0,
- 2 sur les automates 16 et 24 E/S repérés 0 et 1.

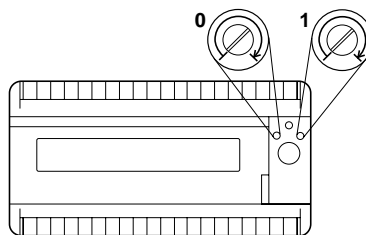
Mots de rangement selon le point de réglage analogique



Position des points de réglage analogique (potentiomètres)



Automate 10 E/S



Automate 16 ou 24 E/S

1.9 La visualisation de l'état automate et des entrées/sorties

• Visualisation de l'état automate

Le résultat des auto-tests effectués en permanence par les automates de base, les extensions automates et les extensions d'E/S, est visualisé en face avant par 4 voyants : RUN, ERR, COM et I/O.

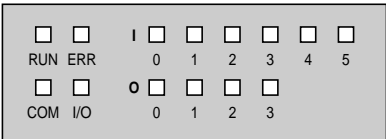


Voyants	état des voyants	Automates de base ou Extension automate	Extension d'E/S
RUN	allumé	automate RUN	idem automate de base
(vert)	clignotant	automate en STOP ou en défaut d'exécution	idem automate de base
	éteint	hors tension ou application non exécutable	non ou mal connectée
ERR	allumé	défauts internes (chien de garde, ...)	idem automate de base
(rouge)	clignotant	application non exécutable	- - - - -
	éteint	fonctionnement OK	idem automate de base
COM	allumé	échange en cours sur liaison extension (1)	idem automate de base
(jaune)	clignotant	échange en cours sur Modbus esclave (1)	- - - - -
	éteint	pas d'échange en cours sur liaison extension ou Modbus	
I/O	allumé	défaut E/S (sorties disjonctées, alim. capteurs)	idem automate de base
(rouge)	clignotant	-	fonctionnement OK
	éteint	fonctionnement OK	-

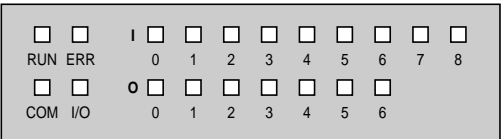
(1) Le fonctionnement extension E/S ou Modbus est exclusif

• Visualisation des entrées/sorties

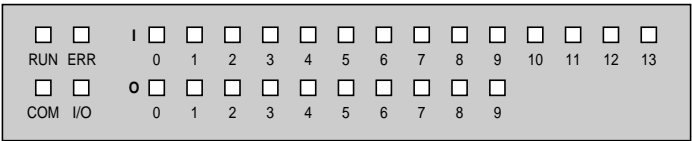
L'état de chaque entrée/sortie est visualisé en face avant de l'automate par un voyant de couleur rouge pour les automates de version inférieure à V3.0 ou de couleur verte pour les versions supérieures ou égales à V3.0.



Automate 10 E/S



Automate 16 E/S



Automate 24 E/S

Voyant allumé : entrée/sortie active,
Voyant éteint : entrée/sortie inactive

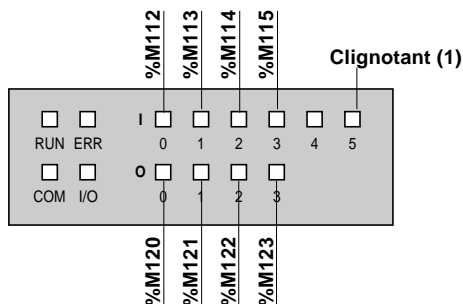
A la mise sous tension, tous les voyants sont allumés pendant une durée d'environ 1s.

• Visualisation de bits internes

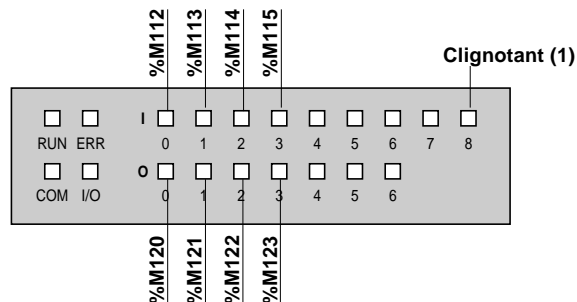
Les automates TSX Nano autorisent à la place de la visualisation de l'état des entrées/sorties, lorsque le bit système %S69 est mis à 1, la visualisation de l'état de 8 ou 16 bits internes.

La visualisation en face avant correspond à :

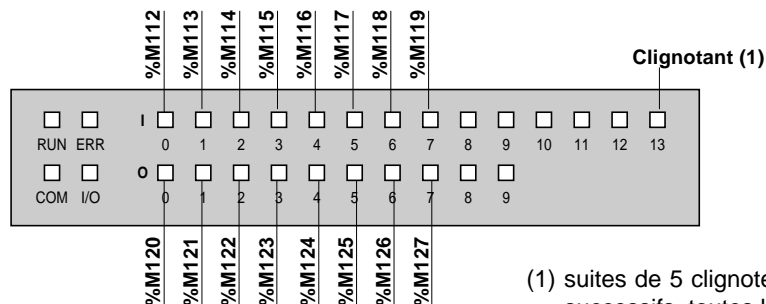
Automate 10 E/S



Automate 16 E/S



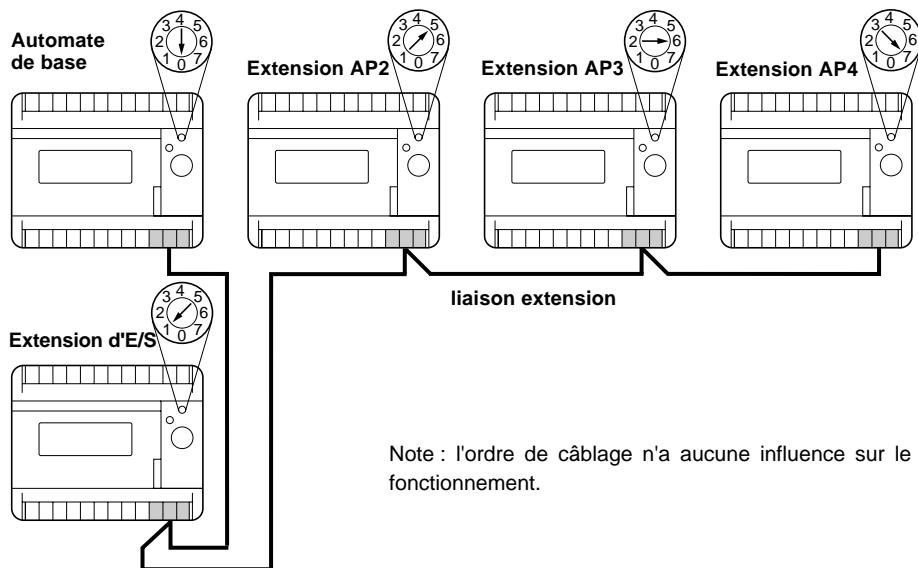
Automate 24 E/S



(1) suites de 5 clignotements brefs successifs, toutes les secondes

1.10 Les extensions automatées

Un maximum de trois extensions automatées avec communication par mots d'échanges (%IW et %QW) peuvent être associées à l'automate de base, dans ce cas seul l'automate de base peut recevoir une extension d'entrées/sorties.



La fonction de chaque automate est définie par positionnement du sélecteur comme indiqué sur la figure ci-dessus.

Fonction automate	AP de base	Ext. E/S	Extension TSX Micro (coupleur TSX STZ10)			Ext. AP2	Ext. AP3	Ext. AP4
Position du sélecteur	0	1	2	3	4	5	6	7
Adresse liaison	0	1	2	3	4	2	3	4

L'adressage des entrées/sorties des extensions automatées AP2, AP3 et AP4 est identique à celui de l'automate de base (%I0.i et %Q0.i)

La liaison extension entre l'automate de base et les extensions (E/S et/ou automates) s'effectue par un câble paire torsadée/blindée. (voir chapitre 3.5 - intercalaire A pour le type de câble à utiliser).

La distance maximale entre l'automate de base et la dernière extension automate est limitée à 200 mètres.

Important : La position du sélecteur n'est prise en compte qu'à la mise sous tension de l'automate.

Afin d'optimiser la durée des échanges entre l'automate de base et les extensions automate ou extension d'entrées/sorties, il est nécessaire de configurer les équipements à scruter et la vitesse de transmission sur la liaison.

Durée d'un cycle global d'échange (cycle complet)

Nombre d'extensions scrutées	Vitesse de transmission	
	9600 bits/s	19200 bits/s
1	17 à 19 ms	6 à 8 ms
2	34 à 35 ms	16 à 18 ms
3	53 à 55 ms	26 à 28 ms
4	72 à 74 ms	35 à 36 ms

Attention

Le bit système %S72 permet l'inhibition complète de la scrutation des extensions automate. Il est prioritaire par rapport aux choix effectués en configuration.

Important

Les bits X1, X2, X3 et X4 du mot système %SW71 permettent de tester l'état de la communication avec chacun des automates extension sur la liaison extension (état à 1: communication OK). En fonctionnement à 19200 bits/s, l'utilisation de PL7-07 en animation de données peut conduire à une dégradation des performances.

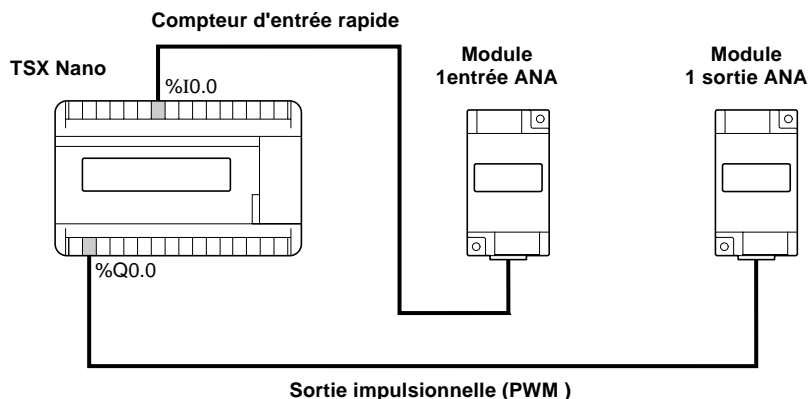
1.11 Les modules analogiques

1.11-1 Présentation

Les automates TSX 07 3..... (soit à partir de la version V3) sont capables de gérer des modules d'entrée ou de sortie analogique.

Chaque automate supporte au maximum un module d'entrée et un module de sortie. La gestion du module d'entrée analogique s'effectue à l'aide de l'entrée de comptage rapide sur l'automate.

La gestion du module de sortie analogique s'effectue à l'aide de la sortie PWM sur l'automate.



La configuration de l'entrée comptage rapide et de la sortie PWM est réalisée dans le menu configuration de PL7-07.

La gestion de commande des fonctions d'entrée, la valeur d'entrée analogique acquise, la commande des fonctions de sortie et la valeur de sortie analogique à générer sont pilotés par lecture et écriture des mots systèmes %SW100 à %SW103 dans l'application.

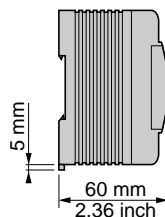
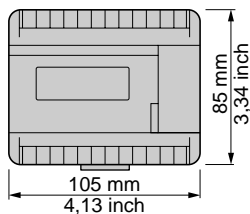
Pour le détail de gestion des modules d'entrée/sortie analogiques se référer au chapitre 4 - intercalaire B.

1.11-2 Rappel des références catalogue

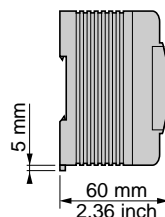
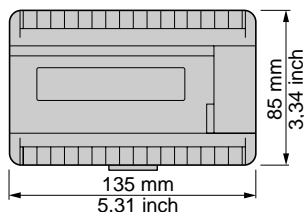
Module		Gamme			Résolution		Références
1 entrée	1 sortie	0-10V	4-20mA	+/- 10V	8 bits	10/12 bits	
●		●				●	TSX AEN 101
●			●			●	TSX AEN 102
●				●		●	TSX AEN 105
	●	●			●		TSX ASN 101
	●		●		●		TSX ASN 102
	●			●	●		TSX ASN 105

2.1 Encombrements

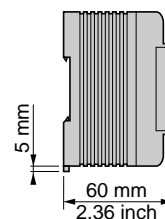
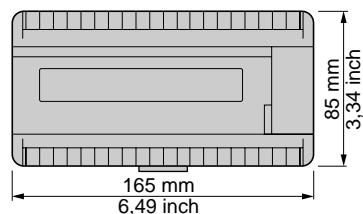
• Automates 10 E/S : TSX 07 .. 10..



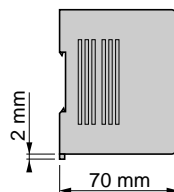
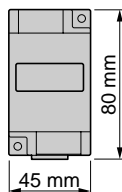
• Automates 16 E/S : TSX 07 .. 16.., sauf TSX 07 .1 1648



• Automates 24 E/S : TSX 07 .. 24.. et 16 E/S: TSX 07 .1 1648



• E/S analogiques : TSX AEN 10.. et TSX ASN 10..

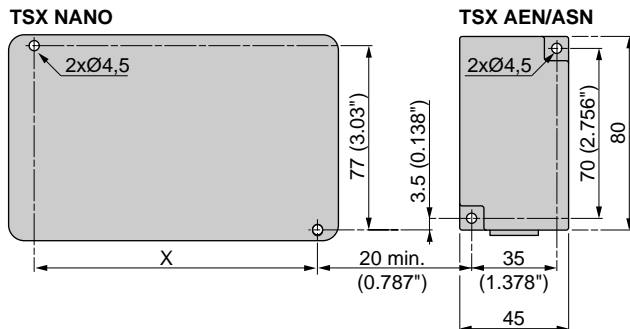


2.2 Montage

Les automates TSX Nano et modules analogiques peuvent être montés :

- sur platine ou panneau avec fixation par 2 vis Ø M3 (non fournies)
- sur profilé DIN largeur 35 mm.

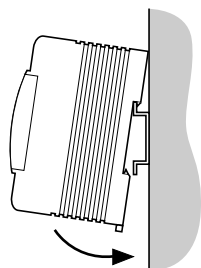
• Montage sur platine ou panneau avec fixation par vis



Automates	TSX 07 .. 10..	TSX 07 .. 16..	TSX 07 .. 1 1648	TSX 07 .. 24..
X	86 mm 3,38 inch	116 mm 4,56 inch	146 mm 5,74 inch	146 mm 5,74 inch

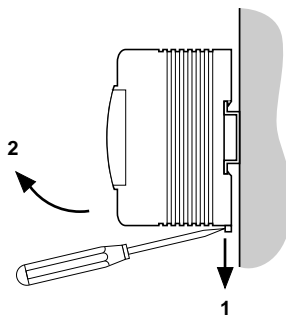
• Montage sur profilé DIN largeur 35 mm

Montage



Positionner et encliqueter l'automate sur le profilé DIN comme indiqué sur la figure ci-dessus.

Démontage



1 A l'aide d'un tournevis, dégager vers le bas le verrou assurant le blocage sur le profilé DIN.

2 Tout en maintenant le verrou débloqué, faire pivoter l'automate comme indiqué sur la figure ci-dessus.

Attention

Le montage des automates et modules analogiques sur profilé DIN nécessitent d'utiliser deux butées de blocage de type AB1-AB8P35.

2.3 Règles d'implantation

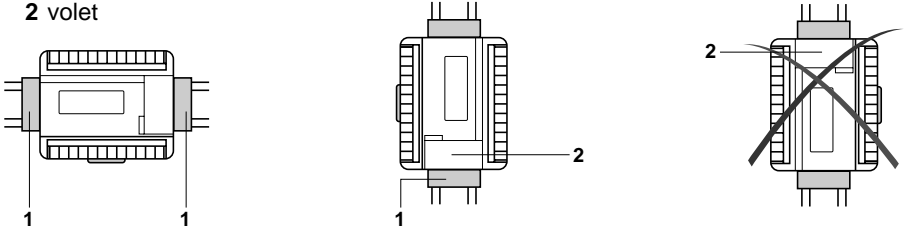
Afin de faciliter la circulation naturelle de l'air, les automates TSX Nano et modules analogiques devront être montés sur un plan vertical et à des distances minimales comme indiqué sur les figures ci-dessous.

- **Positions de montage**

- sur plan vertical

1 Butées de blocage AB1-AB8P35

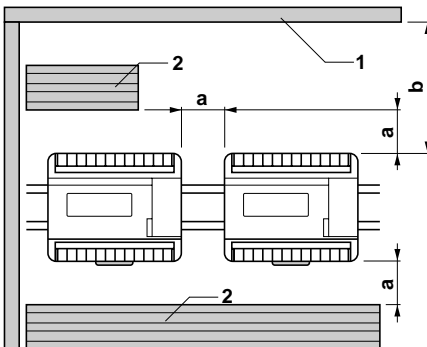
2 volet



- sur plan horizontal: montage interdit



- **Distances minimales à respecter**



1 Appareillage, enveloppe ou bati de machine.

2 Goulotte ou lyre de câblage.

$a \geq 20 \text{ mm}$

$b \geq 40 \text{ mm}$

Eviter de placer sous les automates des appareils générateurs de chaleur (transformateurs, alimentation, contacteurs de puissance,).

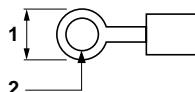
3.1 Précautions et règles de câblage des entrées/sorties

3.1-1 Précautions et règles générales

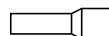
Les borniers de raccordement des automates TSX Nano sont protégés par un cache amovible assurant la protection des personnes lorsque l'automate est sous tension. Chaque borne peut recevoir au maximum 2 fils de 1 mm² équipés d'embouts, de cosses ouvertes ou de cosses fermées. **Couple de serrage des vis borniers : 0,5 Nm maxi.**



1 5,5 mm maximum

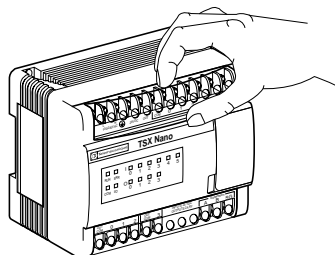


2 Ø 3,2 mm minimum



Dans le cas d'utilisation de cosses fermées, le démontage du cache est nécessaire afin d'extraire la vis pour le montage de la cosse.

Démontage du cache



Les automates TSX Nano intègrent au niveau de leurs entrées/sorties des protections assurant une bonne tenue aux ambiances industrielles. Cependant certaines règles doivent être observées afin de conserver cette immunité.

• Entrées TOR

Tout câble multi-conducteurs véhiculant les informations capteurs doit également inclure le commun de ceux-ci.

• Sorties TOR

Sorties à relais : montage obligatoire en parallèle aux bornes de la bobine du pré-actionneur:

- d'un circuit de protection de type RC ou MOV (ZNO) en courant alternatif,
- d'une diode de décharge en courant continu.

• Cheminement des câbles

- A l'extérieur de l'équipement

Tous les câbles à destination des entrées/sorties doivent être placés dans des gaines distinctes de celles renfermant les câbles véhiculant des énergies élevées, et séparées sur des parcours parallèles d'au moins 100 mm.

- A l'intérieur de l'équipement

Les câbles des circuits de puissance (alimentations, contacteurs de puissance, ...) doivent être séparés des câbles d'entrées (capteurs) et de sorties (pré-actionneurs). Les câbles d'entrées et de sorties doivent être placés si possible dans des chemins de câblage distincts.

3.1-2 Précautions particulières de raccordement des entrées à faible immunité

Certaines entrées peuvent être configurées en :

- comptage rapide ou fréquencemètre, fréquence 5 ou 10 kHz (entrée %I0.0),
- comptage/décomptage, 1 kHz (entrées : comptage %I0.0, décomptage %I0.3)
- mémorisation d'état (entrées %I0.0 à %I0.5),
- TOR sans filtrage (%I0.0 à %I0.3 et %I0.4 à %I0.7).

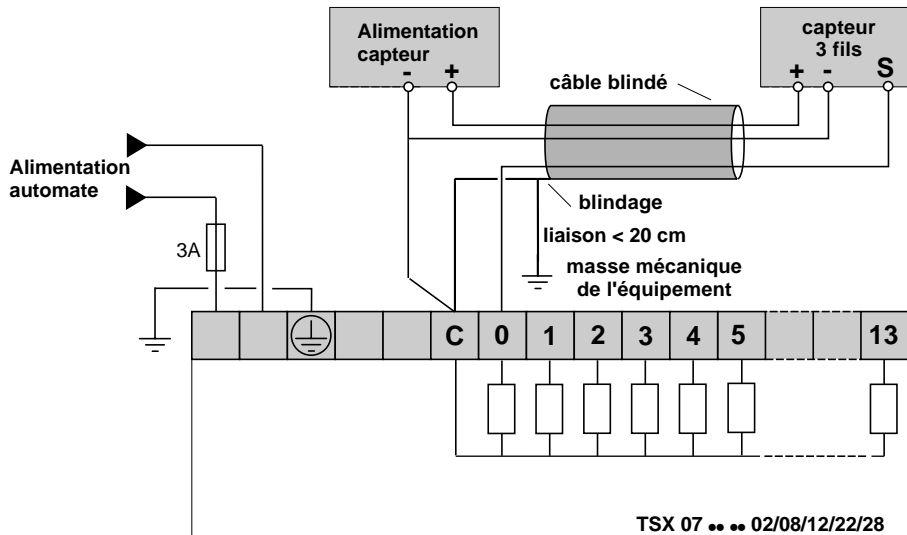
De ce fait, l'immunité des entrées ainsi configurées est réduite d'où nécessité de prendre certaines précautions de raccordement.

- **Entrée %I0.0 utilisée en comptage rapide ou fréquencemètre (5 ou 10 kHz), comptage 1 kHz, mémorisation d'état ou TOR sans filtrage :**

Utiliser obligatoirement un câble blindé avec :

- Le blindage du câble raccordé obligatoirement à la borne C (COM), commun des entrées (au - de l'alimentation avec entrées en logique positive ou au + de l'alimentation avec entrées en logique négative).
- la borne C (COM) raccordée à la masse mécanique de l'équipement.

Exemple : Schéma de principe pour le raccordement de l'entrée %I0.0 avec capteur 3 fils raccordée en logique positive

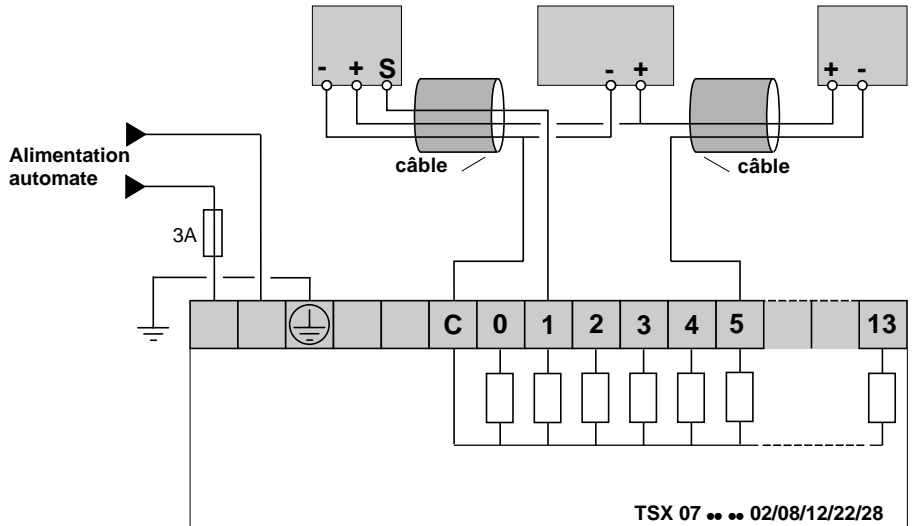


Note : dans le cas d'utilisation de la fonction comptage/décomptage 1 kHz, l'entrée %I0.3 utilisée en entrée décomptage sera raccordée selon le même principe que l'entrée %I0.0

- **Autres entrées utilisées en mémorisation d'état (%I0.1 à %I0.5) ou TOR sans filtrage (%I0.1 à I0.7) :**

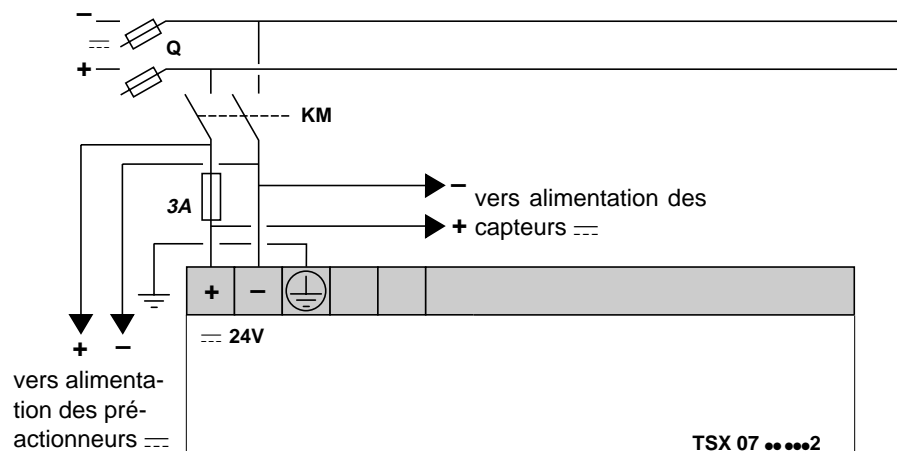
Le principe général est de regrouper dans le même câble l'ensemble des conducteurs relatifs au capteur qui commande l'entrée, d'où nécessité d'utiliser un câble par voie.

Exemple : Schéma de principe avec entrées raccordées en logique positive



3.2 Raccordement des alimentations

• Alimentation courant continu 24 VDC

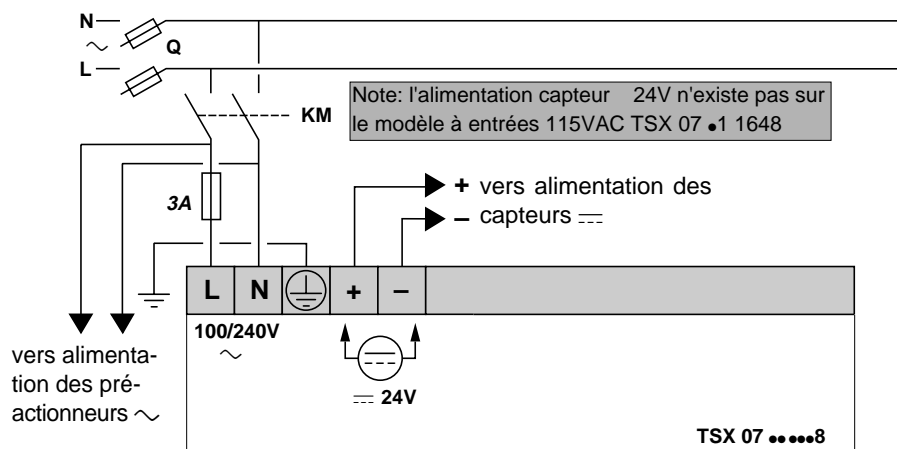


Q Sectionneur général

KM Contacteur de ligne (non indispensable sur petite installation)

Fu1 Fusible de protection de l'alimentation automate (3A)

• Alimentation courant alternatif 100 à 240 VAC



Q Sectionneur général

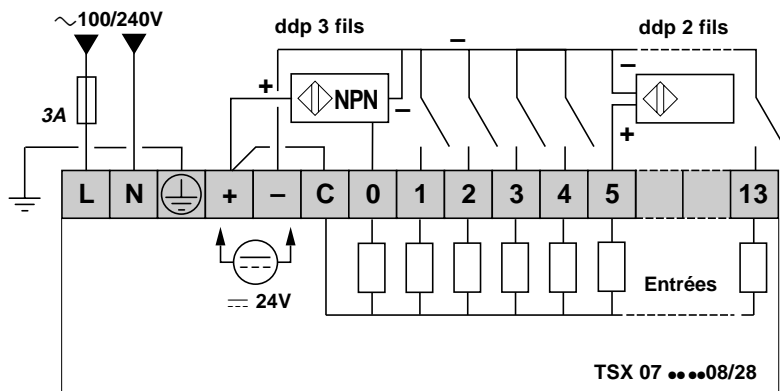
KM Contacteur de ligne ou disjoncteur (non indispensable sur petite installation)

Fu1 Fusible de protection de l'alimentation automate (3A)

• Raccordement des entrées en logique négative

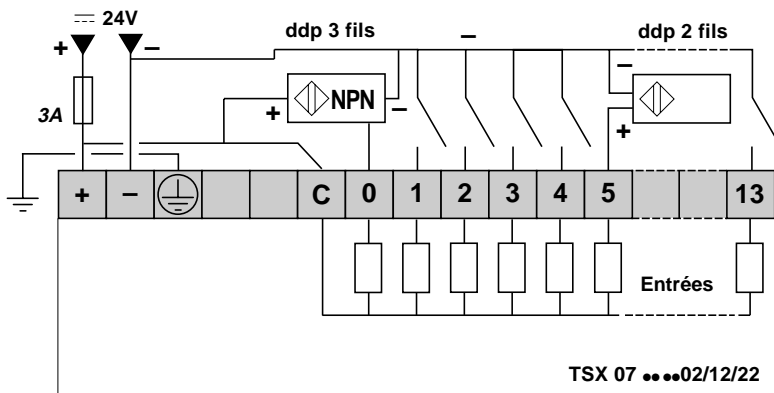
Sur automates alimentés en courant alternatif :

- 10 E/S : TSX 07 .0 1008, TSX 07 .0 1028
- 16 E/S : TSX 07 .1 1608, TSX 07 .1 1628
- 24 E/S : TSX 07 .1 2408, TSX 07 .1 2428



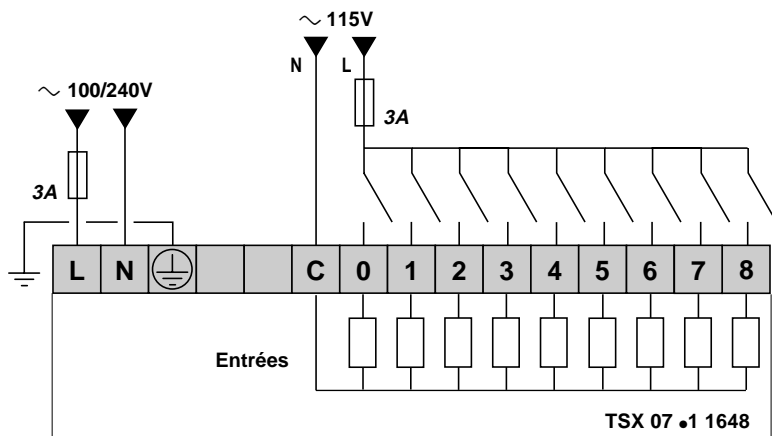
Sur automates alimentés en courant continu :

- 10 E/S : TSX 07 .0 1002, TSX 07 .0 1012, TSX 07 .0 1022
- 16 E/S : TSX 07 .1 1602, TSX 07 .1 1612, TSX 07 .1 1622
- 24 E/S : TSX 07 .1 2402, TSX 07 .1 2412, TSX 07 .1 2422



3.3-2 Raccordement des entrées 115 VAC

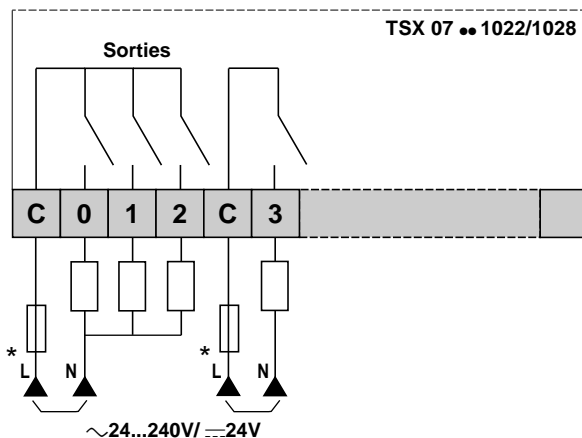
- automate : TSX 07 •1 1648



3.4 Raccordement des sorties TOR

3.4-1 Raccordement des sorties relais (sauf sur TSX 07 •1 1648)

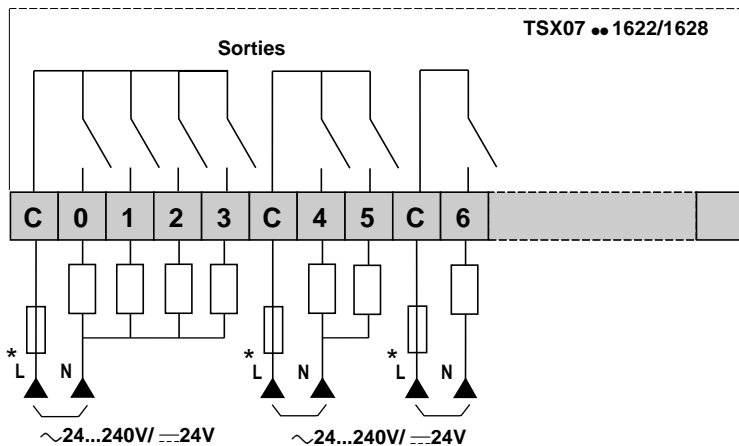
- automates 10 E/S : TSX 07 •0 1022, TSX 07 •0 1028



* fusible à calibrer selon la charge

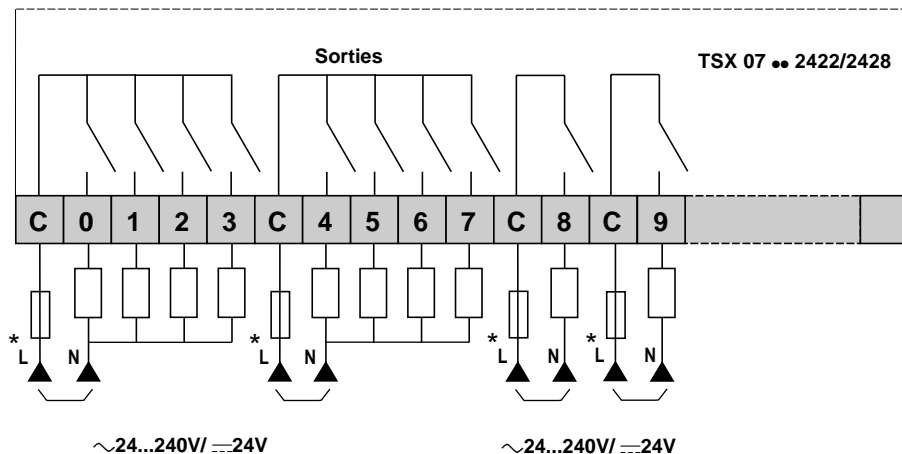
Raccordement des sorties relais (suite)

- automates 16 E/S : TSX 07 •1 1622, TSX 07 •1 1628



* fusible à calibrer selon la charge

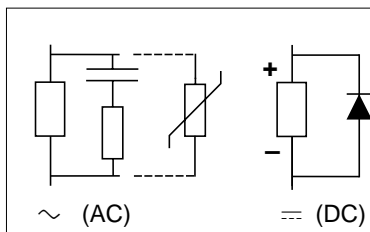
- automates 24 E/S : TSX 07 •1 2422, TSX 07 •1 2428



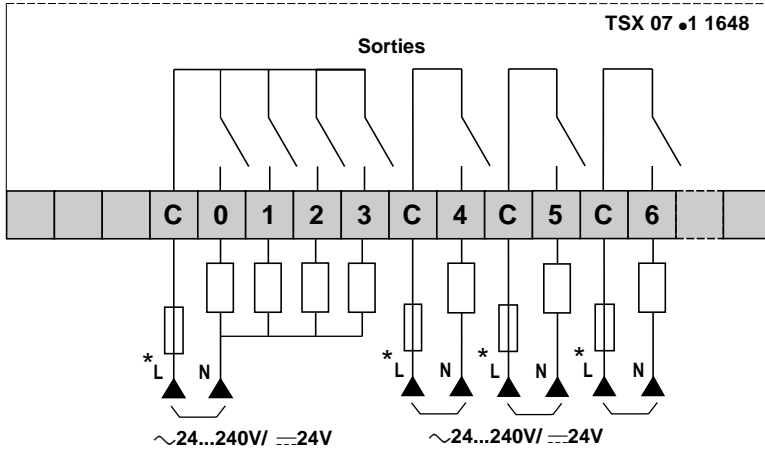
* fusible à calibrer selon la charge

Protections obligatoires à prévoir aux bornes de chaque pré-actionneur

- circuit RC ou écréteur de type MOV (ZNO) en courant alternatif,
- diode de décharge en courant continu.



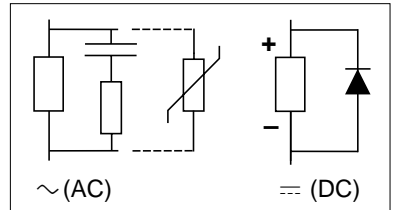
3.4-2 Raccordement des sorties relais sur automate TSX 07.1 1648



* fusible à calibrer selon la charge

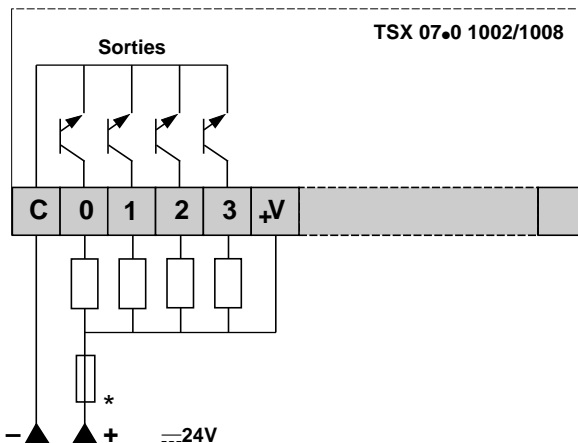
Protections obligatoires à prévoir aux borne de chaque pré-actionneur

- circuit RC ou écrêteur de type MOV (ZNO) en courant alternatif
- diode de décharge en courant continu.



3.4-3 Raccordement des sorties transistors, logique négative

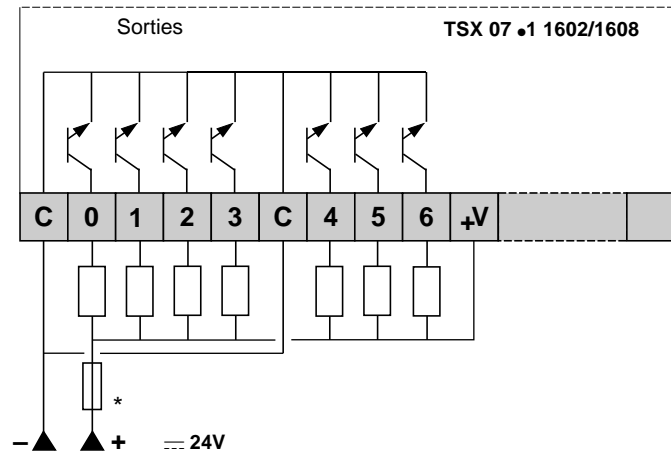
- automates 10 E/S : TSX 07.0 1002 et TSX 07.0 1008



* fusible à calibrer selon la charge

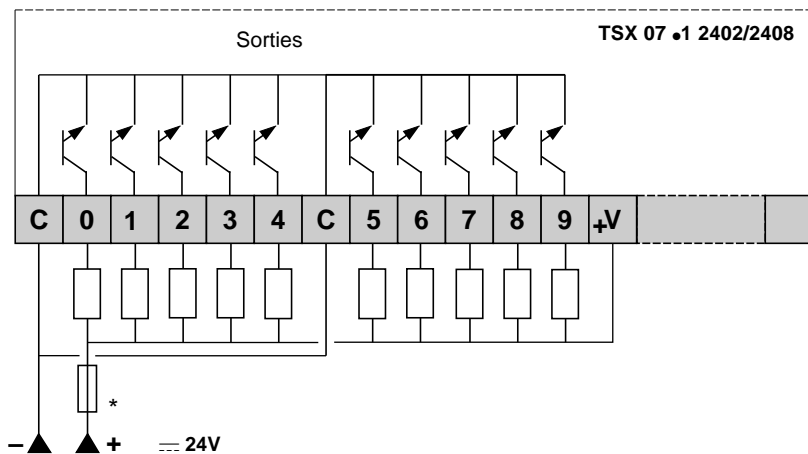
Raccordement des sorties transistors, logique négative (suite)

- automates 16 E/S : TSX 07 •1 1602 et TSX 07 •1 1608



* fusible à calibrer selon la charge

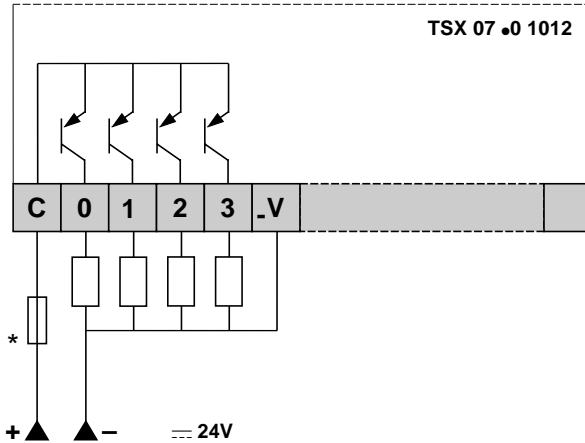
- automates 24 E/S : TSX 07 •1 2402 et TSX 07 •1 2408



* fusible à calibrer selon la charge

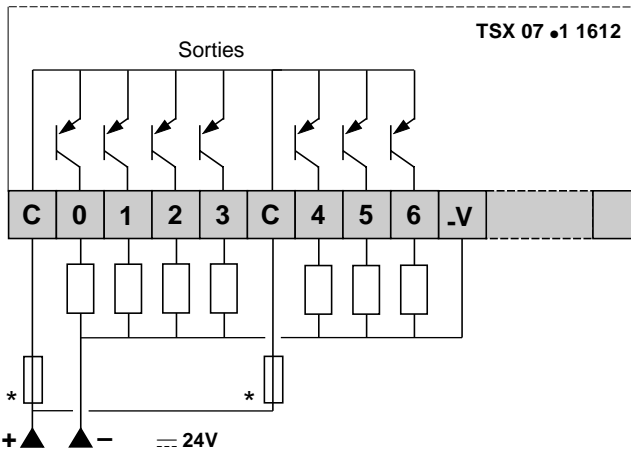
3.4-4 Raccordement des sorties transistors, logique positive

- automate 10 E/S : TSX 07 •0 1012



* fusible à calibrer selon la charge

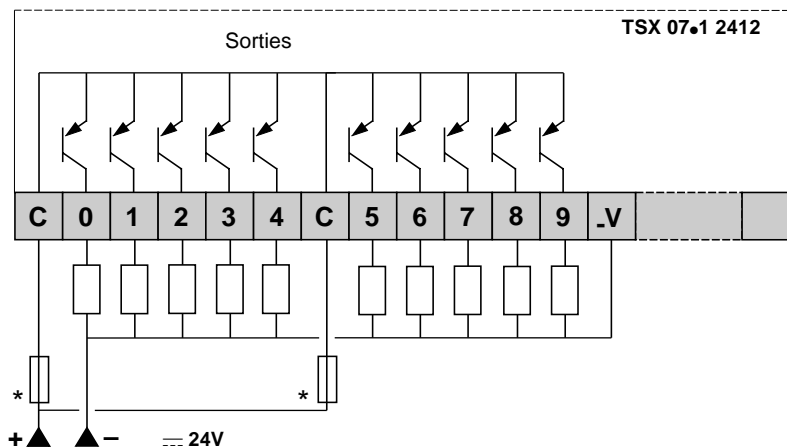
- automate 16 E/S : TSX 07 •1 1612



* fusible à calibrer selon la charge

Raccordement des sorties transistors, logique positive (suite)

- automate 24 E/S : TSX 07 •1 2412



* fusible à calibrer selon la charge

3.5 Raccordement d'une extension d'entrées/sorties

L'extension d'entrées/sorties se raccorde à l'automate de base par l'intermédiaire d'un câble paire torsadée/blindée :

- câble de longueur 30 cm : référence TSX CA 0003,
- pour des longueurs supérieures utiliser :
 - soit le câble UNI-TELWAY double paire torsadée/blindée :
TSX STC 50 : longueur 50 mètres ou TSX STC 200 : longueur 200 mètres
 - soit un câble **paire torsadée/blindée** dont les caractéristiques principales sont définies ci-après :

Caractéristiques mécaniques :

âmes en cuivre étamé : jauges 18 à 24

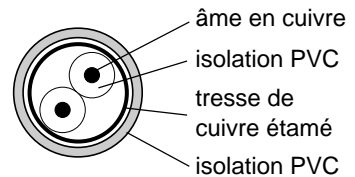
blindage en cuivre étamé

Caractéristiques électriques :

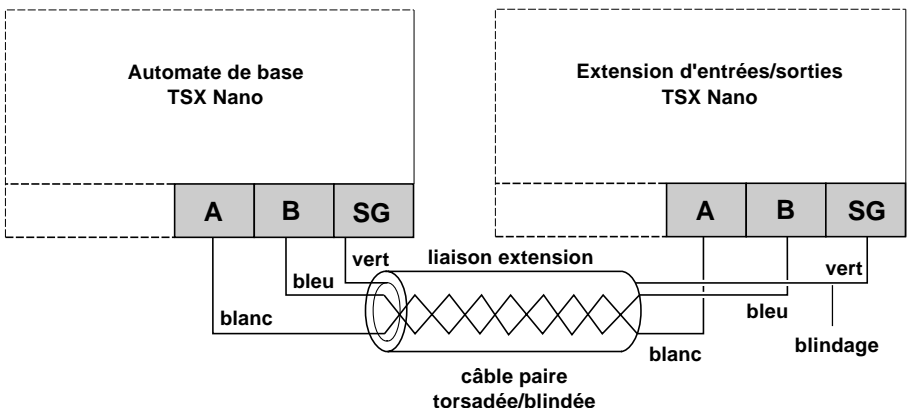
résistance linéique d'un fil : $\leq 85 \Omega / \text{Km}$

résistance linéique du blindage : $\leq 12 \Omega / \text{Km}$

Structure



La distance maximale autorisée entre l'automate de base et l'extension d'entrées/sorties est de 200 mètres.



Note :

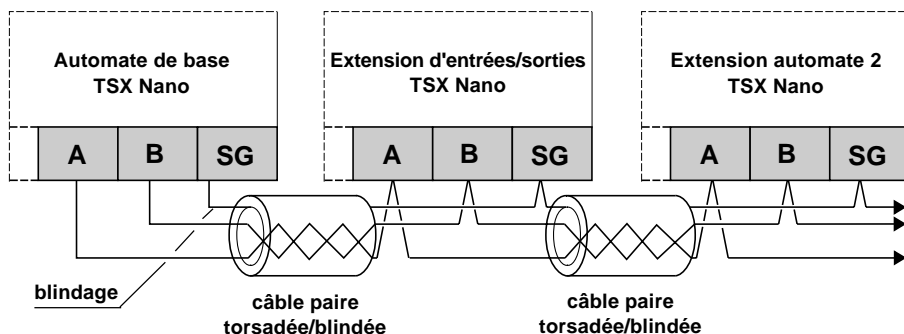
Les couleurs indiquées correspondent à un raccordement à l'aide du câble TSX CA 0003.

3.6 Raccordement des extensions automatées

Comme l'extension d'entrées/sorties, les extensions automate se raccordent à l'automate de base par l'intermédiaire d'un câble paire torsadée/blindée (voir page précédente pour le type de câble à utiliser).

La distance maximale autorisée entre l'automate de base et la dernière extension automate est de 200 mètres.

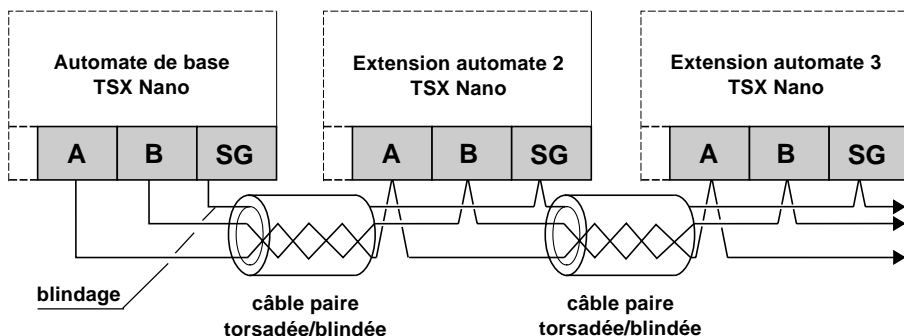
- Raccordements d'une extension d'entrées/sorties associées à l'automate de base et d'une extension automate



Note :

Dans le cas où les extensions automatées 3 et 4 sont utilisées, la continuité de la liaison extension est assurée par un câble paire torsadée/blindée avec raccordements identiques à ceux réalisés entre l'extension d'entrées/sorties et l'extension automate 2.

- Raccordements avec uniquement des extensions automatées associées à l'automate de base

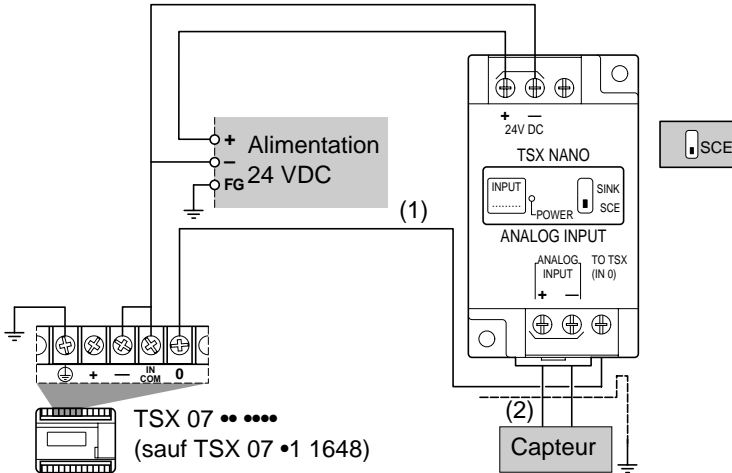


Note :

Dans le cas où l'extension automate 4 est utilisée, la continuité de la liaison extension est assurée par un câble paire torsadée/blindée avec raccordements identiques à ceux réalisés entre les extensions automatées 2 et 3.

3.7 Raccordement des entrées analogiques

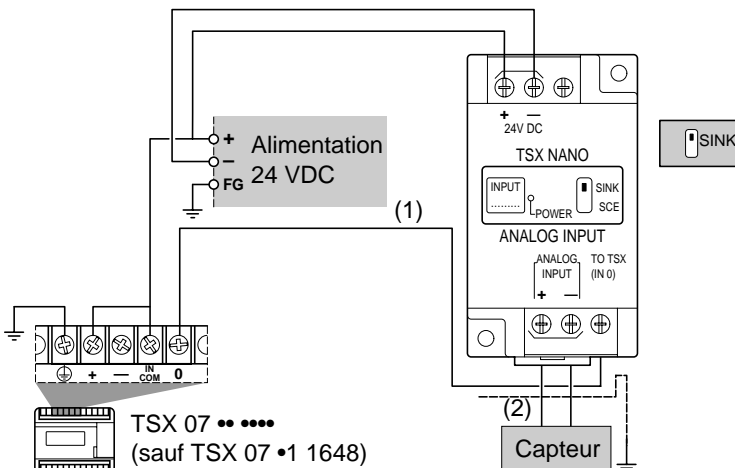
3.7-1 Raccordement avec l'entrée 0 du TSX Nano câblée en sink (logique positive)



(1) 2 mètres maximum avec câble blindé Ø 0,5 mm

(2) 50 mètres maximum avec câble blindé Ø 0,5 mm (blindage connecté coté module)

3.7-2 Raccordement avec l'entrée 0 du TSX Nano câblée en source (logique négative)

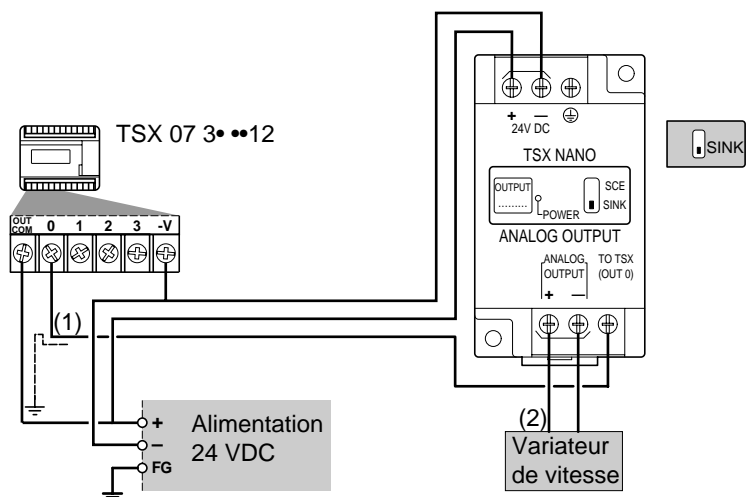


(1) 2 mètres maximum avec câble blindé Ø 0,5 mm

(2) 50 mètres maximum avec câble blindé Ø 0,5 mm (blindage connecté coté module)

3.8 Raccordement des sorties analogiques

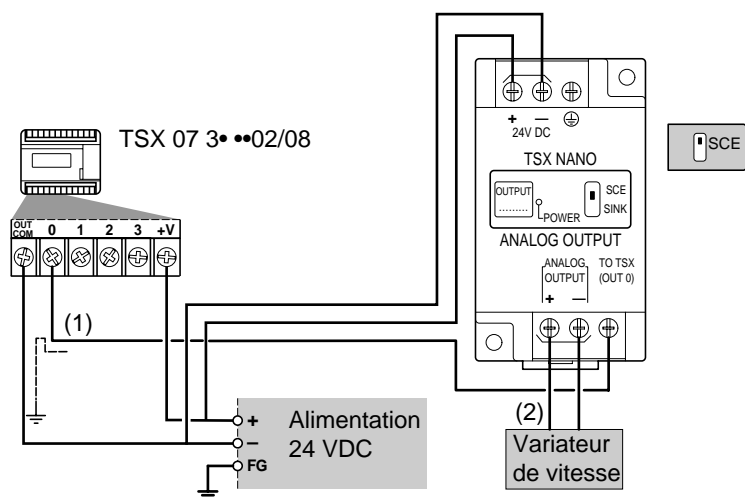
3.8-1 Raccordement avec la sortie 0 source du TSX Nano (logique positive)



(1) 2 mètres maximum avec câble blindé Ø 0,5 mm

(2) 50 mètres maximum avec câble blindé Ø 0,5 mm (blindage connecté coté module)

3.8-2 Raccordement avec la sortie 0 sink du TSX Nano (logique négative)



(1) 2 mètres maximum avec câble blindé Ø 0,5 mm

(2) 50 mètres maximum avec câble blindé Ø 0,5 mm (blindage connecté coté module)

(3) Ces références feront l'objet d'une prochaine commercialisation courant 1997.

4.1 Entrée RUN/STOP

Principe

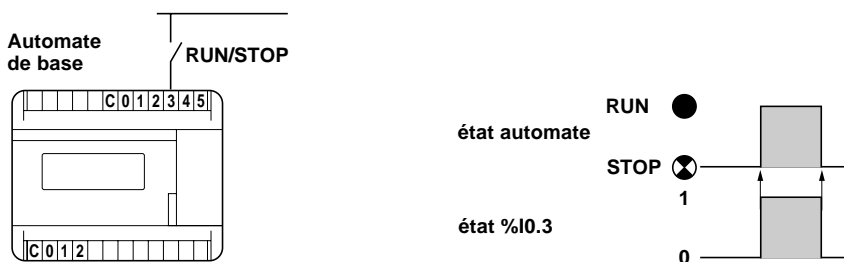
L'entrée RUN/STOP permet le lancement (RUN) ou l'arrêt (STOP) de l'exécution du programme.

Sur un automate de base ou une extension automate, l'une des 6 premières entrées (%I0.0 à %I0.5) peut, après configuration à l'aide du terminal, être affectée à cette fonction. L'entrée physique de mise en STOP est prioritaire par rapport à la mise en RUN à partir d'un terminal connecté.

Sous tension, un front montant sur l'entrée RUN/STOP met l'automate en RUN. L'état 0 sur l'entrée RUN/STOP force l'automate en STOP.

Si l'entrée RUN/STOP est à l'état 0, toute demande de mise en RUN à partir d'un terminal est ignorée.

Ex. : interrupteur RUN/STOP sur entrée %I0.3



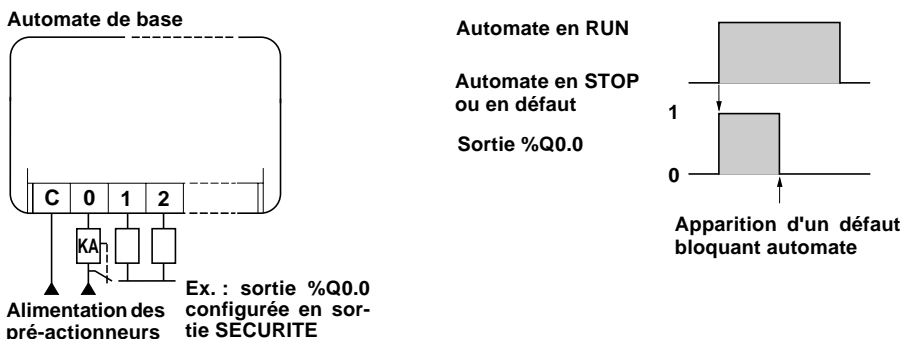
4.2 Sortie SECURITE

Principe

Dès la mise en RUN de l'automate et si aucun défaut bloquant (voir annexe A.6) n'est détecté, la sortie sécurité passe à l'état 1. Elle peut être utilisée dans les circuits de sécurité externes à l'automate, comme par exemple asservir :

- l'alimentation des préactionneurs de sorties.
- l'alimentation de l'automate.

Sur un automate de base ou une extension automate, l'une des 4 premières sorties (%Q0.0 à %Q0.3) peut, après configuration à l'aide du terminal, être affectée à la fonction SECURITE.



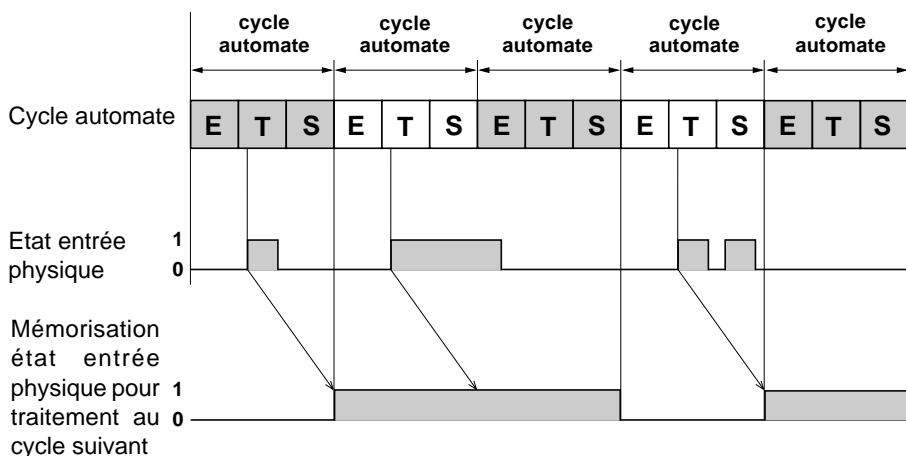
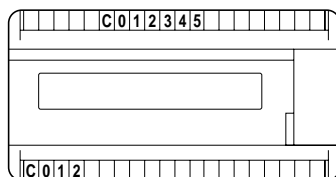
4.3 Entrées mémorisation d'état

Principe

Sur une impulsion de durée inférieure à un cycle et de valeur supérieure ou égale à 100 μ s (1), l'automate va mémoriser l'impulsion qui sera prise en compte au cycle suivant.

Sur un automate de base ou une extension automate, chacune des 6 premières entrées (%I0.0 à %I0.5) peut, après configuration à l'aide du terminal, être affectée à la fonction spécifique de mémorisation d'état.

Automate de base



Légende :

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

Note :

Une impulsion de durée supérieure au temps de cycle sera traitée comme une impulsion reçue sur une entrée standard.

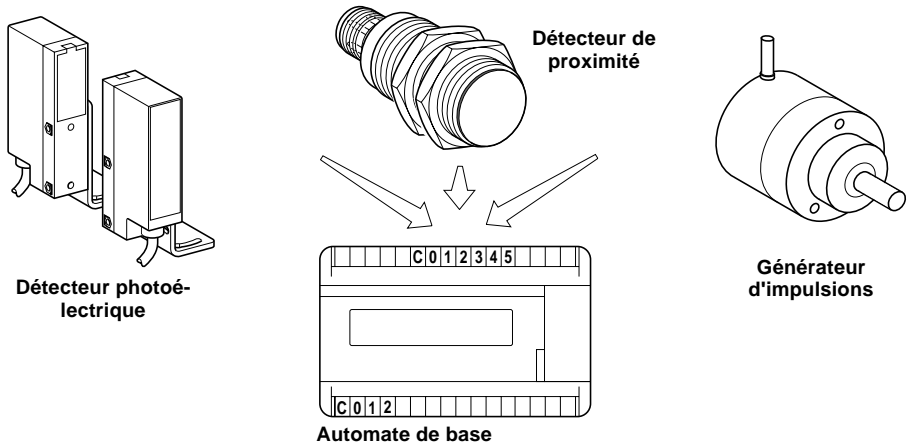
(1) voir intercalaire A, ch 1.7-1

4.4 Entrées/sorties liées au comptage rapide

L'automate TSX Nano dispose d'un compteur rapide pouvant être utilisé de trois manières différentes :

- utilisation en compteur rapide (fréquence maximale 10 kHz),
- utilisation en fréquencemètre (fréquence maximale 10 kHz) permet notamment la gestion d'un module d'entrée analogique,
- utilisation en compteur/décompteur rapide (fréquence maximale 1 kHz).

Les capteurs à utiliser sur les entrées comptage/décomptage %I0.0 et %I0.3 devront être à sorties statiques. L'utilisation de capteurs avec sorties à contacts est interdite sur ces entrées (prise en compte des rebonds compte tenu de leur faible immunité).



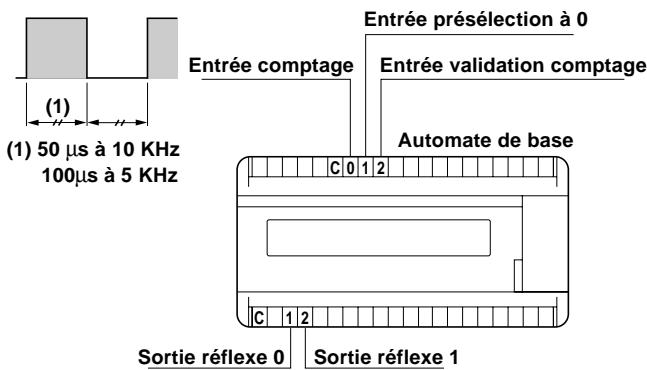
Fonctions	Entrées					Sorties	
	%I0.0	%I0.1	%I0.2	%I0.3	%I0.4	%Q0.1	%Q0.2
Comptage	●	-	-	-	-	-	-
Présélection comptage (remise à 0 compteur)	-	● (1)	-	-	-	-	-
Validation et arrêt comptage/décomptage	-	-	● (1)	-	-	-	-
Décomptage	-	-	-	●	-	-	-
Lecture valeur courante	-	-	-	-	● (1)	-	-
Réflexe 0	-	-	-	-	-	● (1)	-
Réflexe 1	-	-	-	-	-	-	● (1)

Le paramétrage de la fonction à réaliser (compteur rapide, fréquencemètre, compteur/décompteur) est réalisé à partir d'un bloc fonction spécifique %FC (voir chapitre 3.3-5 - intercalaire B).

(1) Ces entrées/sorties sont optionnelles, leur utilisation doit être déclarée en configuration.

4.4-1 Utilisation en compteur rapide

La fonction compteur rapide permet un comptage à une fréquence maximale de 10 kHz ou 5 kHz selon le choix effectué en configuration, avec possibilité de comptage de 0 à 65535 points.



Le compteur reçoit les impulsions à compter sur l'entrée automate (%I0.0). Si l'entrée validation comptage (%I0.2) est à l'état 1, les impulsions sont prises en compte par le compteur et la valeur de comptage (valeur courante FC.V) est comparée en permanence à 1 ou 2 seuils FC.S0 et FC.S1 définis en configuration et modifiables par programme.

La remise à 0 du compteur est provoquée sur front montant de l'entrée (%I0.1).

Les sorties réflexes 0 et 1 (%Q0.1 et %Q0.2) sont pilotées directement par le compteur rapide (sans attendre la mise à jour des sorties en fin de cycle) selon une matrice définie en configuration.

Sortie	FC.V < seuil 0 < seuil 1	seuil 0 ≤ FC.V ≤ seuil 1	seuil 0 < seuil 1 < FC.V
%Q0.1	0 ou 1	0 ou 1	0 ou 1
%Q0.2	0 ou 1	0 ou 1	0 ou 1

Certaines commandes (validation comptage, présélection à 0 de la valeur courante), peuvent également être exécutées à partir du programme utilisateur par des instructions spécifiques.

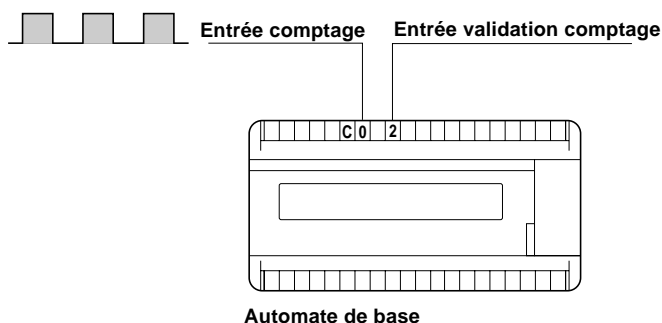
Note :

La mise en œuvre logicielle (configuration, instructions spécifiques, ...) et le chronogramme de fonctionnement du compteur rapide sont explicités chapitre 3.3-5 - intercalaire B.

4.4-2 Utilisation en fréquencesmètre

La fonction fréquencesmètre permet de mesurer la fréquence (en Hz) d'un signal périodique. Le principe de la mesure consiste à compter le nombre d'impulsions reçues dans un temps défini à partir d'une base de temps. La gamme de fréquence pouvant être mesurée s'étend de 1 Hz à 10 KHz. Cette fonction permet notamment de gérer un module d'entrée analogique.

Base de temps	Gamme de mesure	Précision	Rafraîchissement
100ms	10Hz-10kHz	0,1% pour 10kHz 10% pour 100Hz	10 fois par seconde
1s	1Hz-10kHz	0,01% pour 10kHz 10% pour 10Hz	1 fois par seconde



Le fréquencesmètre reçoit les impulsions sur l'entrée automate (%I0.0). Si l'entrée validation comptage (%I0.2) est à l'état 1, les impulsions sont prises en compte par le compteur et la valeur de comptage (valeur courante FC.V) évolue pendant la période de mesure. En fin de période de mesure, il y a lecture de la valeur de comptage (valeur courante FC.V) correspondant à la fréquence.

L'entrée (%I0.1) permet la mise à 0 de la valeur courante %FC.V.

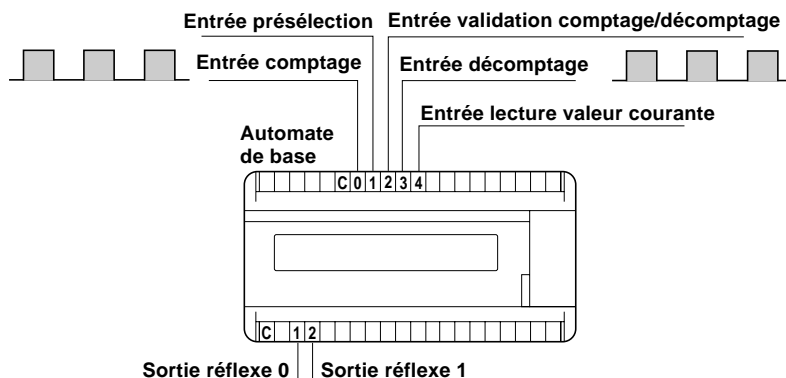
La commande (validation comptage), peut également être exécutée à partir du programme utilisateur par une instruction spécifique.

Note :

La mise en œuvre logicielle (configuration, instructions spécifiques, ...) et le chronogramme de fonctionnement du fréquencesmètre sont explicités chapitre 3.3-5 - intercalaire B.

4.4-3 Utilisation en compteur/décompteur

La fonction compteur/décompteur rapide permet un comptage/décomptage à une fréquence maximale de 1 kHz avec possibilité de comptage/décomptage entre 0 à 65535 points.



Le compteur reçoit les impulsions à compter sur l'entrée automate (**%I0.0**) et les impulsions à décompter sur l'entrée automate (**%I0.3**). Si l'entrée validation comptage/décomptage (**%I0.2**) est à l'état 1, les impulsions sont prises en compte et la valeur de comptage/décomptage (valeur courante FC.V) est comparée en permanence à 1 ou 2 seuils FC.S0 et FC.S1 définis en configuration et modifiables par programme.

L'information compteur en comptage ou en décomptage est disponible dans un bit du mot système SW111.

La valeur de présélection (0 à 65535), définie en configuration et modifiable par programme est chargée dans la valeur courante sur front montant de l'entrée (**%I0.1**). L'entrée (**%I0.4**) permet la lecture à la volée de la valeur courante FC.V.

Les sorties réflexes 0 et 1 (**%Q0.1** et **%Q0.2**) sont pilotées directement par le compteur/décompteur rapide (sans attendre la mise à jour des sorties en fin de cycle) selon une matrice définie en configuration (voir chapitre 4.4-1).

Certaines commandes (validation comptage/décomptage, présélection, ...), peuvent également être exécutées à partir du programme utilisateur par des instructions spécifiques.

Note :

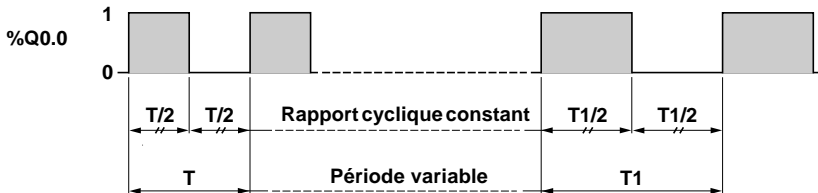
La mise en œuvre logicielle (configuration, instructions spécifiques, ...) et le chronogramme de fonctionnement du compteur/décompteur sont explicités chapitre 3.3-5 - intercalaire B.

4.5 Sortie PULSE : génération train d'impulsions

Sur un automate de base ou une extension automate, la sortie %Q0.0 peut, après configuration à l'aide du terminal, être affectée à la fonction spécifique "PULSE".

Principe

Un bloc fonction paramétrable (%PLS) permet la génération sur la sortie %Q0.0 d'un signal de période variable mais de rapport cyclique constant égal à 50% de la période.



La valeur de la période T et le nombre d'impulsions à générer sont définis par configuration du bloc fonction %PLS.

Paramètres de configuration :

- Définition de la période : $T = BT \times \%PLS.P$
BT = base de temps
 - 0,1 ms (utilisable uniquement sur automate à sorties transistors),
 - 10 ms (valeur par défaut), ou 1s.
 %PLS.P = valeur de présélection :
 - $0 < \%PLS.P < 32767$ avec BT=10ms ou 1s
 - $0 < \%PLS.P < 255$ avec BT = 0,1ms
- Définition du nombre d'impulsions à générer sur la sortie %Q0.0 : %PLS.N
Le nombre d'impulsions de période T à générer (%PLS.N) peut être limité ou illimité selon la définition faite en configuration :
 - $0 < \%PLS.N < 32767$
 - %PLS.N=0 : génération illimitée

Gamme de période pouvant être obtenue :

- 0,2 ms à 26 ms par pas de 0,1ms (38 Hz à 4,9 kHz).
- 20 ms à 5,45 mn par pas de 10 ms,
- 2 s à 9,1 heures par pas de 1s.

Note : le paramétrage complet du bloc fonction %PLS est défini intercalaire B - chapitre 3.3-4

Exemple d'application : commande d'un moteur pas à pas

Automate de base

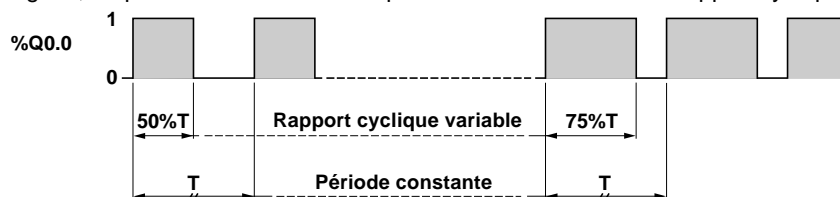


4.6 Sortie PWM : modulation de largeur d'impulsions

Sur un automate de base ou une extension automate, la sortie %Q0.0 peut, après configuration à l'aide du terminal, être affectée à la fonction spécifique "PWM". Cette fonction, à partir des versions TSX07 V3, permet notamment de gérer les modules de sortie analogiques.

Principe

Un bloc fonction paramétrable (%PWM) permet la génération sur la sortie %Q0.0 d'un signal, de période constante avec possibilité de faire varier le rapport cyclique.



La valeur de la période T et le pourcentage du signal à l'état 1 sur une période sont définis par configuration du bloc fonction %PWM.

Paramètres de configuration :

- Définition de la période : $T = BT \times \%PWM.P$
 BT = base de temps :
 - 0,1ms (obligatoire pour gestion de sortie analogique; utilisable uniquement sur automate à sorties transistors),
 - 10ms (valeur par défaut), ou 1s $\%PWM.P$ = valeur de présélection (inutilisé pour la gestion de sortie analogique) :
 - $0 < \%PWM.P < 32767$ avec base de temps 10ms ou 1s
 - $0 < \%PWM.P < 255$ avec base de temps 0,1ms
- Définition du ratio de la période : $\%PWM.R = Tx(\%PWM.R/100)$
 $\%PWM.R$ donne le pourcentage de signal à l'état 1 sur une période ($0 < \%PWM.R < 100$).

Gamme de période pouvant être obtenue :

- 0,2 ms à 26 ms par pas de 0,2 ms (38 Hz à 4,9 kHz).
- 20 ms à 5,45 mn par pas de 10 ms,
- 2 s à 9,1 heures par pas de 1s.

Note :

le paramétrage complet du bloc fonction %PWM est défini intercalaire B - chapitre 3.3-3.

Exemple d'application : commande d'un gradateur de lumière

Automate de base



5.1 Caractéristiques des alimentations

Automates		TSX 0708/28/48	TSX 0702/12/22
Type alimentation		~	==
Tension d'alimentation	nominale	100 à 240V	24V
	limite	85 à 264V	19,2 à 30V
Fréquence	nominale	50/60 Hz	-
	limite	47 à 63 Hz	-
Puissance nécessaire		≤ 30VA	≤ 14W
Alimentation capteur (1) intégrée et protégée		== 24V /150 mA	-
Micro-coupures	durée	≤10ms	≤1 ms
Isolement primaire/terre		2000V eff.-50/60 Hz	2000V eff.-50/60 Hz
Conformité IEC 1131-2		oui	oui

(1) sauf sur automate TSX 07 •1 1648

5.2 Caractéristiques des entrées TOR 24VDC et 115VAC

Automates		TSX 07 .. 02/08/12/22/28		TSX 07 .1 1648
Type		24V (résistive)		115V (capacitive)
Logique (1)		positive	négative	-
Commun des capteurs		au + de l'alim.	au - de l'alim.	-
Nature		Isolées	Isolées	isolées
Valeurs nominales d'entrées	tension	24V	24V	110/120V
	courant	7 mA (2)	- 7mA (2)	10 mA
	courant de pointe à l'enclenchement	-	-	300 mA (pour U=110V)
	alimentation capteurs	19,2 à 30V (ondulation incluse)	-	-
	fréquence	-	-	50...60 Hz
Valeurs limites d'entrées	état 1 tension	$\geq 11V$	$\leq 8V$	$\geq 79 V$
	courant	$\geq 2,5 \text{ mA}$ pour U = 11V	$\geq 2,5 \text{ mA}$ pour U = 8V	$\geq 4\text{mA}$ pour U = 79V
	état 0 tension	$\leq 5V$	Ual - 5V	$\leq 20V$
	courant	$\leq 1,2 \text{ mA}$	$\leq 1,2 \text{ mA}$	$\leq 2 \text{ mA}$
	fréquence	-	-	47....63 Hz
Immunité	valeurs 0 à 1	12 ms	12 ms	12 ms
	par défaut 1 à 0	12 ms	12 ms	12 ms
	valeurs 0 à 1	100µs/3ms/12ms sur I0,0 à I0,7	-	-
	program-mables	375µs/3ms/12ms sur I0,8 à I0,13	-	-
	1 à 0	100µs/3ms/12ms sur I0,0 à I0,7	-	-
Isolement	entre groupe de voies	1500V eff. -50/60 Hz	-	1500V eff. -50/60 Hz
Compatibilité ddp 2fils		oui (TE)	oui (TE)	oui (TE)
Compatibilité ddp 3fils		oui	oui	-
Conformité IEC 1131-2		oui (type 1)	-	oui (type 1)

Note :

Les caractéristiques des entrées/sorties sont données pour un taux de charge de 100% pour les automates 10 E/S et de 80% pour les automates 16 et 24 E/S.

Taux de charge = nombre total d'E/S simultanément à l'état 1/ nombre total d'E/S de l'automate

(1) Logique positive ou négative selon câblage

(2) 13 mA (ou -13 mA) pour l'entrée %I.0.

5.3 Caractéristiques des sorties TOR transistors 24VDC

Automates		TSX 07 .. 02/08	TSX 07 .. 12
Nature		Transistors non protégés	Transistors protégés
Charges courant continu	tension nominale	$\overline{\text{---}}$ 24V	\sim 24V
	courant nominal	0,5 A	0,5 A
	voyant à fil de tungstène	$\leq 10W$	$\leq 10W$
Valeurs limites	tension (ondulation incluse)	19,2 à 30V	19,2 à 30V
	courant	0,625 A pour U = 30V	0,625 A pour U = 30V
Logique		négative, courant absorbé	positive, courant émis
Commun des charges		au + de l'alimentation	au - de l'alimentation
Courant de fuite à l'état 0		$\leq 1 \text{ mA}$	$\leq 1 \text{ mA}$
Tension de déchet à l'état 1		$\leq 1,5 \text{ V}$ pour I = 0,5 A	$\leq 2V$ pour I = 0,5 A
Temps de réponse sur charge résistive	0 à 1	$\leq 1 \text{ ms}$	$\leq 1 \text{ ms}$
	1 à 0	$\leq 1 \text{ ms}$	$\leq 1 \text{ ms}$
Protections incorporées	contre les courts circuits et surcharges	aucune (prévoir un fusible sur commun des pré-actionneurs)	oui
	contre les surtensions	oui	oui
	contre les inversions de polarité	oui	oui
Isolement	entre groupe de voies	1500V eff.-50/60 Hz	1500V eff.-50/60 Hz
Conformité à la norme IEC 1131-2		sans objet	oui

Notes :

- Les caractéristiques des entrées/sorties sont données pour un taux de charge de 100% pour les automates 10 E/S et de 80% pour les automates 16 et 24 E/S.
Taux de charge = nombre total d'E/S simultanément à l'état 1/ nombre total d'E/S de l'automate
- Il est possible de mettre 2 sorties en parallèles dans ce cas, il est nécessaire de mettre une diode de décharge aux bornes des sorties de l'automate (pas aux bornes de la charge).

5.4 Caractéristiques des sorties TOR relais

Automates		TSX 0722/28/48			
Nature		relais			
Charge courant alternatif	résistive régime AC12	tension	~ 24V	~ 48V	~ 110V ~ 220V
		courant	2A (1) 1A (2)	1A (2)	1A (3) 0,5A (4)
	inductive régime AC15	tension	~ 24V	~ 48V	~ 110V ~ 220V
		courant	1A (1) 0,5A (3)	0,5A (3) 0,2A (5)	0,45A (3) 0,090A (5)
Charge courant continu	résistive régime DC12	tension	--- 24V	-	-
		courant	1A (1)	-	-
	inductive régime DC13	tension	--- 24V	-	-
		courant	0,4A (3)	-	-
Temps de réponse	à l'enclenchement	≤ 5 ms			
	au déclenchement	≤ 10 ms			
Protections incorporées	contre les surcharges et les courts-circuits	aucune, prévoir un fusible par voie ou groupe de voies			
	contre les surtensions inductives en ~	aucune, montage obligatoire d' un circuit RC ou écréteur MOV (ZNO) en parallèle aux bornes de chaque pré-actionneur			
	contre les surtensions inductives en ---	aucune, montage obligatoire d'une diode de décharge en parallèle aux bornes de chaque pré-actionneur			
Isolement	entre groupe de voies	1500V eff. - 50/60 Hz			
Conformité IEC 1131-2		oui	oui	oui	oui

Note :

Les caractéristiques des entrées/sorties sont données pour un taux de charge de 100% pour les automates 10 E/S et de 80% pour les automates 16 et 24 E/S.

Taux de charge = nombre total d'E/S simultanément à l'état 1/ nombre total d'E/S de l'automate

(1) 0,3 x 10⁶ manoeuvres

(2) 0,5 x 10⁶ manoeuvres

(3) 1 x 10⁶ manoeuvres

(4) 2 x 10⁶ manoeuvres

(5) 10 x 10⁶ manoeuvres

5.5 Caractéristiques des E/S analogiques

Caractéristiques communes aux entrées et sorties analogiques

Alimentation	tension nominale	--- 24V
	tensions limites	--- 21V à 30V
	courant nominal à 24V	104 mA
	courant d'appel	10 A max
	puissance absorbée	2,5 W
Isolement	tension entre l'alimentation et la terre	~ 1500 V
	tension entre l'entrée ou la sortie et la terre	~ 1500 V
	résistance entre l'alimentation et la terre (500V)	> 10 MΩ
	résistance entre l'entrée ou la sortie et la terre (500V)	> 10 MΩ
Chocs	300 m/s ² , 3 chocs par axe, 3 axes	
Vibrations	5 à 55Hz, 60 m/s ² , 2 heures par axe, 3 axes	
Environnement climatique	température de fonctionnement	0 à 60 °C
	température de stockage	-25 °C à 70 °C
	humidité relative (sans condensation)	45 à 85 %
	altitude	0 à 2000m

Caractéristiques des entrées analogiques

Entrée	nombre de voie	1 (haut niveau)
	impédance d'entrée	6,6 MΩ (1) 250 Ω (2)
	tension admissible sans destruction	+/- 16 V
Conversion	méthode de conversion	U (V) → F (HZ)
	résolution	10 bits ou 12 bits
	temps de conversion	125 ms (3) /500ms (4)
	précision (pleine échelle)	+/- 1% à 25 °C
Sortie fréquence	tension nominale	--- 24V
	logique (commutateur en face avant)	positive ou négative
	protection contre les court circuits	non
Isolement	tension entre l'entrée et la sortie fréquence	~ 500 V
	résistance entre l'entrée et la sortie fréquence	>10 MΩ

(1) Modules 0/10V et -10/+10V

(2) module 4/20 mA

(3) Résolution 10 bits

(4) Résolution 12 bits

Caractéristiques des sorties analogiques

Sortie	nombre de voie	1 (haut niveau)
	valeur signal de sortie	0/10 V
	impédance de charge	$\leq 5 \text{ K}\Omega$ (1) $\leq 250 \Omega$ (2)
	tension admissible sans destruction	+/- 12 V
	protection contre les court circuits	oui (permanent)
Conversion	méthode de conversion	F (Hz) \rightarrow U (V)
	résolution	8 bits
	temps de conversion	de 0 à 90%: 500ms max
	précision (pleine échelle)	+/- 1% à 25 °C
Entrée fréquence	tension nominale	--- 24V
	logique (commutateur en face avant)	positive ou négative
	fréquence d'entrée	312,5 Hz
Isolement	tension entre la sortie et l'entrée fréquence	$\sim 500 \text{ V}$
	résistance entre la sortie et l'entrée fréquence	$>10 \text{ M}\Omega$

(1) Modules 0/10V et -10/+10V

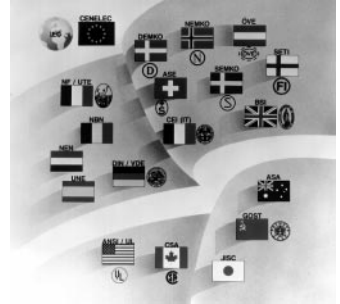
(2) module 4/20 mA

5.6 Conditions de service

5.6-1 Normes

Les automates TSX Nano ont été développés pour être conformes aux principales normes nationales et internationales, concernant les équipements électroniques de contrôle industriel.

- Prescriptions spécifiques automates programmables: caractéristiques fonctionnelles, immunité, robustesse, sécurité, ...
EN61131-2 (IEC1131-2) , CSA 22.2 , UL508
- Stricte limitation des nuisances électromagnétiques produites: Marquage CE (Directives Européennes Basse Tension et CEM)
- Qualités électriques et autoextinguibilité des matériaux isolants: UL 746C, UL 94, ...



5.6-2 Environnement, conditions normales de service

- **Environnement climatique, conditions normales**

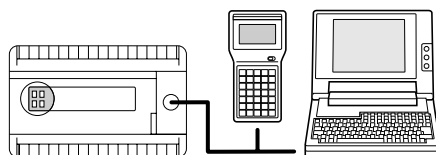
Température	Hygrométrie et altitude
Température de fonctionnement : 0 à 60°C (1)	Humidité relative : 5% à 95% (sans condensation)
Température de stockage: -25°C à + 70°C	Altitude : 0 à 2000 mètres

- **Tenue aux vibrations** : Selon IEC 68-2-6 Essais FC
- **Tenue aux chocs mécaniques** : selon IEC 68-2-27 Essais EA

(1) 0 à 55°C avec TSX 07 ●●●12 monté verticalement (sur plan vertical)

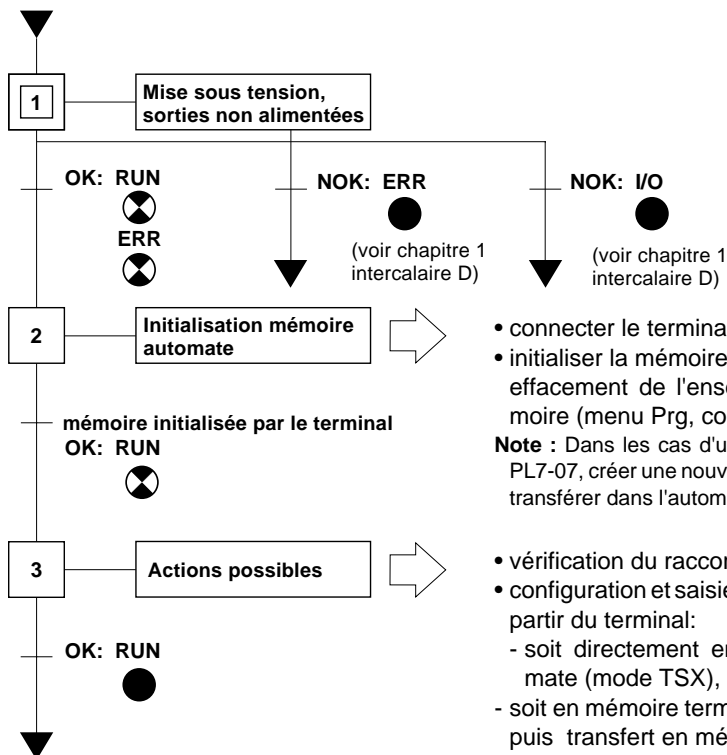
6.1 Procédure de première mise sous tension

Les nombreux auto-tests incorporés au TSX Nano assurent le contrôle permanent de son bon fonctionnement. Le résultat de ces auto-tests est visualisé en face avant de l'automate. Il peut être approfondi à l'aide du terminal (1). Le diagramme ci-dessous indique la procédure à suivre lors de la première mise sous tension.



Automate TSXNano

Terminaux



- connecter le terminal
- initialiser la mémoire automate par un effacement de l'ensemble de la mémoire (menu Prg, commande Clr).

Note : Dans les cas d'utilisation du logiciel PL7-07, créer une nouvelle application et la transférer dans l'automate.

- vérification du raccordement des E/S
- configuration et saisie du programme à partir du terminal:
 - soit directement en mémoire automate (mode TSX),
 - soit en mémoire terminal (mode FTX), puis transfert en mémoire automate.
- mise en RUN par entrée RUN/STOP %I0. si configurée ou par le terminal.
- ...

Légende



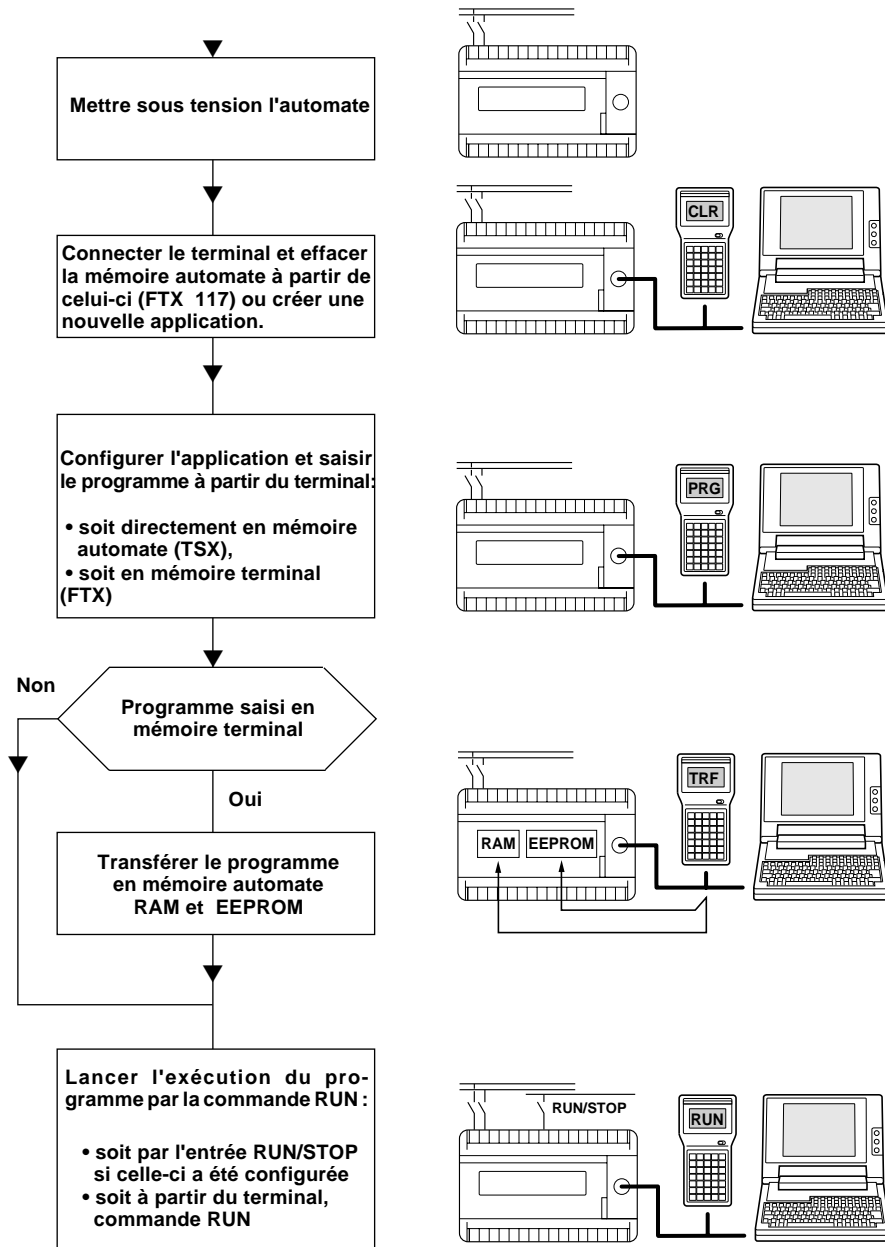
Voyant clignotant



Voyant allumé fixe

- (1) terminal : terminal FTX 117 ou compatible PC / terminal FTX 417/517 équipé du logiciel PL7-07.

Synthèse



6.2 Vérification du raccordement des entrées/sorties

• Principe

Cette vérification consiste à s'assurer que :

- les informations en provenance des capteurs sont prises en compte par les entrées et transmises au processeur,
- les ordres de commande du processeur activent les sorties et sont transmis au pré-actionneurs correspondants.

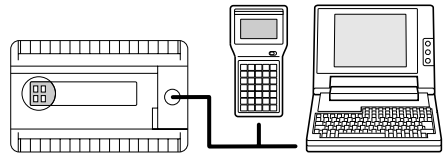
• Recommandations

Afin d'éviter tout mouvement aléatoire de machine, il est recommandé de :

- retirer les fusibles puissances des commandes moteurs,
- couper les arrivées pneumatiques et hydrauliques.

• Procédure

- effectuer la première mise sous tension comme indiqué chapitre 5.2 et s'assurer que le voyant I/O n'est pas allumé en fixe,



Automate TSX Nano

Terminaux

- laisser l'automate en STOP,
- demander le mode Data (REGLAGE) sur le terminal FTX 117, ou terminal équipé du logiciel PL7-07,
- modifier l'état du bit système %S8 (sécurité sorties) : le mettre à l'état 0,
- vérification des entrées par activation de chaque capteur :
 - vérifier que le voyant face avant de l'entrée correspondante change d'état,
 - vérifier sur l'écran du terminal que le bit correspondant change lui aussi d'état.
- vérification des sorties à l'aide du terminal :
 - forcer à l'état 1 le bit correspondant à chaque sortie
 - vérifier que le voyant de la sortie correspondante change d'état ainsi que le pré-actionneur associé.
- à l'aide du terminal :
 - supprimer tous les forçages,
 - repositionner le bit système %S8 à l'état 1 (les sorties sont positionnées à 0).

Note :

L'ensemble de cette procédure peut, si l'automate ne comporte pas de programme application, être effectué en RUN. Dans ce cas, le bit %S8 peut rester à l'état 1 (état par défaut).

7.1 Coupures et reprises secteur

- **Caractéristiques des coupures secteur**
 - si coupure < autonomie alimentation : exécution normale du programme,
 - si coupure > autonomie alimentation : processeur hors tension avec sauvegarde du contexte (données et programme).
- **Reprise à chaud : redémarrage de l'automate avec le contenu de la mémoire de données dans l'état au moment de la coupure secteur.**

Enumération des causes possibles :

- redémarrage de l'automate après coupure secteur > autonomie de l'alimentation,
- mise à l'état 1 du bit système %S1 par le programme ou le terminal.

Conséquences à la reprise

- bit système %S1 mis à l'état 1,
 - **mise à l'état 0 des bits des entrées non forcées (attention: cette mise à zéro génère un "faux" front montant sur une entrée physiquement à 1 lors de la reprise à chaud, voir intercalaire B, ch 2.1-2).**
 - mise à l'état 0 de tous les bits d'entrées/sorties,
 - mise à l'état 0 de tous les bits internes non sauvegardés (%M64 à %%127),
 - maintien de l'état des bits internes sauvegardés (%M0 à %M63), des valeurs courantes des blocs fonctions (temporisateurs, compteurs, ...),
 - reprise du cycle d'exécution là où il s'est arrêté lors de la coupure secteur sans mise à jour des sorties en fin de cycle puis reprise normale du cycle :
Acquisition des entrées → Traitement du programme → Mise à jour des sorties et mise à l'état 0 du bit système %S1.
- **Reprise à froid : redémarrage de l'automate avec perte du contenu de la mémoire de données.**

Enumération des causes possibles :

- batterie de sauvegarde défectueuse,
- mise à l'état 1 du bit système %S0 par le programme ou le terminal,
- initialisation de l'automate par le terminal.

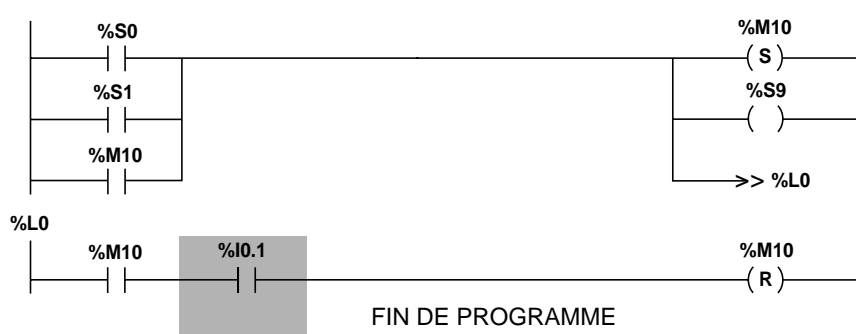
Conséquences à la reprise

- bit système %S0 mis à l'état 1,
- mise à l'état 0 de tous les bits internes, bits d'entrées/sorties, mots internes,
- mise à l'état 0 des valeurs courantes des blocs fonctions (temporisateurs, compteurs,...),des registres, pas à pas,... ,
- perte des valeurs de réglage modifiées par le terminal avec reprise des valeurs de présélection définies en configuration,
- initialisation des bits et mots système (sauf %S0 et les données de l'horodateur),
- annulation des forçages,
- réinitialisation du Grafset,
- reprise du cycle d'exécution en début de cycle :
Acquisition des entrées → Traitement du programme → Mise à jour des sorties et mise à l'état 0 du bit système %S0.

Exemple de programmation

Afin d'éviter un redémarrage automatique de l'automatisme lors d'une reprise secteur, le programme ci-dessous impose une intervention humaine par appui sur un bouton "REPRISE" et le maintien à l'état 0 des sorties pendant la coupure. Ce programme doit s'ajouter au programme de l'application.

Une partie de ce programme doit se trouver en début de programme (zone de traitement préliminaire), l'autre partie en fin de programme (zone de traitement postérieure).



000	LD %S0	Si %S0 est au niveau logique 1 : reprise à froid] dans zone de traitement préliminaire (début de programme)
001	OR %S1	OU %S1 est au niveau logique 1 : reprise à chaud	
002	OR %M10	OU %M10 est au niveau logique 1 : auto maintien	
003	S %M10	Mise à 1 du bit interne %M10	
004	ST %S9	Mise à 0 des sorties	
005	JMPC %L0	Saut à l'étiquette %L0] Programme application
006			
007			

098] dans zone de traitement postérieur (fin de programme)
099			
100	%L0:	Adresse de saut	
101	LD %M10	Si %M10 est au niveau logique 1	
102	AND %I0.1	ET l'entrée 1 est au niveau logique 1 : REPRISE	
103	R %M10	Mise à 0 du bit interne %M10	
104	END	Fin de programme	

7.2 Initialisation automate

Présentation

Une initialisation par programme peut être obtenue par la mise à l'état 1 du bit système %S0, correspondant à une reprise à froid (voir chapitre 7.1). Lors d'une reprise à chaud, il peut être nécessaire d'effectuer une initialisation. L'exemple ci-dessous montre comment programmer celle-ci. Une initialisation peut également être demandée à partir du terminal par la commande INIT.

Programmation



LD %S1

ST %S0

Si %S1 = 1 (reprise à chaud), alors mettre %S0 à 1 pour effectuer une initialisation. Ces deux bits sont remis à 0 par le système à la fin du cycle suivant.

IMPORTANT

Le bit système %S0 ne doit pas être positionné à 1 pendant plus d'un cycle automate.

7.3 Sauvegarde du programme et des données

Sauvegarde dans la RAM

Le programme utilisateur et les données sont contenus dans la mémoire RAM de l'automate. Cette mémoire a une autonomie de 30 jours assurée par une batterie interne à l'automate.

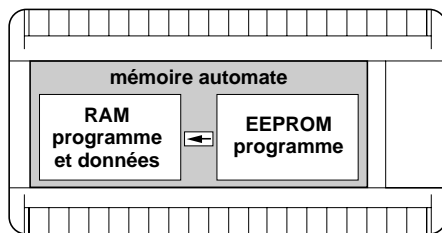
ATTENTION

Cette autonomie est garantie à condition de charger la batterie au moins 6 heures consécutives avant l'arrêt de l'automate.

Sauvegarde EEPROM

Elle permet d'éviter le risque d'altération du programme écrit en mémoire RAM sur défaut batterie ou coupure secteur supérieure à 30 jours.

A la mise sous tension, l'automate vérifie la conformité des programmes contenus en mémoires RAM et EEPROM. En cas de non conformité, le programme contenu dans la mémoire EEPROM est automatiquement transféré dans la mémoire RAM si l'option MST (autochargement) a été choisie (voir intercalaire C, ch 12.4).



ATTENTION

Une fois l'application mise au point, il est recommandé de la transférer dans l'EEPROM avec l'option MST.

Chapitre	Page
1 Introduction	1/1
1.1 Généralités	1/1
1.2 Langage Liste d'instructions (List ou IL)	1/1
1.3 Grafcet	1/3
1.4 Langage à contacts (Ladder ou LD)	1/4
1.4-1 Principes de programmation	1/6
1.4-2 Réversibilité	1/8
1.4-3 Conventions de programmation en langage List réversible	1/9
2 Instructions combinatoires et séquentielles	2/1
2.1 Traitement booléen	2/1
2.1-1 Définition des principaux objets bits	2/1
2.1-2 Présentation des instructions booléennes	2/2
2.1-3 Instructions de chargement LD, LDN, LDR, LDF	2/4
2.1-4 Instructions d'affectation ST, STN, S, R	2/4
2.1-5 Instructions ET Logique : AND, ANDN, ANDR, ANDF	2/5
2.1-6 Instructions OU Logique : OR, ORN, ORR, ORF	2/5
2.1-7 Instructions OU Exclusif : XOR, XORN, XORR, XORF	2/6
2.1-8 Instruction Négation : N	2/7
2.1-9 Utilisation des parenthèses	2/7
2.1-10 Instructions MPS, MRD, MPP	2/9
2.1-11 Instructions OPEN et SHORT spécifiques au langage à contacts	2/10
2.2 Blocs fonction standards	2/11
2.2-1 Objets bits et mots associés aux blocs fonction standards	2/11
2.2-2 Principes de programmation	2/12
2.2-3 Blocs fonction temporisateur %Tmi	2/13
2.2-4 Blocs fonction compteur/décompteur %Ci	2/17
2.2-5 Blocs fonction registre %Ri	2/20
2.2-6 Blocs fonction programmeur cyclique %DRi	2/23

Chapitre	Page
2.3 Instructions Grafcet	2/26
2.3-1 Description	2/26
2.3-2 Structure d'un programme	2/28
2.4 Instructions sur programme	2/29
2.4-1 Instructions de fin de programme END, ENDC, ENDCN	2/29
2.4-2 Instruction NOP	2/29
2.4-3 Instructions de saut JMP, JMPC, JMPCN vers une étiquette %Li:	2/30
2.4-4 Instructions sur sous-programme SRn, SRn:, RET	2/31
2.4-5 Instructions de relais maître MCS et MCR	2/32

3 Instructions numériques et spécifiques	3/1
---	------------

3.1 Traitement numérique	3/1
3.1-1 Définition des principaux objets mots	3/1
3.1-2 Objets structurés	3/3
3.1-3 Présentation des instructions numériques	3/5
3.1-4 Instructions d'affectation	3/5
3.1-5 Instructions de comparaison	3/8
3.1-6 Instructions arithmétiques	3/9
3.1-7 Instructions logiques	3/11
3.1-8 Instructions de décalage	3/12
3.1-9 Instructions de conversion	3/13
3.2 Points de réglage analogique	3/14
3.3 Blocs fonction spécifiques	3/15
3.3-1 Objets bits et mots associés aux blocs fonction spécifiques	3/15
3.3-2 Principes de programmation	3/15
3.3-3 Sortie modulation de largeur %PWM	3/16
3.3-4 Sortie générateur d'impulsions %PLS	3/18
3.3-5 Fonctions comptage rapide, fréquencemètre et compteur/décompteur %FC	3/20
3.3-6 Emission/Réception de message et contrôle des échanges	3/29
3.3-7 Blocs fonction registre à décalage bit %SBRi	3/44
3.3-8 Blocs fonction pas à pas %SCi	3/46

Chapitre	Page
3.4 Communication inter-automates	3/48

4 Gestion des E/S analogiques	4/1
--------------------------------------	------------

4.1 Présentation	4/1
------------------	-----

4.2 Modules d'entrée analogique	4/1
---------------------------------	-----

4.2-1 Configuration des entrées analogiques	4/1
---	-----

4.2-2 Programmation des entrées analogiques	4/2
---	-----

4.2-3 Temps de réponse des entrées analogiques	4/3
--	-----

4.2-4 Exemple de programmation des entrées analogiques	4/4
--	-----

4.2-5 Caractéristiques des entrées analogiques	4/4
--	-----

4.3 Modules de sortie analogique	4/5
----------------------------------	-----

4.3-1 Configuration des sorties analogiques	4/5
---	-----

4.3-2 Programmation des sorties analogiques	4/5
---	-----

4.3-3 Temps de réponse des sorties analogiques	4/6
--	-----

4.3-4 Exemple de programmation des sorties analogiques	4/7
--	-----

4.3-5 Caractéristiques des sorties analogiques	4/7
--	-----

5 Horodateur	5/1
---------------------	------------

5.1 Présentation	5/1
------------------	-----

5.2 Programmeur temporel	5/1
--------------------------	-----

5.2-1 Caractéristiques	5/1
------------------------	-----

5.2-2 Horodatage par programme	5/2
--------------------------------	-----

5.3 Consigneur temporel	5/3
-------------------------	-----

5.4 Mise à l'heure de l'horodateur	5/4
------------------------------------	-----

5.4-1 Mise à jour de la date et heure par terminal	5/4
--	-----

5.4-2 Mise à jour de la date et heure par mots système	5/4
--	-----

Chapitre	Page
6 Rôle des bits et des mots système	6/1
6.1 Bits système	6/1
6.1-1 Liste des bits système	6/1
6.1-2 Description détaillée des bits système	6/2
6.2 Mots système	6/7
6.2-1 Liste des mots système	6/7
6.2-2 Description détaillée des mots système	6/9
7 Aide à la programmation	7/1
7.1 Modes de marche	7/1
7.2 Conseils de programmation	7/2
7.3 Réarmement des sorties statiques protégées sur TSX 07...12	7/4
7.4 Conditions de réversibilité	7/6
7.5 Règles de réversibilité	7/6

Cette partie B comporte 2 niveaux d'informations :

- les informations permettant de réaliser des fonctions simples. Dans ce cas, il n'est pas besoin de lire la totalité de la documentation, se reporter uniquement aux paragraphes en grisés.
- les informations permettant de réaliser toutes les fonctions offertes par l'automate TSX Nano dans ce cas se reporter à la totalité du manuel.

1.1 Généralités

le développement d'une application destinée à l'automate TSX Nano peut être réalisé à partir de deux outils de programmation:

- Le terminal de programmation FTX 117 qui propose le langage liste d'instructions (List ou IL), langage booléen qui permet l'écriture de traitements logique et numérique.
- Le logiciel de programmation PL7-07 pour IBM PC ou compatible qui propose le langage List et le langage à contacts (Ladder ou LD), langage graphique permettant la transcription de schémas à relais, à l'aide de symboles (contacts, bobines), l'écriture de calculs numériques, est possible à l'intérieur de blocs opérations.

Le PL707 permet la réversibilité des langages: passage du langage Ladder au langage List et inversement.

Le TSX Nano supporte en outre les instructions GRAFCET.


1.2 Langage Liste d'instructions (List ou IL)

Structure d'un programme

Un programme en langage List comprend une suite d'instructions (jusqu'à 1000 instructions) de différentes familles.

Chaque ligne de programme comporte un numéro généré automatiquement, un code instruction et un opérande de type bit ou mot.

Exemple d'instruction :

003	LD	%I0.1
Numéro		Opérande
	Code instruction	

Le programme en langage List est une série d'expressions logiques écrites sous la forme d'une séquence d'instructions booléennes. Chaque instruction booléenne, à l'exception de LOAD, STORE et NOT, fonctionne sur deux opérandes (L'un explicite, l'autre implicite)

L'opérande implicite constitue l'accumulateur booléen et contient soit le contenu de la première instruction d'une séquence d'instructions (ex:LD %I0,0) soit, pour les instructions suivantes, le résultat de l'instruction précédente (ex: AND %I1,2),

Exemple :

001	LD	%i0,1
002	AND	%I1.2
003	ST	%Q1,0

L'opération AND %I1.2, exécute un ET logique entre le contenu de l'accumulateur (%I0,1) et l'entrée %I1.2, et remplace le contenu de l'accumulateur par ce résultat.

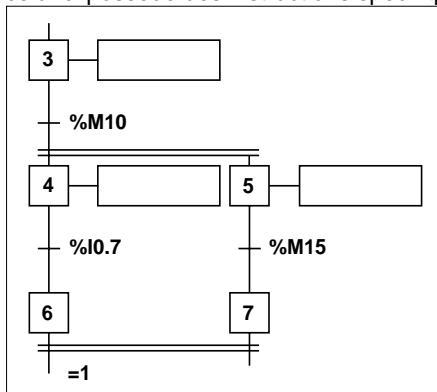
Les instructions LOAD et STORE chargent respectivement l'accumulateur avec la valeur de l'opérande, ou bien stockent l'accumulateur dans l'opérande. L'instruction NOT ne possède pas d'opérande explicite, elle inverse simplement l'état de l'accumulateur.

Famille d'instructions		
Type d'instructions	Exemple	
• Instructions sur bit	004 LD %M10	Lit le bit interne %M10
• Instructions sur bloc	008 IN %TM0	Lance la temporisation %TM0
• Instructions sur mot	010 [%MW10 := %MW50+100]	Addition
• Instructions sur programme	015 SR5	Appel au sous-programme n°5
• Instructions Grafcet	020 -*8	Etape n°8

1.3 Grafcet

Le Grafcet est une méthode d'analyse consistant à décomposer tout automatisme séquentiel en une succession d'étapes, auxquelles sont associées des actions, des transitions et des réceptivités.

Le Grafcet graphique n'étant pas supporté par le logiciel de programmation PL7-07, celui-ci possède des instructions spécifiques Grafcet.

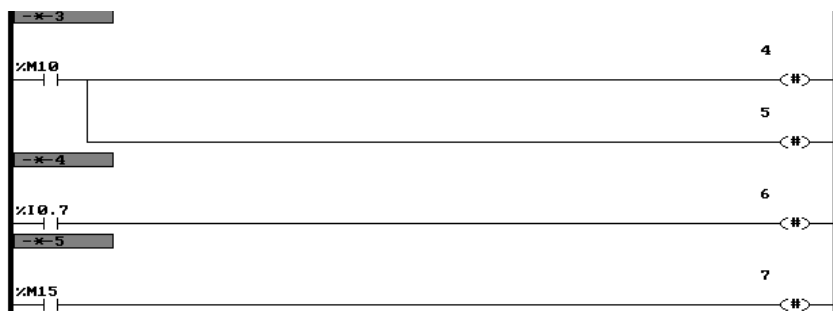


Grafcet graphique

```

-*. 3
LD  %M10
#   4
#   5
-*. 4
LD  %I0.7
#   6
-*. 5
LD  %M15
#   7
...
  
```

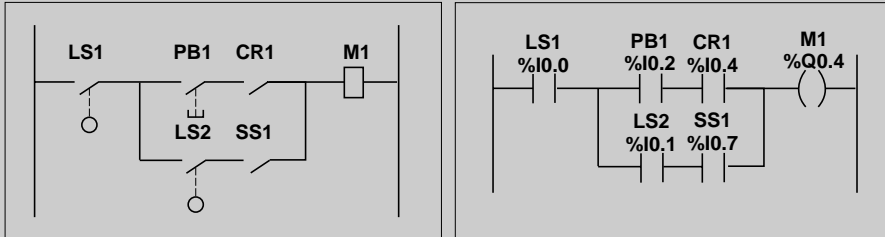
Grafcet Liste d'instructions



Grafcet langage à contacts

1.4 Langage à contacts (Ladder ou LD)

Un programme écrit en langage à contacts se compose d'une suite de réseaux exécutés séquentiellement par l'automate. La représentation d'un réseau est proche de celle d'un schéma électrique à relais. Des éléments graphiques de tests symbolisent les contacts (poussoirs, contacts fin de course, etc...), de même, des éléments graphiques d'actions symbolisent les bobines.



La figure ci-dessus illustre le schéma de branchement simplifié d'un circuit logique à relais et son équivalent représenté en langage à contacts. Sur ce dernier, il convient de noter que toutes les entrées associées à un dispositif de commutation sur le logigramme à relais sont représentées par des symboles contacts, la bobine M1 représentée par un symbole bobine. Les repères qui apparaissent au-dessus de chaque symbole contact/bobine indiquent l'emplacement des connexions d'entrée/sortie externes sur l'automate.

Un réseau de contacts se compose d'une série d'instructions graphiques spécifiques, liées entre elles, et situées entre les deux barres verticales externes représentant le potentiel.

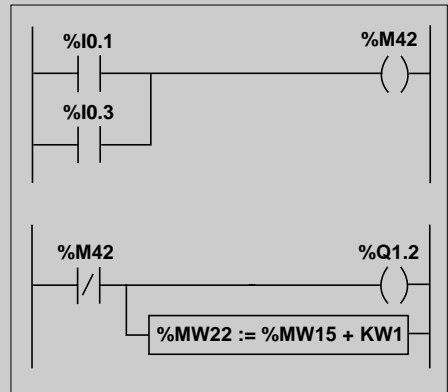
Le jeu d'instructions graphiques représente :

- les entrées/sorties de l'automate (boutons-poussoirs, capteurs, relais, témoins de fonctionnement,...)
- les fonctions de l'automate (temporisateurs, compteurs...),
- les opérations mathématiques et logiques (addition, division, et, xor...),
- les opérateurs comparaison et autres opérations numériques ($A < B$, $A = B$, décalage, rotation...),
- les variables internes de l'automate (bits, mots...).

Ces instructions graphiques sont liées entre elles par des connexions horizontales et verticales qui aboutissent à une ou plusieurs sorties et/ou actions.

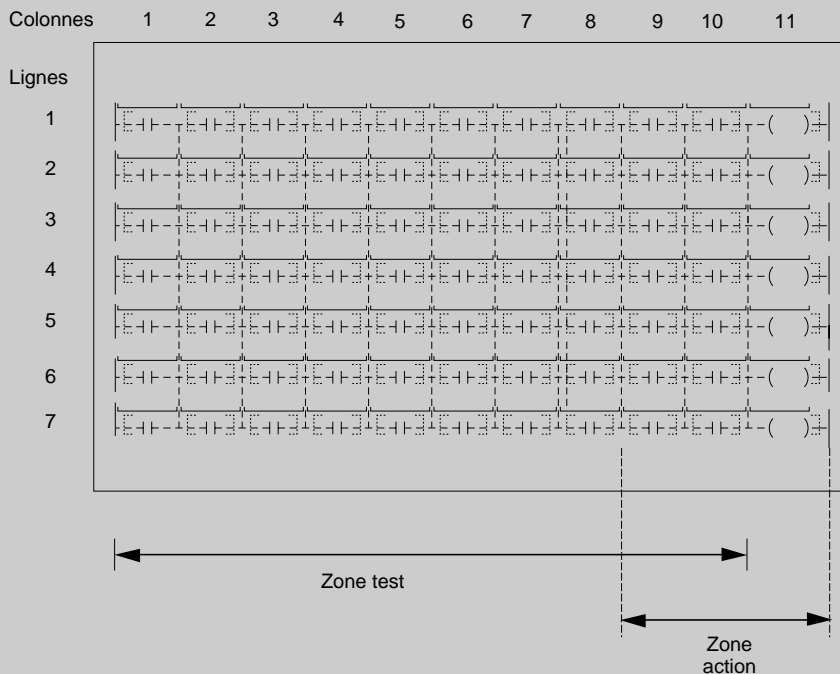
Un réseau ne doit pas supporter plus d'un groupe d'instructions liées.

Ainsi, le programme ci-contre est constitué de deux réseaux distincts.



1.4-1 Principes de programmation

Chaque réseau de contacts est composé de 7 lignes et de 11 colonnes et se partage en deux zones :



- une zone de test qui contient les conditions devant être remplies pour l'exécution d'une action,
- une zone d'action qui contient la sortie ou opération résultant des tests qui y sont associés.

Le réseau est visualisé par une grille de programmation constituée de 7 lignes sur 11 colonnes et commençant par la cellule supérieure gauche de la grille. Les instructions, comparaisons et fonctions relatives aux tests sont écrites dans la zone de test. Ces instructions sont ensuite justifiées à gauche, elles constituent ainsi la continuité de la zone d'action dans laquelle sont écrites les instructions relatives aux bobines, aux opérations numériques et à la gestion du programme. Ces instructions sont justifiées à droite. Le réseau est résolu ou exécuté (les tests sont effectués et les sorties affectées) du haut vers le bas et de la gauche vers la droite.

En outre, un en-tête de réseau apparaît immédiatement au-dessus de celui-ci. Cet en-tête permet de spécifier la finalité logique du réseau. Il comporte le numéro du réseau, toutes les étiquettes (%Li) ou les déclarations de sous-programmes (SRI), le nom du réseau, ainsi que des observations relatives au réseau. Pour toute information complémentaire sur l'en-tête de réseau et sur sa correspondance avec les commentaires de ligne de type List, se reporter au chapitre B.1.4-2..

• Contacts, bobines et instructions relatives au déroulement du programme

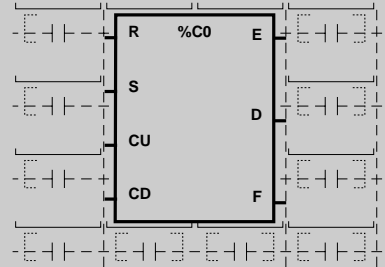
Les instructions relatives aux contacts, bobines et déroulement du programme (saut et appel) occupent une seule cellule de la grille de programmation. Les blocs fonctions, comparaisons et opérations occupent des cellules multiples.



• Blocs fonctions

Les blocs fonctions sont placés dans la zone de test de la grille de programmation. Le bloc doit apparaître sur la première ligne, aucune instruction ni ligne de continuité langage à contacts ne doit apparaître au-dessus ni au-dessous du bloc fonctions. Les instructions relatives au test langage à contacts conduisent vers l'entrée du bloc fonctions et les instructions relatives aux tests et/ou aux actions partent de la sortie du bloc.

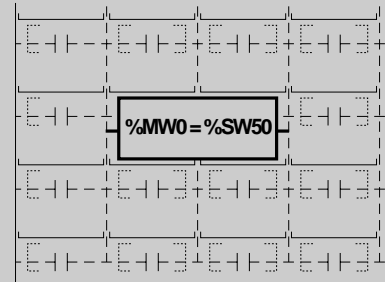
Les blocs fonctions sont disposés verticalement et occupent 2 colonnes sur 4 lignes de la grille de programmation.



• Blocs comparaisons

Les blocs comparaisons sont placés dans la zone de test de la grille de programmation. Le bloc peut apparaître sur n'importe quelle ligne ou colonne de la zone de test tant que la totalité de l'instruction se trouve dans la zone de test.

Les blocs comparaisons sont disposés horizontalement et occupent 2 colonnes sur 1 ligne de la grille de programmation.



• Blocs opérations

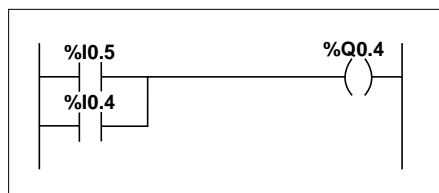
Les blocs opérations sont placés dans la zone d'action de la grille de programmation. Le bloc peut apparaître sur n'importe quelle ligne de la zone d'action. L'instruction est justifiée à droite. Par conséquent, elle apparaît sur la droite et finit dans la dernière colonne.

Les blocs opérations sont disposés horizontalement et occupent 4 colonnes sur 1 ligne de la grille de programmation.



1.4-2 Réversibilité

Dans ce manuel, le terme "réversibilité" fait référence à la capacité du logiciel de programmation PL7-07 du TSX Nano à transcrire les programmes d'application du TSX Nano, du langage Ladder en langage List et inversement. Les programmes PL7-07 peuvent être affichés dans le format désiré. Il suffit pour cela de définir sa préférence pour l'un ou l'autre. Le logiciel PL7-07 permet aussi la conversion d'un réseau Ladder individuel en langage List, et le restaurer ensuite en sélectionnant simplement l'option Ladder/List du menu Outils de l'éditeur Ladder/List.



LD	%I0.5
OR	%I0.4
ST	%Q0.4

La compréhension de la notion de réversibilité repose sur la relation entre le "réseau" (ensemble des instructions de programmation langage à contacts qui constituent une expression logique) et la "phrase" (ensemble des instructions de programmation langage liste d'instructions qui accomplissent la même fonction). La figure ci-dessus illustre l'exemple d'un réseau courant exprimé dans le langage à contacts d'un programme utilisateur quelconque. A côté de cette figure est représentée la logique équivalente exprimée sous forme de phrase en langage List.

Une application écrite en langage PL7-07 (Ladder ou List) est stockée en mémoire en langage List.

La structure du langage Ladder du logiciel PL7-07 permet la conversion d'un programme Ladder en programme List sans avoir pris au préalable de disposition particulière.

La conversion d'une application développée en List nécessite le respect d'une série de conventions de réversibilité décrites dans le chapitre B.1.4-3. Cependant, la non-réversibilité d'un programme List n'affecte en rien le fonctionnement d'une application.

1.4-3 Conventions de programmation en langage List réversible

La structure d'un bloc fonction réversible en langage List nécessite l'emploi de certaines instructions spécifiques. Il s'agit de:

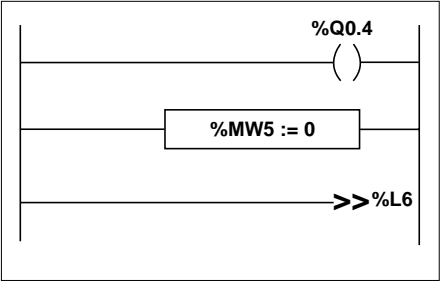
- BLK qui marque le début du bloc et définit le commencement du réseau et le début de la portion d'entrée dans le bloc
- OUT_BLK (marque le début de la portion de sortie du bloc
- END_BLK (marque bien entendu la fin du bloc et du réseau).

L'utilisation des instructions réversibles relatives au bloc fonctions n'est pas indispensable au bon fonctionnement du programme langage List. Pour certaines instructions, il est possible de programmer en langage List non réversible. La programmation langage List non réversible des blocs fonctions est décrite au chapitre B.2.2 .

Une autre convention importante est d'éviter l'utilisation de certaines instructions langage List, ou de certaines combinaisons d'instructions et opérandes qui n'ont pas d'équivalence en langage à contacts. L'instruction N (inverse le contenu de l'accumulateur), par exemple, n'a pas d'équivalent en langage à contacts. Le tableau ci-dessous liste toutes les instructions de programmation langage List non convertibles en langage à contacts.

Instruction langage liste	Opérande	Description
JMPCN	%Li	Saut conditionnel Not
N	aucun	Négation (NOT)
ENDCN	aucun	Fin conditionnelle Not
or'd XORN	quel qu'il soit	XORN précédé de OU logique

Les réseaux inconditionnels suivent également une convention de programmation langage List afin d'assurer la réversibilité langage Ladder/List. Un réseau inconditionnel est un réseau dans lequel il n'y a ni test, ni condition; la (ou les) instruction(s) relative(s) à la sortie et/ou l'action est (sont) excitée(s), ou exécutée(s), en permanence. La figure ci-dessous représente des réseaux inconditionnels et les phrases équivalentes en langage List.



```
LD 1
ST %Q0.4
LD 1
[%MW5 := 0]
JMP %L6
```

Il convient de noter que chacune des phrases inconditionnelles, à l'exception d'une seule, commence en langage List par l'instruction LD (Load) suivie du chiffre 1. La combinaison affecte la valeur 1 à l'accumulateur booléen, et par conséquent, la valeur 1 à la bobine (instruction de mémorisation), et la valeur 0 à %MW5 à chaque cycle de scrutation du programme. L'instruction de saut inconditionnel constitue une exception. L'instruction langage List est exécutée quelle que soit la valeur de l'accumulateur et ne nécessite par conséquent pas de mise à 1 de l'accumulateur, à l'inverse des deux exemples précédents.

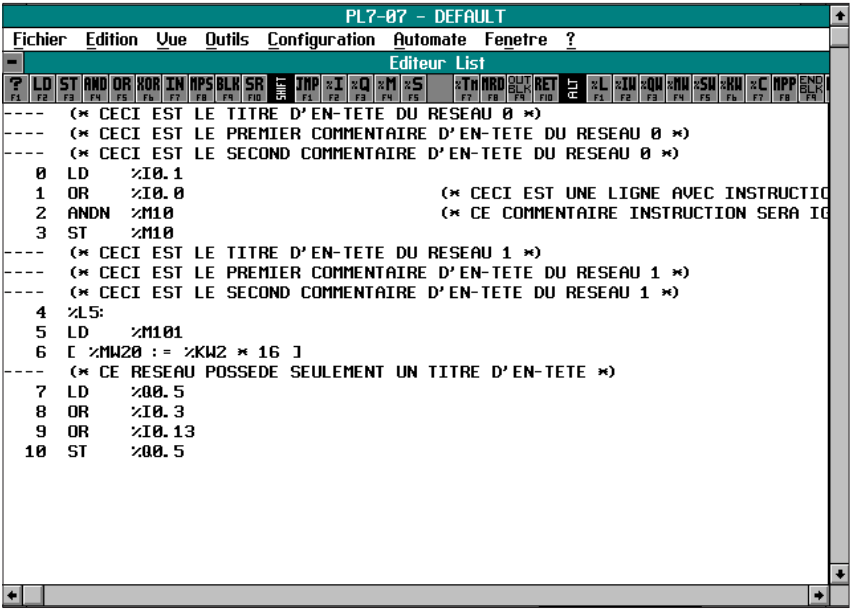
Il est possible de convertir un programme List qui n'est pas complètement réversible, les portions réversibles sont affichées en Ladder, et les portions non réversibles, restent en List . Le programme ainsi convertit conserve l'ordre d'écriture initial. Les "réseaux" d'instruction List non-réversibles peuvent être visualisés et modifiés à l'aide de l'éditeur de réseau List obtenu en cliquant deux fois sur le réseau concerné.

Description du programme

L'éditeur List permet d'insérer dans le programme des lignes de commentaires. Ces commentaires peuvent apparaître isolés ou sur la même ligne que les instructions de programmation.

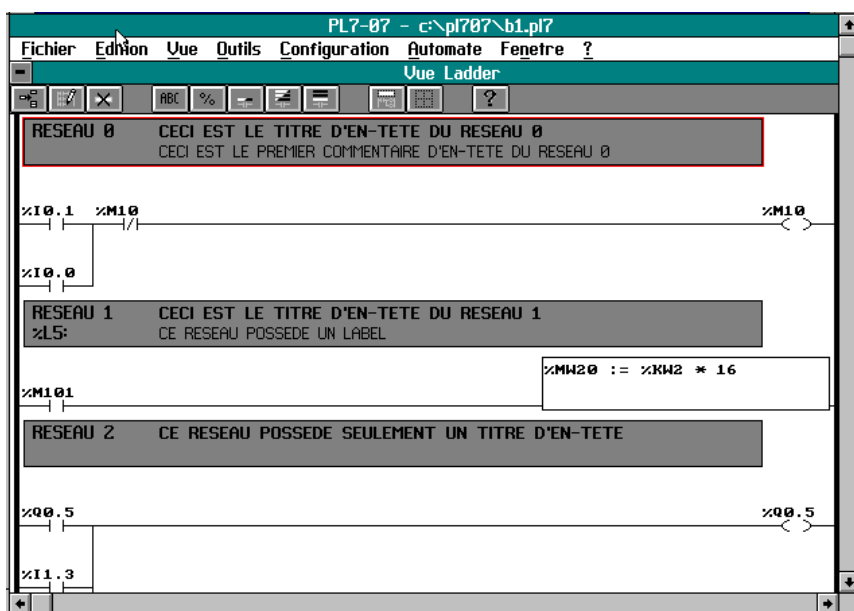
L'Editeur Ladder vous permet de documenter votre programme à l'aide d'en-têtes réseaux placés juste au-dessus de ces derniers.

Le logiciel PL7-07 prend en compte ces commentaires lors de la conversion. Lorsqu'il convertit un programme List en Ladder, le PL7-07 utilise les lignes de commentaires isolés, situées au dessus des phrases List pour élaborer les en-têtes réseaux correspondants.



La première ligne de commentaires isolés est utilisée pour documenter la première ligne d'en-tête du réseau Ladder correspondant. L'en-tête d'un réseau est ainsi documenté en utilisant les lignes de commentaires isolés dans l'ordre d'écriture. Lorsque les lignes d'en-tête sont remplies, le reste des lignes de commentaires isolés est ignoré, de même que tous les commentaires qui se trouvent sur les lignes de d'instructions List.

Lorsqu'un réseau Ladder contenant un en-tête réseau est converti en langage List, la description de l'en-tête réseau est insérée entre les phrases List. Toute étiquette ou déclaration de sous-programme (%Li ou SRI) est placée sur la ligne qui suit l'en-tête, juste avant le début de la phrase List. Si le réseau qui est converti était écrit à l'origine en List, et que des commentaires ont été ignorés lors de la conversion de List en Ladder, ces commentaires réapparaissent dans l'éditeur List.



2.1 Traitement booléen

2.1-1 Définition des principaux objets bits

- **Bits d'entrées/sorties**

L'adressage de ces bits est défini de manière détaillée chapitre 1.5 intercalaire A. Ces bits sont les "images logiques" des états électriques des entrées/sorties. Ils sont rangés dans la mémoire de données et sont mis à jour à chaque scrutation du programme.

- **Bits internes**

Les bits internes mémorisent des états intermédiaires durant l'exécution du programme.

Note : Les bits d'entrées/sorties non utilisés ne peuvent pas être employés comme bits internes.

- **Bits système**

Les bits système %S0 à %S127 surveillent le bon fonctionnement de l'automate ainsi que le déroulement du programme application. Le rôle et l'utilisation de ces bits sont développés dans le chapitre 5 du présent intercalaire.

- **Bits étapes**

Les bits %X1 à %X62 sont les bits associés aux étapes Grafcet. Le bit étape Xi est à 1 lorsque l'étape correspondante est active et est à 0 lorsque cette étape est inactive.

- **Bits extraits de mots : voir ch3.1-1**

Liste des opérandes bits

Le tableau suivant donne la liste de tous les types d'opérandes bits

Type	Adresse (ou valeur)	Nombre maximum	Accès en écriture (1)	Voir Ch.
Valeur immédiate	0 ou 1	-	-	-
Bits d'entrées de sorties	%I0.i ou %I1.i (2) %Q0.i ou %Q1.i (2)	28 20	non oui	1.5 Inter A
Bits internes	%Mi	128 (3)	oui	
Bits système	%Si	128	selon i	5.1
Bits d'étape Grafcet	%Xi	62	oui	2.3-1
Bits de blocs fonction	%Tmi.Q %DRi.F.....		non (4)	2.2-1
Bits de blocs fonction réversible	E,D,F,Q,TH0,TH1		non	3.3-1
Bits extraits de mots				3.1-1

(1) écriture par programme ou en mode réglage par terminal.

(2) avec i = 0 pour un automate de base ou une extension automate, i = 1 pour une extension d'E/S, et j = n° de la voie. Les bits d'entrées/sorties peuvent être forcés à 0 ou 1 en mode réglage des données.

(3) les 64 premiers sont sauvegardés sur coupure secteur.

(4) excepté %SBri.j et %Sci.j ces bits peuvent être lus et écrits.

2.1-2 Présentation des instructions booléennes

Les instructions booléennes peuvent être comparées à des éléments de langage à contacts.

Éléments de test, exemple : l'instruction LD est équivalent à un contact ouvert.



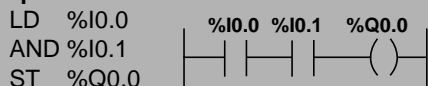
Contact passant quand l'objet bit qui le pilote est à l'état 1.

Éléments d'action, exemple : l'instruction ST est équivalente à une bobine directe.



L'objet bit associé prend la valeur logique du résultat logique de l'élément de test.

Equation booléenne :

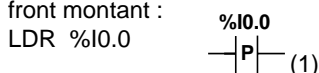


Le résultat booléen des éléments de test, est appliqué à l'élément d'action.

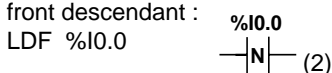
Fronts montant et descendant

Les instructions de test permettent de détecter les fronts montant ou descendant sur les entrées automate. Un front est détecté lorsque l'état d'une entrée a changé entre le cycle n-1 et le cycle n en cours, il reste détecté durant le cycle en cours.

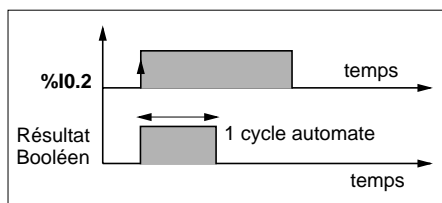
L'instruction LDR (R: Rising edge) est équivalent à un contact à détection de front montant :



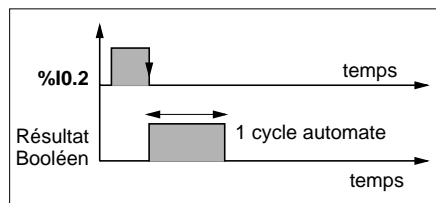
L'instruction LDF (F: Falling edge) est équivalent à un contact à détection de front descendant :



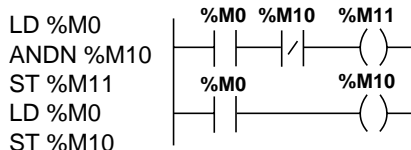
Front montant : détection du passage de 0 à 1 de l'entrée qui le pilote (3).



Front descendant : détection du passage de 1 à 0 de l'entrée qui le pilote.



Les instructions sur front portent sur les entrées %I, mais il est possible de détecter des fronts sur tout autre bit (ou résultat booléen) en utilisant 2 bits internes. Dans l'exemple ci-contre le bit %M11 mémorise le front montant du bit %M0.



(1) Contact à détection de transition Positive

(2) Contact à détection de transition Négative

(3) **Sur démarrage à froid et à chaud, l'application détecte un front montant même si l'entrée est restée à 1. Il est possible de masquer ce phénomène en débutant un programme par les instructions LD %S1 et ENDC.**

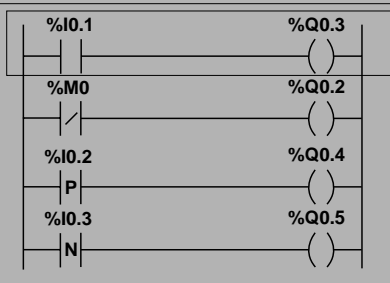
Description des instructions

La description des instructions booléennes est réalisée de la façon suivante :

L'instruction booléenne décrite est présentée en gras. Chaque équation est illustrée par le schéma à contacts correspondant.

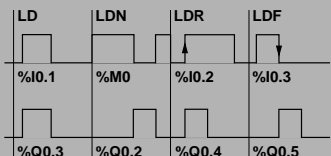
Instructions de chargement LD, LDN, LDR, LDF

Les instructions LD, LDN, LDR et LDF correspondent respectivement aux contacts ouvert, fermé, à front montant et à front descendant.



LD %I0.1
ST %Q0.3
LDN %M0
ST %Q0.2
LDR %I0.2
ST %Q0.4
LDF %I0.3
ST %Q0.5

Code	Opérande
LD	0/1, %I, %Q, %M, %S, %X, %BLK.x, %*:Xk, [
LDN	%I, %Q, %M, %S, %X, %BLK.x, %*:Xk, [
LDR	%I
LDF	%I

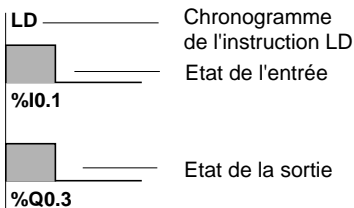


Liste des opérandes

0/1 valeur immédiate 0 ou 1
%I entrée automate %Ii.j
%Q sortie automate %Qi.j
%M bit interne %Mi
%S bit système %Si
%X bit étape %Xi
%BLK.x bit de bloc fonction, ex : %TMi.Q
%*:Xk bit de mot, ex : %MWi:Xk
[Expression de comparaison
ex : [%MWi<1000]

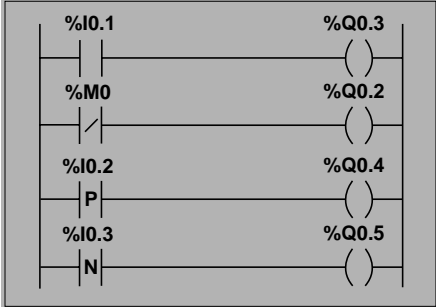
Chronogramme

Les 4 chronogrammes ont été regroupés ensemble.



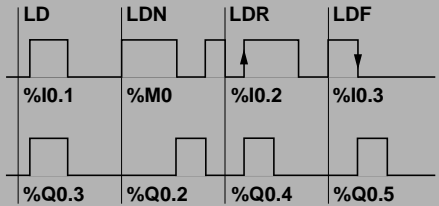
2.1-3 Instructions de chargement LD, LDN, LDR, LDF

Les instructions LD, LDN, LDR et LDF correspondent respectivement aux contacts ouvert, fermé, à front montant et à front descendant (LDR et LDF uniquement sur entrées automate).



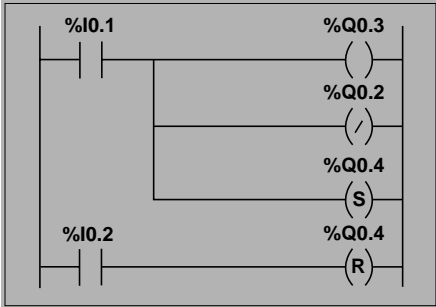
LD	%I0.1
ST	%Q0.3
LDN	%M0
ST	%Q0.2
LDR	%I0.2
ST	%Q0.4
LDF	%I0.3
ST	%Q0.5

Code	Opérande
LD	0/1, %I, %Q, %M, %S, %X, %BLK.x, %•:Xk, [
LDN	%I, %Q, %M, %S, %X, %BLK.x, %•:Xk, [
LDR	%I
LDF	%I



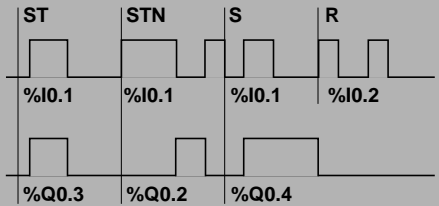
2.1-4 Instructions d'affectation ST, STN, S, R

Les instructions ST, STN, S et R correspondent respectivement aux bobines directe, inverse, à enclenchement et à déclenchement.



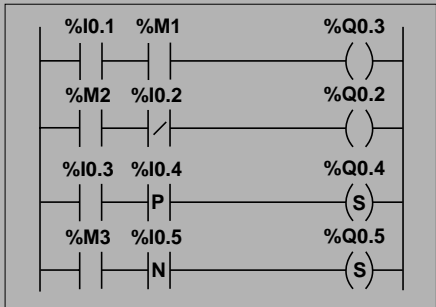
LD	%I0.1
ST	%Q0.3
STN	%Q0.2
S	%Q0.4
LD	%I0.2
R	%Q0.4

Code	Opérande
ST	%Q, %M, %S, %BLK.x, %•:Xk
STN	%Q, %M, %S, %BLK.x, %•:Xk
S	%Q, %M, %S, %X, %BLK.x, %•:Xk
R	%Q, %M, %S, %X, %BLK.x, %•:Xk



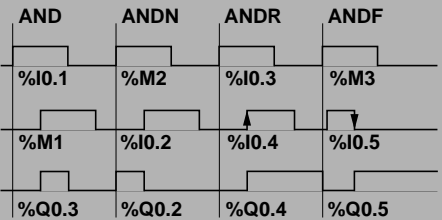
2.1-5 Instructions ET Logique : AND, ANDN, ANDR, ANDF

Ces instructions effectuent un ET logique entre l'opérande (ou son inverse, ou son front montant, ou son front descendant) et le résultat booléen de l'instruction précédente.



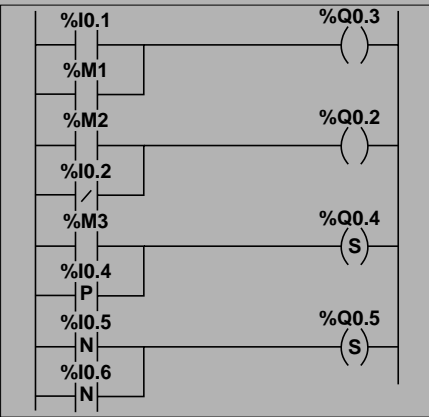
```
LD    %I0.1
AND   %M1
ST    %Q0.3
LD    %M2
ANDN  %I0.2
ST    %Q0.2
LD    %I0.3
ANDR  %I0.4
S     %Q0.4
LD    %M3
ANDF  %I0.5
R     %Q0.5
```

Code	Opérande
AND	0/1, %I, %Q, %M, %S, %X, %BLK.x, %*:Xk
ANDN	%I, %Q, %M, %S, %X, %BLK.x, %*:Xk
ANDR	%I
ANDF	%I



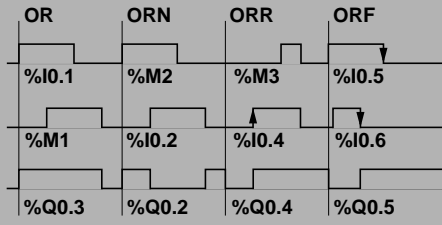
2.1-6 Instructions OU Logique : OR, ORN, ORR, ORF

Ces instructions effectuent un OU entre l'opérande (ou son inverse, ou son front montant, ou son front descendant) et le résultat booléen de l'instruction précédente.



```
LD    %I0.1
OR    %M1
ST    %Q0.3
LD    %M2
ORN   %I0.2
ST    %Q0.2
LD    %M3
ORR   %I0.4
S     %Q0.4
LDF   %I0.5
ORF   %I0.6
R     %Q0.5
```

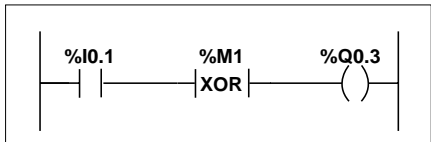
Code	Opérande
OR	0/1, %I, %Q, %M, %S, %X, %BLK.x, %*:Xk
ORN	%I, %Q, %M, %S, %X, %BLK.x, %*:Xk
ORR	%I
ORF	%I



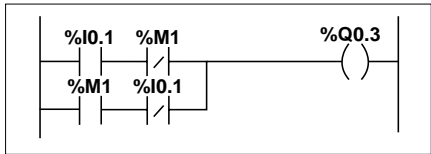
B

2.1-7 Instructions OU Exclusif : XOR, XORN, XORR, XORF

Ces instructions effectuent un OU exclusif entre l'opérande (ou son inverse, ou son front montant, ou son front descendant) et le résultat booléen de l'instruction précédente.

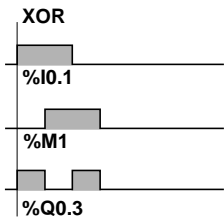


```
LD    %I0.1
XOR   %M1
ST    %Q0.3
```



```
LD    %I0.1
ANDN  %M1
OR(    %M1
ANDN  %I0.1
)
ST    %Q0.3
```

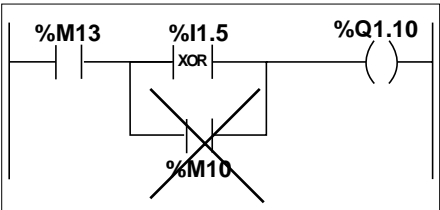
Code	Opérande
XOR	%I,%Q,%M,%S,%X,%BLK.x,%*:Xk
XORN	%I,%Q,%M,%S,%X,%BLK.x,%*:Xk
XORR	%I
XORF	%I



• Cas spécifique

En langage à contacts, l'instruction XOR ne doit pas :

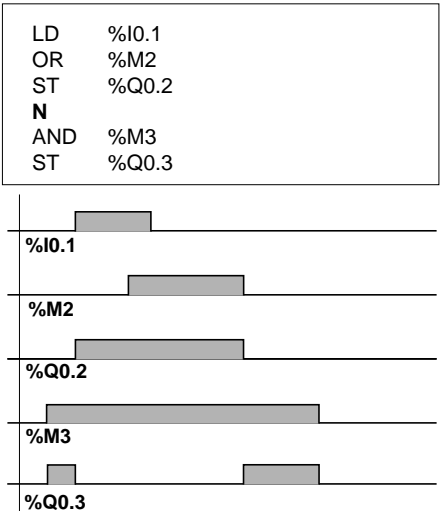
- se situer à gauche du réseau de contacts (première position),
- être mise en parallèle.



2.1-8 Instruction Négation : N

Cette instruction effectue la négation du résultat booléen de l'instruction précédente.

Code	Opérande
N	-

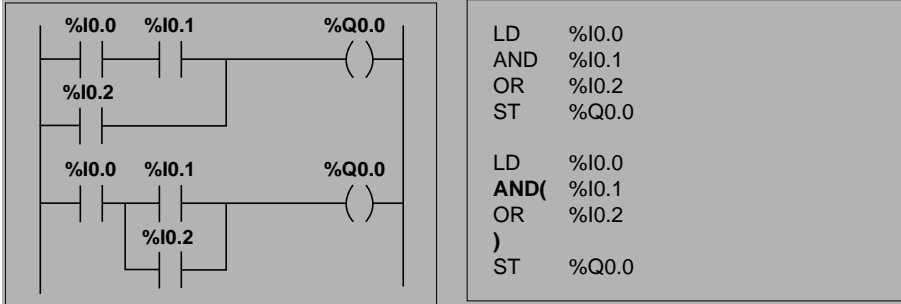


Note : l'instruction N n'est pas réversible.

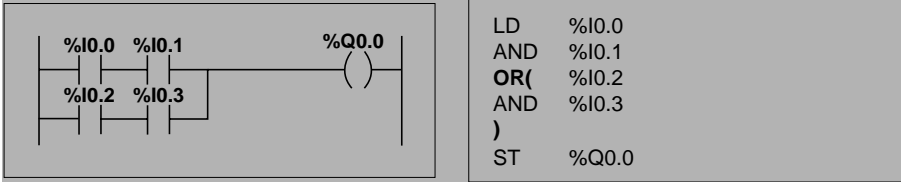
2.1-9 Utilisation des parenthèses

Les instructions AND et OR peuvent utiliser des parenthèses. Ces parenthèses permettent de réaliser des schémas à contacts de façon simple. L'ouverture de parenthèse est associée à l'instruction AND ou OR. La fermeture de parenthèse est une instruction, elle est obligatoire pour chaque parenthèse ouverte.

Exemple : AND(...)

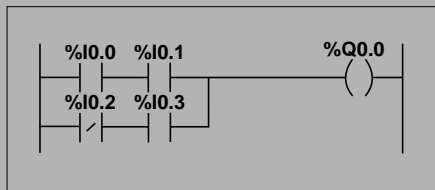


Exemple : OR(...)



Aux parenthèses peuvent être associées les modificateurs N, F, R ou [:

- N négation, ex : AND(N ou OR(N
- F front montant, ex : AND(F ou OR(F
- R front descendant, ex : AND(R ou OR(R
- [comparaison, voir ch 3.1-5

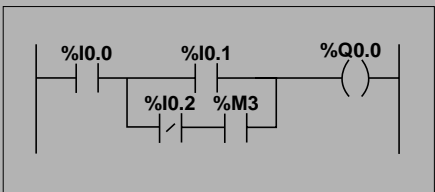


```
LD    %I0.0
AND   %I0.1
OR(N %I0.2
AND   %I0.3
)
ST    %Q0.0
```

Imbrication de parenthèses

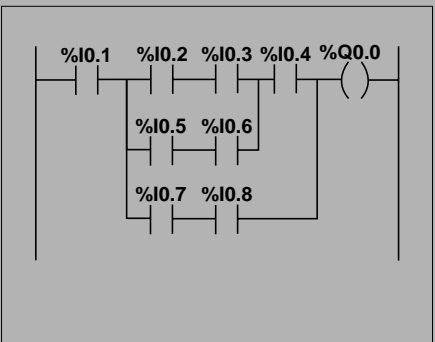
Il est possible d'imbriquer jusqu'à 8 niveaux de parenthèses.

Exemple



```
LD    %I0.0
AND( %I0.1
OR(N %I0.2
AND   %M3
)
)
ST    %Q0.0
```

Exemple



```
LD    %I0.1
AND( %I0.2
AND   %I0.3
OR( %I0.5
AND   %I0.6
)
AND   %I0.4
OR( %I0.7
AND   %I0.8
)
)
ST    %Q0.0
```

Notes :

- chaque parenthèse ouverte doit être impérativement refermée.
- les étiquettes %Li: et les sous programmes %SRi: ne doivent pas être placées dans des expressions entre parenthèses, ainsi que les instructions de saut JMP et d'appel à sous-programme SRi,
- les instructions d'affectation ST, STN, S et R ne doivent pas être programmées entre parenthèses.

2.1-10 Instructions MPS, MRD, MPP

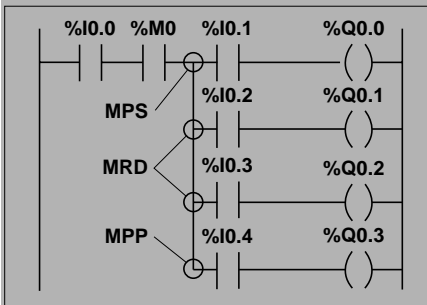
Les 3 types d'instruction permettent de traiter les aiguillages vers les bobines. Ces instructions utilisent une mémoire intermédiaire appelée pile pouvant stocker jusqu'à 8 informations booléennes.

L'instruction MPS a pour effet de stocker le résultat de la dernière instruction de test au sommet de la pile et de décaler les autres valeurs vers le fond de la pile.

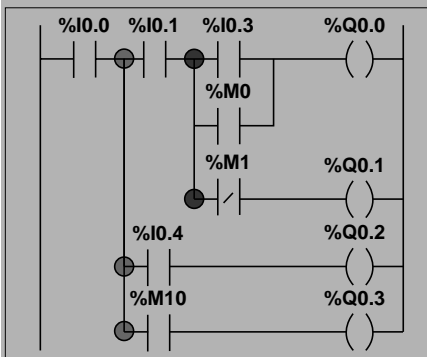
L'instruction MRD lit le sommet de la pile.

L'instruction MPP a pour effet de lire, de déstocker le sommet de la pile et de décaler les autres valeurs vers le sommet de la pile.

Exemples :



```
LD    %I0.0
AND   %M0
MPS
AND   %I0.1
ST    %Q0.0
MRD
AND   %I0.2
ST    %Q0.1
MRD
AND   %I0.3
ST    %Q0.2
MPP
AND   %I0.4
ST    %Q0.3
```



```
LD    %I0.0
MPS
AND   %I0.1
MPS
AND(  %I0.3
OR    %M0
)
ST    %Q0.0
MPP
ANDN  %M1
ST    %Q0.1
MRD
AND   %I0.4
ST    %Q0.2
MPP
AND   %M10
ST    %Q0.3
```

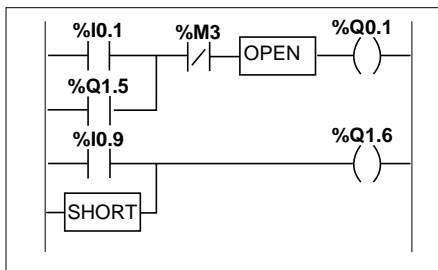
Note : ces instructions ne peuvent pas être utilisées au sein d'une expression entre parenthèses.

2.1-11 Instructions OPEN et SHORT spécifiques au langage à contacts

Deux instructions spécifiques au langage à contacts peuvent être utilisées afin d'aider la mise au point d'un programme.

L'instruction OPEN permet de couper la continuité (logique 0) quelque soit le résultat de la précédente opération logique (équivalent langage List : instruction AND 0).

L'instruction SHORT permet d'assurer la continuité (logique 1) quelque soit le résultat de la précédente opération logique (équivalent langage List : OR 1).



```
LD    %I0.1
OR    %Q1.5
ANDN  %M3
AND   0
ST    %Q0.1
LD    %I0.9
OR    1
ST    %Q1.6
```

2.2 Blocs fonction standards

2.2-1 Objets bits et mots associés aux blocs fonction standards

Les blocs fonction mettent en œuvre des objets bits et des mots spécifiques.

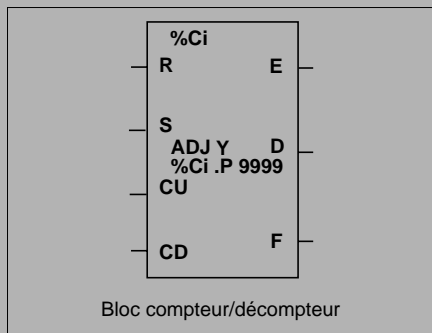
- **objets bits :**

Ils correspondent aux sorties des blocs.
Ces bits sont accessibles par les instructions booléennes de test.

Elles sont adressables :

- soit directement (ex : LD E) si elles sont câblées au bloc en programmation réversible (voir ch 2.2-2),
- soit en spécifiant le type de bloc (ex : LD %Ci.E),

Les entrées sont accessibles sous forme d'instructions.



- **objets mots :**

Ils correspondent :

- aux paramètres de configuration du bloc, ces paramètres peuvent être accessibles (ex : paramètre de présélection) ou pas (ex: base de temps) par programme,
- aux valeurs courantes (ex : %Ci.V valeur de comptage en cours).

Liste des objet bits et mots de blocs fonction accessibles par programme

Blocs fonction standards	Mots et bits associés		Adresses	Accès écriture	Voir Ch.
Temporisateur %Tmi (i=0 à 31)	Mot	Valeur courante	%Tmi.V	non	2.2-3
		Valeur de présélection	%Tmi.P	oui	
	Bit	Sortie temporisateur	%Tmi.Q	non	
Compteur/ décompteur %Ci (i=0 à 15)	Mot	Valeur courante	%Ci.V	non	2.2-4
		Valeur de présélection	%Ci.P	oui	
	Bit	Sortie débordement (vide)	%Ci.E	non	
		Sortie présélection atteinte	%Ci.D	non	
		Sortie débordement (plein)	%Ci.F	non	
Registre mot %Ri (i= 0 à 3)	Mot	Accès au registre	%Ri.I	oui	2.2-5
		Sortie du registre	%Ri.O	oui	
	Bit	Sortie registre plein	%Ri.F	non	
		Sortie registre vide	%Ri.E	non	
Programmateur cyclique %DRi (i=0 à 3)	Mot	Numéro de pas en cours	%DRi.S	oui	2.2-6
	Bit	Dernier pas défini en cours	%DRi.F	non	

2.2-2 Principes de programmation

- Les blocs fonction standards peuvent être programmés de 2 façons différentes :
- avec instructions de bloc fonction (ex : BLK %TM2), cette façon réversible en langage à contacts autorise d'effectuer les opérations sur le bloc en un seul endroit du programme,
 - avec instructions spécifiques (ex : CU %Ci), cette façon non réversible permet d'effectuer des opérations sur les entrées du bloc en divers endroits du programme (ex: ligne 100 CU %C1, ligne 174 CD %C1, ligne 209 LD %C1.D).

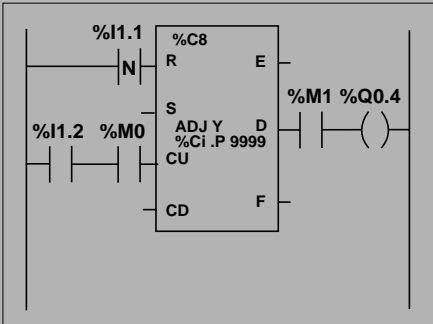
Principes de programmation réversible des blocs fonction standards

Ce type de programmation utilise les instructions de bloc BLK, OUT_BLK et END_BLK. **BLK** indique le début du bloc fonction.

OUT_BLK optionnel, permet de "câbler" directement les sorties du bloc.

END_BLK indique la fin du bloc.

Exemple réversible avec sorties câblées



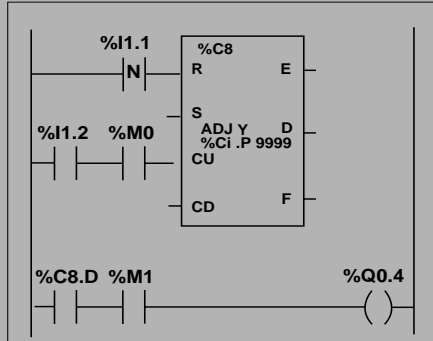
```

BLK %C8
LDF %I1.1
R
LD %I1.2
AND %M0
CU
OUT_BLK
LD D
AND %M1
ST %Q0.4
END_BLK
    
```

Traitement des entrées

Traitement des sorties

Exemple réversible sans câblage des sorties



```

BLK %C8
LDF %I1.1
R
LD %I1.2
AND %M0
CU
END_BLK
.....
LD %C8.D
AND %M1
ST %Q0.4
    
```

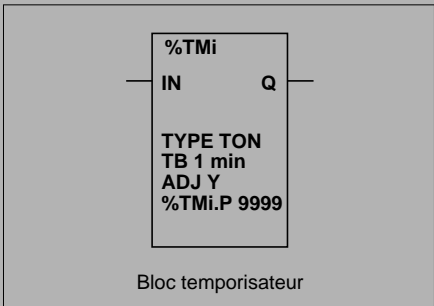
Traitement des entrées

Traitement des sorties

Note :

Seules les instructions de test et d'entrées sur le bloc concerné sont autorisées entre les instructions BLK et OUT_BLK (ou entre BLK et END_BLK, lorsque OUT_BLK n'est pas programmée).

2.2-3 Blocs fonction temporisateur %TMI



3 types de temporisateur sont proposés :

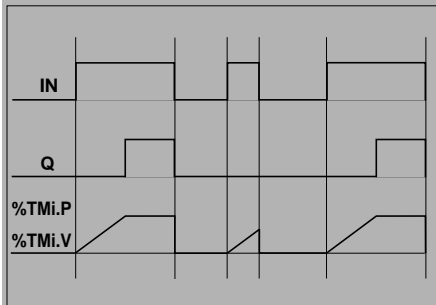
- **TON** : ce type de temporisateur permet de gérer des retards à l'enclenchement. Ce retard est programmable et peut être modifiable ou non par terminal.
- **TOF** : ce type de temporisateur permet de gérer des retards au déclenchement. Ce retard est programmable et peut être modifiable ou non par terminal.
- **TP** : ce type de temporisateur permet d'élaborer une impulsion de durée précise. Cette durée est programmable et peut être modifiable ou non par terminal.

Caractéristiques

Numéro temporisateur	%TMI	0 à 31
Type	TON TOF TP	<ul style="list-style-type: none">• retard à l'enclenchement (par défaut)• retard au déclenchement• monostable
Base de temps	BT	1mn (par défaut), 1s, 100ms, 10ms, 1ms (pour TM0 et TM1). Plus la base de temps est faible, plus la précision du temporisateur sera grande.
Valeur courante	%TMI.V	Mot qui croît de 0 à %TMI.P sur écoulement du temporisateur. Peut être lu, testé, mais non écrit par programme (1).
Valeur de présélection	%TMI.P	$0 \leq \%TMI.P \leq 9999$. Mot qui peut être lu, testé, et écrit par programme. Est mis à la valeur 9999 par défaut. La durée ou retard élaboré est égal à %TMI.P x BT.
Réglage par terminal	O/N	O : possibilité de modification de la valeur de présélection %TMI.P en mode Réglage. N : pas d'accès en mode réglage.
Entrée (ou instruction) "Armement"	IN	Sur front montant (type TON ou TP) ou front descendant (type TOF), démarre le temporisateur.
Sortie "Temporisateur"	Q	Bit associé %TMI.Q, sa mise à 1 dépend de la fonction réalisée TON, TOF ou TP.

(1) %TMI.V peut être modifiée par terminal en mode Réglage.

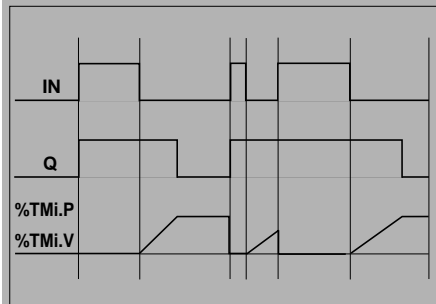
Utilisation en temporisation à retard à l'enclenchement : type TON



Lors d'un front montant sur l'entrée IN (1), le temporisateur est lancé : sa valeur courante %TMI.V croît de 0 vers %TMI.P d'une unité à chaque impulsion de la base de temps BT. Le bit de sortie %TMI.Q passe à 1 dès que la valeur courante a atteint %TMI.P puis reste à 1 tant qu'un front descendant n'est pas détecté sur l'entrée IN.

Lors d'un front descendant sur l'entrée IN (2), le temporisateur est arrêté même si le temporisateur n'a pas atteint sa valeur de présélection %TMI.P.

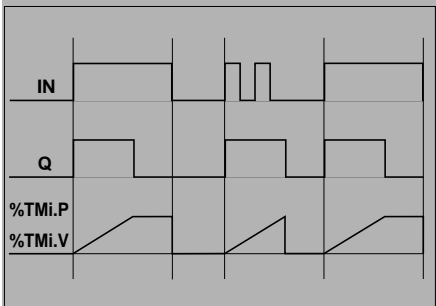
Utilisation en temporisation à retard au déclenchement : type TOF



La valeur courante %TMI.V prend la valeur 0, sur un front montant de l'entrée IN (1) (même si le temporisateur est en cours d'évolution). Lors du front descendant sur l'entrée IN, le temporisateur est lancé.

Puis la valeur courante croît vers %TMI.P d'une unité à chaque impulsion de la base de temps BT. Le bit de sortie %TMI.Q passe à 1 dès qu'un front montant est détecté sur l'entrée IN et retombe à 0 quand la valeur courante a atteint %TMI.P.

Utilisation en monostable : type TP



Lors d'un front montant sur l'entrée IN (1), le temporisateur est lancé. (si le temporisateur n'est pas déjà en cours d'évolution).

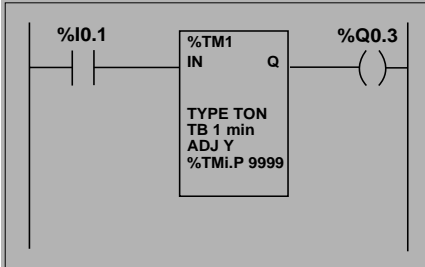
%TMI.V croît de 0 vers %TMI.P d'une unité à chaque impulsion de la base de temps BT. Le bit de sortie %TMI.Q passe à 1 dès que le temporisateur est lancé et retombe à 0 quand la valeur courante a atteint %TMI.P.

Quand la valeur courante %TMI.V a atteint la valeur de présélection %TMI.P, %TMI.V prend la valeur 0 si l'entrée IN est à 0. Ce monostable n'est pas réarmable.

- (1) ou activation de l'instruction IN.
(2) ou désactivation de l'instruction IN.

Programmation et configuration

La programmation des blocs fonction temporisateur est identique quel que soit le type d'utilisation qui en est faite. Le choix du fonctionnement TON, TOF ou TP s'effectue en configuration.

Programmation réversible

```

BLK  %TM1
LD   %I0.1
IN
OUT_BLK
LD   Q
ST   %Q0.3
END_BLK

```

Configuration

Les paramètres suivants sont à saisir en configuration.

- Type : TON, TOF ou TP
- BT : 1mn, 1s, 100ms, 10ms ou 1ms
- %Tmi.P : 0 à 9999
- Réglage : O ou N

Programmation non réversible

```

LD   %I0.1
IN   %TM1
LD   %TM1.Q
ST   %Q0.3

```

Cas spécifiques

- **Incidence d'une reprise "à froid"** : (%S0=1), provoque la mise à 0 de la valeur courante, la mise à 0 de la sortie %Tmi.Q et la valeur de présélection est ré-initialisée à la valeur définie en configuration.
- **Incidence d'une "reprise à chaud"** : (%S1=1) n'a pas d'incidence sur la valeur courante du temporisateur, ni sur la valeur de présélection. La valeur courante n'évolue pas pendant le temps de la coupure secteur.
- **Incidence d'un passage en stop** : le passage en stop de l'automate ne fige pas la valeur courante.
- **Incidence d'un saut de programme** : le fait de ne pas scruter les instructions où est programmé le bloc temporisateur ne fige pas la valeur courante %Tmi.V qui continue à croître vers %Tmi.P. De même le bit %Tmi.Q associé à la sortie Q du bloc temporisateur conserve son fonctionnement normal et peut être ainsi testé par une autre instruction. Par contre la sortie, directement câblée à la sortie du bloc, n'est pas activée puisque non scrutée par l'automate,
- **Test du bit %Tmi.Q** : il est conseillé de tester le bit %Tmi.Q qu'une seule fois dans le programme.
- **Incidence des instructions relais maître MCS/MCR** : un bloc temporisation programmé entre 2 instructions MCS/MCR est réinitialisé lorsque l'instruction MCS est active.
- **Incidence de la modification de la présélection %Tmi.P** : la modification de la valeur de présélection par instruction ou en réglage n'est prise en compte qu'à la prochaine activation du temporisateur.

Temporisateurs à base de temps de 1 ms (TSX 07 3• •••)

La base de temps de 1 ms n'est disponible que sur les temporisateurs %TM0 et %TM1. Si l'utilisateur en a le besoin, il peut utiliser les quatre mots systèmes %SW76, %SW77, %SW78 et %SW79 comme "sabliers".

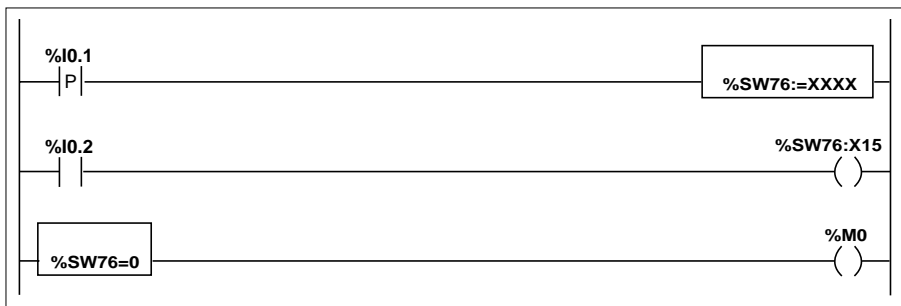
Ces quatre mots systèmes sont décrémentés individuellement par le système toutes les millisecondes **si leur valeur est positive**.

Une temporisation multiple peut être réalisée par chargement successifs d'un de ces mots ou par le test de valeurs intermédiaires.

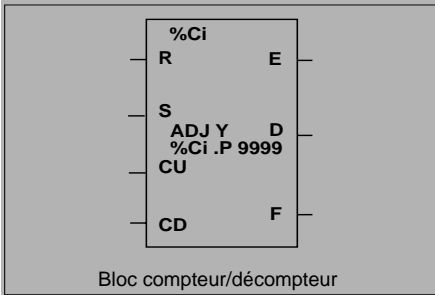
Si un de ces quatre mots système est inférieur à 0, il ne sera pas modifié, un temporisateur peut être ainsi "gelé" par la mise à 1 du bit 15 correspondant puis "dégelé" par sa remise à 0.

Exemple de programmation:

```
LDR %I0.1      (lancement du temporisateur sur front montant de %I0.1)
[%SW76:=XXXX] (XXXX= valeur désirée)
LD  %I0.2      (gestion optionnelle du gel, l'entrée I0.2 sert de gel)
ST  %SW76:X15
LD  [%SW76=0]  (test de fin du temporisateur)
ST  %M0
.....
```



2.2-4 Blocs fonction compteur/décompteur %Ci



Le bloc fonction compteur/décompteur permet d'effectuer le comptage ou le décomptage d'événements, ces deux opérations pouvant être simultanées.

Caractéristiques

Numéro de compteur	%Ci	0 à 15
Valeur courante	%Ci.V	Mot incrémenté ou décrémenté en fonction des entrées (ou des instructions) CU et CD. Peut être lu, testé mais non écrit par programme (1).
Valeur de présélection	%Ci.P	$0 \leq \%Ci.P \leq 9999$. Mot pouvant être lu, testé, écrit (mis à la valeur 9999 par défaut).
Réglage par terminal	O/N	O: possibilité de modification de la valeur de présélection en mode Réglage. N : pas d'accès en mode Réglage.
Entrée (ou instruction) remise à zéro	R	Sur état 1 : %Ci.V = 0.
Entrée (ou instruction) présélection	S	Sur état 1: %Ci.V = %Ci.P.
Entrée (ou instruction) comptage	CU	Incrémente %Ci.V sur front montant.
Entrée (ou instruction) décomptage	CD	Décrémente %Ci.V sur front montant.
Sortie débordement	E (Empty)	Le bit associé %Ci.E=1, lorsque décomptage %Ci.V passe de 0 à 9999 (mis à 1 quand %Ci.V devient égal à 9999, est remis à 0 si le compteur continue à décompter).
Sortie présélection atteinte	D (Done)	Le bit associé %Ci.D=1, lorsque %Ci.V=%Ci.P.
Sortie débordement	F (Full)	Le bit associé %Ci.F =1 lorsque %Ci.V passe de 9999 à 0 (mis à 1 quand %Ci.V devient égal à 0, est remis à 0 si le compteur continue à compter).

(1) %Ci.V peut être modifiée par le terminal en mode Réglage.

Fonctionnement

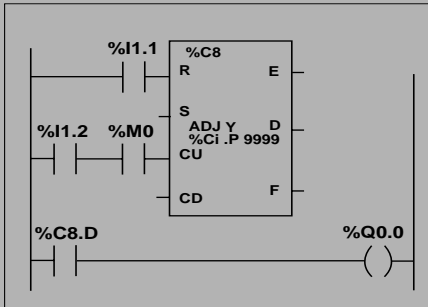
- **Comptage** : à l'apparition d'un front montant sur l'entrée comptage CU (ou activation de l'instruction CU), la valeur courante est incrémentée d'une unité. Lorsque cette valeur est égale à la valeur de présélection %Ci.P, le bit de sortie %Ci.D "présélection atteinte" associé à la sortie D passe à l'état 1. Le bit de sortie %Ci.F (débordement comptage) passe à l'état 1 lorsque %Ci.V passe de 9999 à 0, il est remis à 0 si le compteur continue à compter.
- **Décomptage** : à l'apparition d'un front montant sur l'entrée "décomptage" CD (ou activation de l'instruction CD), la valeur courante %Ci.V est décrémentée d'une unité. Le bit de sortie %Ci.E (débordement décomptage) passe à l'état 1 lorsque %Ci.V passe de 0 à 9999, , il est remis à 0 si le compteur continue à décompter.
- **Comptage/décomptage** : pour utiliser simultanément les fonctions comptage et décomptage (ou activer les instructions CD et CU), il est nécessaire de commander les deux entrées correspondantes CU et CD ; ces deux entrées étant scrutées successivement. Si les deux entrées sont à 1 simultanément, la valeur courante reste inchangée (ou si les 2 instructions sont activées simultanément).
- **Remise à zéro** : dès la mise à l'état 1 de l'entrée R (ou activation de l'instruction), la valeur courante %Ci.V est forcée à 0, les sorties %Ci.E, %Ci.D et %Ci.F sont à 0. L'entrée "remise à zéro" est prioritaire.
- **Présélection** : si l'entrée S "présélection" est à l'état 1 (ou l'instruction S active) et l'entrée R "remise à zéro" à l'état 0 (ou l'instruction R non active), la valeur courante %Ci.V prend la valeur %Ci.P et la sortie %Ci.D prend la valeur 1.

Cas spécifiques.

- **Incidence d'une "reprise à froid" : (%S0=1)**
 - mise à zéro de la valeur courante %Ci.V.
 - mise à 0 des bits de sorties %Ci.E, %Ci.D et %Ci.F.
 - l'initialisation de la valeur de présélection par celle définie en configuration.
- **Incidence d'une reprise à chaud (%S1=1), d'un passage en STOP** : n'a pas d'incidence sur la valeur courante du compteur (%Ci.V).
- **Incidence de la modification de la présélection %Ci.P** : la modification de la valeur de présélection par instruction ou en réglage est prise en compte lors de la gestion du bloc par l'application (activation de l'une des entrées).

Configuration et programmation

Comptage d'un nombre de pièces = 5000. Chaque impulsion sur l'entrée %I1.2 (lorsque le bit interne %M0 est à 1) provoque l'incrémentement du compteur %C8 et ce jusqu'à la valeur de présélection finale du compteur %C8 (bit %C8.D=1). La remise à zéro du compteur est provoquée par l'entrée %I1.1.

**Configuration**

Les paramètres à saisir en configuration sont les suivants :

- %Ci.P, fixé à 5000 dans cet exemple,
- Réglage : O

Programmation réversible

```

BLK  %C8
LD   %I1.1
R    %C8
LD   %I1.2
AND  %M0
CU   %C8
END_BLK
LD   %C8.D
ST   %Q0.0

```

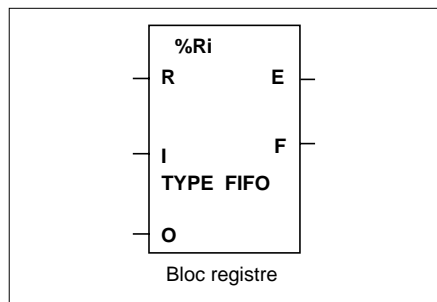
Programmation non réversible

```

LD   %I1.1
R    %C8
LD   %I1.2
AND  %M0
CU   %C8
LD   %C8.D
ST   %Q0.0

```

2.2-5 Blocs fonction registre %Ri



Un registre est un bloc mémoire permettant de stocker jusqu'à 16 mots de 16 bits de deux manières différentes :

- file d'attente (premier entré, premier sorti) appelée pile FIFO (First In, First Out),
- pile (dernier entré, premier sorti) appelée pile LIFO (Last In, First Out).

Caractéristiques

Numéro Registre	%Ri	0 à 3
Type	FIFO LIFO	File d'attente (choix par défaut). Pile
Mot d'entrée	%Ri.I	Mot d'entrée au registre. Peut être lu, testé, écrit.
Mot de sortie	%Ri.O	Mot de sortie du registre. Peut être lu, testé, écrit
Entrée (ou instruction) "Stockage"	I (In)	Sur front montant provoque le stockage du contenu du mot %Ri.I dans le registre.
Entrée (ou instruction) "Déstockage"	O (Out)	Sur front montant provoque le rangement d'un mot d'information dans le mot %Ri.O.
Entrée (ou instruction) "Remise à zéro"	R (Reset)	Sur état 1 initialise le registre.
Sortie "Vide"	E (Empty)	Le bit %Ri.E associé indique que le registre est vide. Peut être testé.
Sortie "Plein"	F (Full)	Le bit %Ri.F associé indique que le registre est plein. Peut être testé.

Fonctionnement

FIFO (First In, First Out)

La première information entrée est la première à être sortie.

Lorsqu'une demande d'entrée est prise en compte (front montant sur l'entrée I ou activation de l'instruction I), le contenu du mot d'entrée %Ri.I préalablement chargé est stocké au plus haut de la file (fig a).

Lorsque la file est pleine (sortie F=1), le stockage est impossible.

Lorsqu'une demande de sortie est prise en compte (front montant sur entrée O ou activation de l'instruction O) le mot d'information le plus bas de la file est rangé dans le mot de sortie %Ri.O et le contenu du registre est décalé d'un pas vers le bas (fig. b).

Lorsque le registre est vide (sortie E=1) le déstockage est impossible ; le mot de sortie %Ri.O n'évolue plus et conserve sa valeur.

La file peut être réinitialisée à tout moment (état 1 sur l'entrée R ou activation de l'instruction R).

LIFO (Last In, First Out)

La dernière information entrée est la première à être sortie.

Lorsqu'une demande d'entrée est prise en compte (front montant sur l'entrée (ou activation de l'instruction I), le contenu du mot d'entrée %Ri.I préalablement chargé est stocké au plus haut de la pile (fig. c).

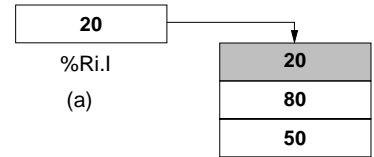
Lorsque la pile est pleine (sortie F à 1), le stockage est impossible.

Lorsqu'une demande de sortie est prise en compte (front montant sur l'entrée O ou activation de l'instruction O), le mot d'information le plus haut (dernière information entrée) est rangé dans le mot %Ri.O (fig d).

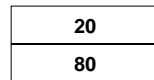
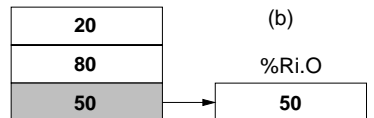
Lorsque le registre est vide (sortie E= 1), le déstockage est impossible, le mot de sortie %Ri.O n'évolue plus et conserve sa dernière valeur. La pile peut être réinitialisée à tout moment (état 1 sur entrée R ou activation de l'instruction R). L'élément pointé est alors le plus haut dans la pile.

Exemple :

Stockage du contenu de %Ri.I au plus haut de la pile.

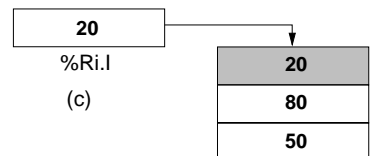


Déstockage de la première information et rangement de cette dernière dans %Ri.O

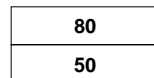
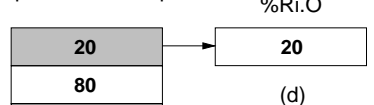


Exemple :

Stockage du contenu de %Ri.I au plus haut de la pile.



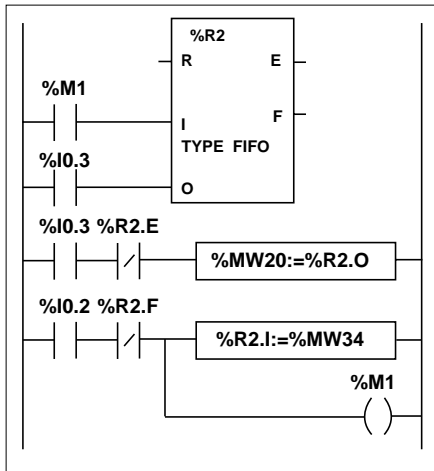
Déstockage du mot d'information le plus haut de la pile



Cas spécifiques

- **Incidence d'une reprise "à froid"** : (%S0=1) provoque l'initialisation du contenu du registre. Le bit de sortie %Ri.E associé à la sortie E est mis à 1. Les mots %Ri.I et %Ri.O sont remis à 0.
- **Incidence d'une reprise "à chaud"** (%S1=1) n'a pas d'incidence sur le contenu du registre ainsi que sur l'état des bits de sorties.

Programmation et configuration



L'exemple de programme montre le chargement de %R2.I par le mot %MW34 sur demande de l'entrée %I0.2, si le registre R2 n'est pas plein (%R2.F=0). La demande d'entrée dans le registre est assurée par %M1. La demande de sortie est faite par l'entrée %I0.3 et le rangement de %R2.O dans %MW20 s'effectue si le registre n'est pas vide (%R2.E=0).

Configuration

Le seul paramètre à saisir en configuration est le type de registre FIFO (par défaut) ou LIFO.

Programme réversible

```

BLK  %R2
LD   %M1
I
LD   %I0.3
O
END_BLK
LD   %I0.3
ANDN %R2.E
[%MW20:=%R2.O]
LD   %I0.2
ANDN %R2.F
[%R2.I:=%MW34]
ST   %M1

```

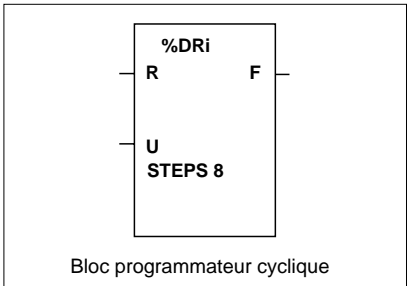
Programme non réversible

```

LD   %M1
I   %R2
LD   %I0.3
O   %R2
LD   %I0.3
ANDN %R2.E
[%MW20:=%R2.O]
LD   %I0.2
ANDN %R2.F
[%R2.I:=%MW34]
ST   %M1

```

2.2-6 Blocs fonction programmeur cyclique %DRi



D'un principe de fonctionnement similaire au programmeur à cames, le programmeur cyclique change de pas en fonction d'événements extérieurs. A chaque pas, le point haut d'une came donne un ordre exploité par l'automatisme. Dans le cas du programmeur cyclique, ces points hauts seront symbolisés par un état 1 au niveau de chaque pas et sont affectés à des bits de sortie %Qi.j ou interne %Mi appelés bits d'ordres.

Caractéristiques

Numéro	%DRi	0 à 3
Numéro du pas en cours	%DRi.S	$0 \leq \%DRi.S \leq 7$. Mot pouvant être lu et testé. Ne peut être écrit dans le programme qu'à partir d'une valeur immédiate décimale.
Nombre de pas		1 à 8 (8 par défaut)
Entrée (ou instruction) "retour au pas 0"	R (RESET)	Sur état 1 initialise le programmeur au pas 0.
Entrée (ou instruction) "avance"	U (UP)	Sur front montant provoque l'avance d'un pas du programmeur et la mise à jour des bits d'ordres.
Sortie	F (FULL)	Indique que le dernier pas défini est en cours. Le bit %DRi.F associé peut être testé (%DRi.F = 1 si %DRi.S = nombre de pas configuré - 1).
Bits d'ordres		Sorties ou bits internes associés au pas (16 bits d'ordres).

Fonctionnement

Le programmeur cyclique se compose :

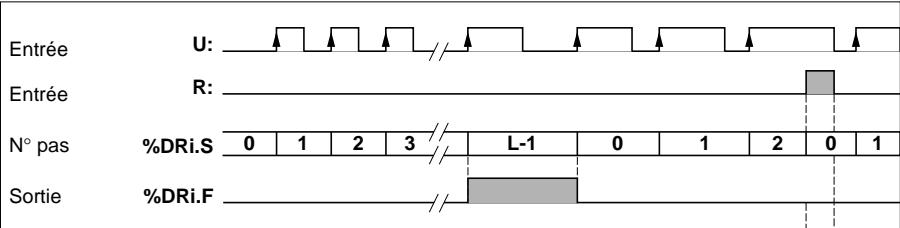
- d'une matrice de données constantes (les cames) organisée en 8 pas de 0 à 7 et de 16 informations binaires (état du pas) rangées en colonnes repérées de 0 à F.
- d'une liste de bits d'ordres (1 par colonne) correspondants à des sorties %Q0.i ou %Q1.i ou à des bits internes %Mi. Lors du pas en cours les bits d'ordres prennent les états binaires définis pour ce pas,

Le tableau ci-dessous résume les caractéristiques principales du programmeur cyclique.

Colonne	0	1	2		D	E	F
Bits d'ordres	%Q0.1	%Q0.3	%Q1.5		%Q0.6	%Q0.5	%Q1.0
Pas 0	0	0	1		1	1	0
Pas 1	1	0	1		1	0	0
Pas 5	1	1	1		0	0	0
Pas 6	0	1	1		0	1	0
Pas 7	1	1	1		1	0	0

Dans l'exemple ci-dessus, le pas 5 étant en cours, les bits d'ordres %Q0.1; %Q0.3 et %Q1.5 sont mis à l'état 1 ; les bits d'ordres %Q0.6 ; %Q0.5 et %Q1.0 sont mis à l'état 0. Le numéro du pas en cours est incrémenté à chaque front montant sur l'entrée U (ou activation de l'instruction U). Ce numéro peut être modifié par programme.

Diagramme de fonctionnement

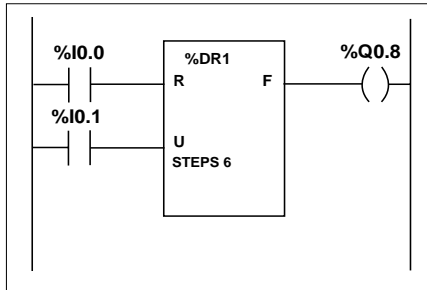


Cas spécifiques

- Incidence d'une "reprise à froid" :** (%S0=1) provoque la réinitialisation du programmeur au pas 0 (avec mise à jour des bits d'ordres).
- Incidence d'une "reprise à chaud" :** (%S1=1) provoque la mise à jour des bits d'ordres, suivant le pas en cours.
- Incidence d'un saut de programme :** le fait de ne pas scruter le programmeur cyclique ne provoque pas de remise à 0 des bits d'ordres.
- Mise à jour des bits d'ordre :** ne s'effectue que lors d'un changement de pas ou lors d'une reprise à chaud ou à froid.
- Incidence des instructions relais maître MCS/MCR :** lorsqu'un programmeur cyclique est utilisé entre 2 instructions MCS/MCR, les bits d'ordres sont remis à 0 si la condition précédent MCS est à 0.

Programmation et configuration

Dans cet exemple, les 5 premières sorties %Q0.0 à %Q0.4 sont activées les unes à la suite des autres, chaque fois que l'entrée %I0.1 est mise à 1. L'entrée I0.0 réinitialise les sorties à 0.



Programmation réversible

```

BLK  %DR1
LD   %I0.0
R
LD   %I0.1
U
OUT_BLK
LD   F
ST   %Q0.8
END_BLK

```

Configuration

Les informations suivantes sont définies en configuration :

- le nombre de pas : 5
- les états des sorties (bits d'ordres) pour chaque pas du programmeur

Q0.0	1	2	3	4
Pas 0 :	0	0	0	0
Pas 1 :	1	0	0	0
Pas 2 :	0	1	0	0
Pas 3 :	0	0	1	0
Pas 4 :	0	0	0	1
Pas 5 :	0	0	0	1
- l'affectation des bits d'ordres

0: %Q0.0	1: %Q0.1
2: %Q0.2	3: %Q0.3
4: %Q0.4	

Programmation non réversible

```

LD   %I0.0
R    %DR1
LD   %I0.1
U    %DR1
LD   %DR1.F
ST   %Q0.8

```







2.3 Instructions Grafcet

2.3-1 Description

Les instructions Grafcet du langage PL7 permettent de traduire un Grafcet graphique de façon simple.

Le langage PL7 comprend 62 étapes maximum, y compris la ou les étapes initiales. Le nombre d'étapes actives simultanément n'est limité que par le nombre d'étapes.

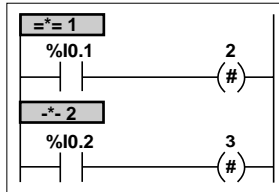
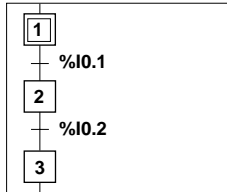
Le tableau ci-dessous regroupe l'ensemble des instructions et objets permettant de programmer un Grafcet.

Représentation graphique	Transcription en langage PL7	Rôle
 Etape initiale	=* = i	Début de l'étape initiale (1)
 Transition	# i	Activation de l'étape i après désactivation de l'étape en cours
 Etape	-* - i	Début de l'étape i et validation de la transition associée (1)
	#	Désactivation de l'étape en cours sans activation de tout autre étape
	#Di	Désactivation de l'étape en cours et de l'étape i spécifiée.
	=* = POST	Début du traitement postérieur et fin du traitement séquentiel
	%Xi	Bit associé à l'étape i, peut être testé n'importe où dans le programme mais ne peut être écrit que dans le traitement préliminaire (nombre d'étapes maxi : 62).
 Xi	LD %Xi, LDN %Xi, AND %Xi, ANDN %Xi OR %Xi, ORN %Xi XOR %Xi, XORN %Xi	Test de l'activité de l'étape i
 Xi (S)	S %Xi	Activation de l'étape i
 Xi (R)	R %Xi	Désactivation de l'étape i

(1) la première étape =* =i ou -* - écrite indique le début du traitement séquentiel donc la fin du traitement préliminaire.

Exemples

Séquence linéaire

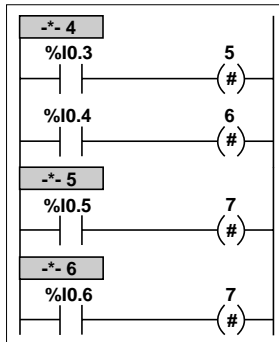
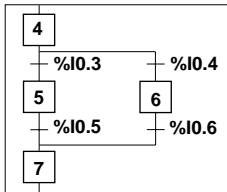


```

=* = 1
LD %I0.1
# 2
-* = 2
LD %I0.2
# 3

```

Aiguillage

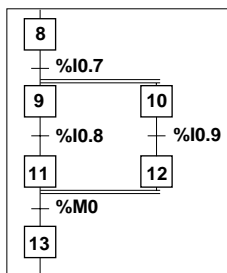


```

-* = 4
LD %I0.3
# 5
LD %I0.4
# 6
-* = 5
LD %I0.5
# 7
-* = 6
LD %I0.6
# 7

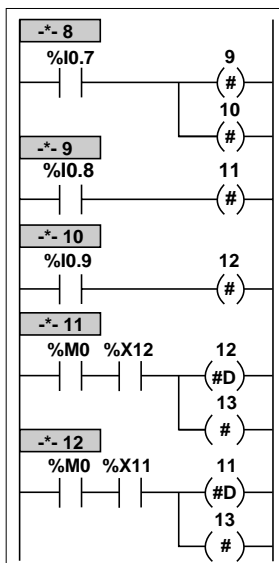
```

Séquences simultanées



Note :

Pour qu'un Grafcet soit opérationnel, il est nécessaire de déclarer au minimum une étape initiale $=* = i$ ou de prépositionner le graphe dans le traitement préliminaire à l'aide du bitsystème %S23. Voir exemple annexe A.10 intercalaire G.



```

-* = 8
LD %I0.7
# 9
LD %I0.8
# 10
-* = 9
LD %I0.9
# 11
-* = 10
LD %I0.9
# 12
-* = 11
LD %M0
AND %X12
#D 12
# 13
-* = 12
LD %M0
AND %X11
#D 11
# 13

```

2.3-2 Structure d'un programme

Un programme Grafset PL7 est structuré en 3 parties, chacune ayant un rôle spécifique.

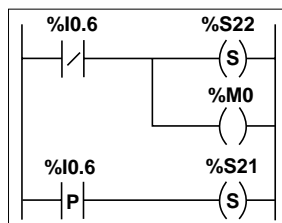
Traitement Exemple

Traitement préliminaire

Il est constitué d'une suite d'instructions assurant le traitement des :

- reprises secteur
- défaillances
- changements de modes
- prépositionnement du graphe
- logiques d'entrées

Il se termine par la première instruction *= ou *- rencontrée.



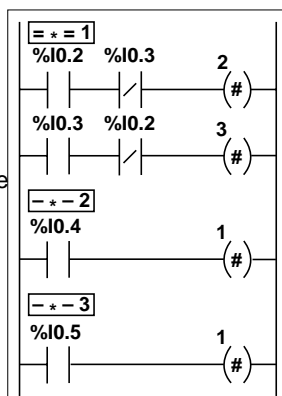
```
000 LDN  %I0.6
001 S     %S22
002 ST    %M0
003 LDR   %I0.6
004 S     %S21
```

Traitement séquentiel

Il est constitué du graphe (instructions représentant le graphe) :

- étapes
- actions associées à l'étape (voir annexe A.11 intercalaire G).
- transitions
- réceptivités

Il se termine par l'exécution de l'instruction *=POST

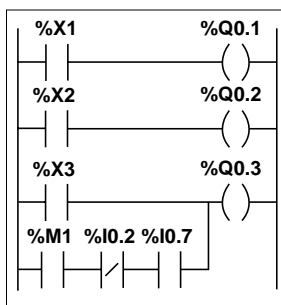


```
005 *= 1
006 LD  %I0.2
007 ANDN %I0.3
008 # 2
009 LD  %I0.3
010 ANDN %I0.2
011 # 3
012 *- 2
013 LD  %I0.4
014 # 1
015 *- 3
016 LD  %I0.5
017 # 1
```

Traitement postérieur

Il est constitué d'une suite d'instructions assurant le traitement :

- des ordres émanant du traitement séquentiel pour la commande des sorties
- sécurités indirectes spécifiques aux sorties.



```
018 *= POST
019 LD  %X1
020 ST  %Q0.1
021 LD  %X2
022 ST  %Q0.2
023 LD  %X3
024 OR(  %M1
025 ANDN %I0.2
026 AND  %I0.7
027 )
028 ST  %Q0.3
```

Remarque :

Le cycle de scrutation est celui défini ch 1.3 intercalaire A. Pour le traitement séquentiel, seules les étapes actives au départ du cycle et leurs instructions associées sont exécutées.

2.4 Instructions sur programme

2.4-1 Instructions de fin de programme END, ENDC, ENDCN

Les instructions END, ENDC et ENDCN permettent de définir la fin d'exécution de cycle du programme :

END : fin de programme inconditionnelle

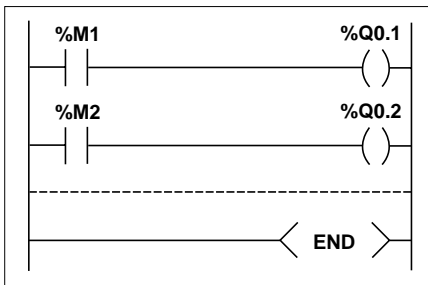
ENDC : fin de programme si résultat booléen de l'instruction de test précédente est à 1.

ENDCN : fin de programme si résultat booléen de l'instruction de test précédente est à 0

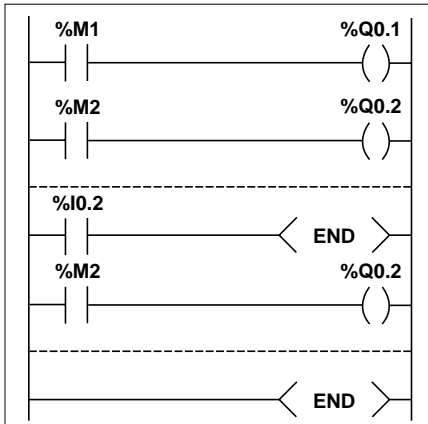
Par défaut (mode normal), lorsque la fin de programme est activée, il y a mise à jour des sorties et passage au cycle suivant.

Si la scrutation est périodique, il y a attente de fin de période, mise à jour des sorties et passage au cycle suivant.

Exemple :



```
LD %M1
ST %Q0.1
LD %M2
ST %Q0.2
.....
END
```



```
LD %M1
ST %Q0.1
LD %M2
ST %Q0.2
.....
LD %I0.2 → Si %I0.2 =1, fin de
ENDC      scrutation du pro-
             gramme
LD %M2
ST %Q0.2    Si %I0.2 =0, continue la
             scrutation du pro-
             gramme jusqu'à la nou-
             velle instruction END.
.....
END
```

2.4-2 Instruction NOP

L'instruction **NOP** n'effectue aucune action. Elle permet de "réserver" des lignes dans un programme et ainsi de pouvoir écrire par la suite des instructions sans modification des numéros de lignes.

2.4-3 Instructions de saut JMP, JMPC, JMPCN vers une étiquette %Li:

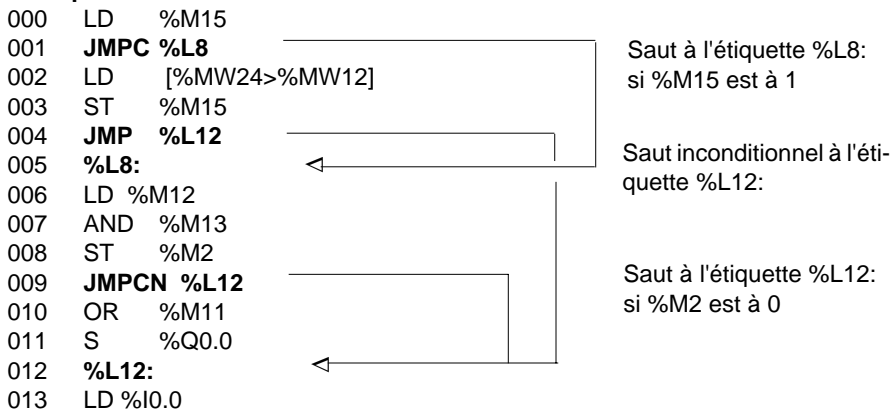
Les instructions JMP, JMPC et JMPCN provoquent l'interruption immédiate de l'exécution et la poursuite du programme à partir de la ligne de programme comportant l'étiquette %Li: (i = 0 à 15).

JMP : saut de programme inconditionnel

JMPC : saut de programme si résultat booléen de l'instruction de test précédente est à 1

JMPCN : saut de programme si résultat booléen de l'instruction de test précédente est à 0.

Exemples :



Notes :

- Cette instruction est interdite entre des parenthèses, elle ne doit donc pas figurer entre des instructions AND(, OR(, et une instruction fermeture de parenthèse").
- L'étiquette ne peut figurer qu'avant une instruction LD, LDN, LDR, LDF ou BLK.
- Le numéro i d'une étiquette %Li ne doit être déclaré qu'une seule fois dans un programme.
- Le saut de programme s'effectue vers une ligne de programmation située en aval ou en amont. Dans le cas d'un saut amont, il faut faire attention au temps d'exécution du programme, il est allongé et peut entraîner un dépassement de la période ou du cycle automate ayant pour effet le déclenchement du chien de garde.

2.4-4 Instructions sur sous-programme SRn, SRn:, RET

L'instruction **SRn** effectue l'appel au sous-programme repéré par l'étiquette **SRn**: si le résultat de l'instruction booléenne précédente est à 1.

L'instruction **RET** placée à la fin du sous-programme commande le retour au programme principal.

Le sous-programme est repéré par une étiquette SRn: avec n=0 à 15.

Exemple :

```

000 LD    %M15
001 AND   %M5
002 ST    %Q0.0
003 LD    [%MW24>%MW12]
004 SR8
005 LD    %I0.4
006 AND   %M13
007 .
008 .
009 .
010 END

```

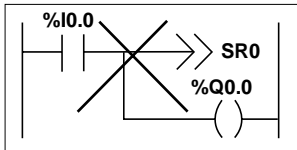
```

011 SR8:
012 LD 1
013 IN  %TM0
014 LD  %TM0.Q
015 ST  %M10
016 RET

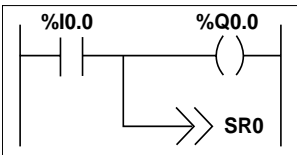
```

Notes :

- Un sous-programme ne doit pas appeler un autre sous-programme.
- Cette instruction est interdite entre des parenthèses, elle ne doit donc pas figurer entre des instructions AND(, OR(, et une instruction fermeture de parenthèse").
- L'étiquette ne peut figurer qu'avant une instruction LD ou BLK marquant le début d'une équation ou d'un réseau de contacts.
- Un appel de sous-programme ne doit pas être avant une instruction d'affectation, exemple :



~~LD %I0.0
SR0
ST %Q0.0~~



LD %I0.0
ST %Q0.0
SR0

2.4-5 Instructions de relais maître MCS et MCR

Lorsque le résultat booléen de l'instruction précédant l'instruction MCS est à 0, l'exécution des lignes de programme suivant cette instruction est modifiée comme cela est indiquée dans le tableau ci-après jusqu'à ce que l'instruction MCR (non conditionnelle) soit exécutée (voir également chapitre C 4.6).

Instructions/blocs	Comportement
ST, STN	objet associé mis à 0
S, R	instructions non exécutées
SRi, JMP, JMPC, JMPCN	non exécutées
%TMi	réinitialisé
%DRi	bits d'ordres mis à 0
%FC	compteur figé et sorties réflexes remis à zéro
%PWM, %PLS	arrêt de génération des signaux de sorties
Autres blocs fonctions	non exécutés (maintenu dans l'état)
Blocs opérations	non exécutés

Exemple :

```

.....
002 LD    %I0.1
003 MCS
004 LD    %M1
005 ST    %Q0.1
006 LD    %I0.2
007 S     %Q0.2
008 MCR
.....

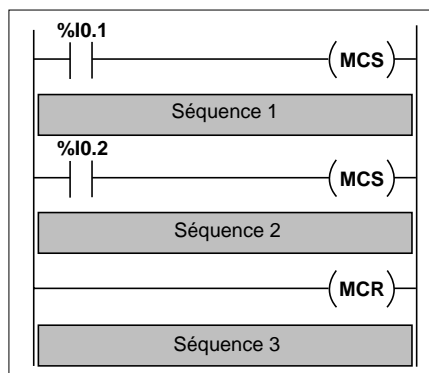
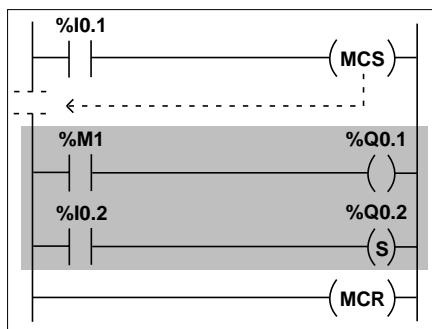
```

Lorsque %I0.1 est à 0, l'instruction MCS est activée, %Q0.1 est forcé à 0, la sortie %Q0.2 est maintenue.

Il est possible d'utiliser plusieurs instructions MCS pour une seule instruction MCR. Toutes les instructions MCS sont désactivées par une seule instruction MCR.

Lorsque %I0.1 est à 0, les séquences 1 et 2 sont modifiées, la séquence 3 est exécutée normalement.

Lorsque %I0.1 est à 1 et %I0.2 est à 0, la séquence 2 est modifiée, les séquences 1 et 3 sont exécutées normalement.



Si aucune instruction MCR n'est programmée après une instruction MCS, l'instruction MCS est effective jusqu'à l'instruction END ou la fin de programme.

Important

L'utilisation des instructions MCS et MCR est interdite dans les sous programmes, les réceptivités et les actions Grafcet.

3.1 Traitement numérique

3.1-1 Définition des principaux objets mots

Les objets mots sont adressés sous le format mot de longueur 16 bits, situés dans la mémoire de données, ils contiennent une valeur algébrique comprise entre -32768 et 32767 (excepté le compteur rapide qui évolue entre 0 et 65535).

Valeurs immédiates

Ce sont des valeurs algébriques de format homogène avec celui des mots 16 bits, qui permettent d'affecter des valeurs à ces mots. Elles sont stockées dans la mémoire programme et sont comprises entre -32768 et 32767.

Format des mots

Le contenu des mots ou valeurs est rangé en mémoire utilisateur en code binaire, sur 16 bits, avec la convention illustrée ci-dessous :

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Rang des bits
0	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1	Etat des bits
16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1		Poids des bits

En binaire signé, le bit de rang "F" est attribué par convention au signe de la valeur codée :

- bit "F" à 0 : le contenu du mot est une valeur positive,
- bit "F" à 1 : le contenu du mot est une valeur négative (les valeurs négatives sont exprimées en logique complément à 2).

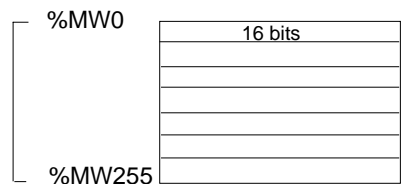
Les mots et valeurs immédiates peuvent être saisis ou restitués sous forme :

- **décimale** 1579 (maxi : 32767, mini :-32768)
- **hexadécimale** 16#A536 (maxi : 16#FFFF, mini : 16#0000)
autre syntaxe possible : #A536.

Mots internes

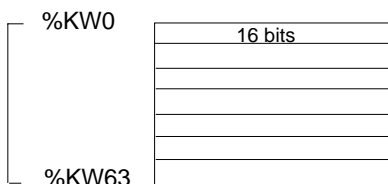
Les mots internes sont destinés à stocker des valeurs.

Les mots %MW0 à %MW255 sont accessibles directement par programme (en lecture/écriture). Ils sont utilisés comme mots de travail.



Mots constants

Les mots constants mémorisent des valeurs constantes ou des messages alphanumériques. Leur contenu ne peut être écrit ou modifié que par le terminal (en mode CONFIGURATION). Ces mots sont rangés dans la mémoire programme. Les mots constants %KW0 à %KW63 sont accessibles directement par programme (en lecture uniquement).



Mots d'échange d'entrées/sorties

Les mots d'échange %IW/QW sont associés aux automates raccordés sur câble d'extension. Ils permettent les échanges entre les automates (voir ch 3-4).

Mots système

Ces mots de 16 bits assurent plusieurs fonctions, ils donnent accès à des informations provenant directement de l'automate par lecture des mots %SWi (ex : valeurs des points de réglage analogique), ils permettent d'agir sur l'application (ex : réglage de l'horodateur). Le rôle de chacun de ces mots est expliqué chapitre 5.

Objets bits extraits de mots

Il est possible d'extraire d'un mot l'un de ses 16 bits. Le repère du mot est alors complété par le rang du bit extrait séparé par deux points.

Syntaxe : % Objet Mot:Xk avec k = 0 à 15 rang du bit de l'objet mot.

Exemple : %MW5:X6 bit de rang 6 du mot interne %MW5.

Liste des opérandes mots

Type	Adresse (ou valeur)	Nombre maximum	Accès en écriture	Voir Ch.
Valeurs immédiates • base 10 • base 16	ex : 2103 ex : 16#AF0D		Non	
Mots internes	%MWi	256	Oui	-
Mots constants	%KWi	64	Non (1)	-
Mots système	%SWi	128	selon i	5.2
Mots de blocs fonction	%TMi.P %Ci.P...			2.2-1 3.3
Mots d'échange d'entrées de sorties	%lWi.j %QWi.j	8 8	Non Oui	3.5
Bits extraits de mots				
• internes	%MWi:Xk	256 x 16	oui	
• système	%SWi:Xk	128 x 16	selon i	
• constants	%KWi:Xk	64 x 16	non	
• d'entrées	%lWi.j:Xk	8 x 16	non	
• de sorties	%QWi.i:Xk	8 x 16	oui	

(1) la saisie des mots constants s'effectue dans le mode configuration.

3.1-2 Objets structurés

Chaînes de bits

Les chaînes de bits sont des suites d'objets bits adjacents de même type et de longueur définie : L .

Exemple de chaîne de bits :

%M8:6 (1)

%M8

%M9

%M10

%M11

%M12

%M13

Type	Adresse	Taille maximum	Accès en écriture
Bits d'entrées TOR	%I0:L ou %I1:L	0<L<17	Non
Bits de sorties TOR	%Q0:L ou %Q1:L	0<L<17	Oui
Bits système	%Si:L avec i multiple de 8	0<L<17 et i+L≤128	Selon i
Bits étapes	%Xi:L avec i multiple de 8	0<L<17 et i+L≤63	Oui (par programme)
Bits internes	%Mi:L avec i multiple de 8	0<L<17 et i+L≤128	Oui

Les chaînes de bits sont exploitables à l'aide de l'instruction d'affectation := (voir instructions d'affectation ch 3.1-4).

Tableaux de mots

Les tableaux de mots sont des suites de mots adjacents de même type et de longueur définie : L.

Exemple de tableaux de mots :

%KW10

16 bits

%KW10:7

%KW16

Type	Adresse	Taille maximum	Accès en écriture
Mots internes	%MWi:L	0<L<256 et i+L≤256	Oui
Mots constantes	%KWi:L	0<L et i+L≤64	Non
Mots système	%SWi:L	0<L et i+L≤128	Selon i

Les tableaux de mots sont exploitables à l'aide de l'instruction d'affectation := (voir instructions d'affectation ch 3.1-4).

- (1) %M8:6 est correct (8 est un multiple de 8)
- %M10:16 est incorrect (10 n'est pas un multiple de 8)

Mots indexés

• Adressage direct

L'adressage des objets est dit direct, quand l'adresse de ces objets est fixe et définie à l'écriture du programme.

Exemple : %MW26 (mot interne d'adresse 26)

• Adressage indexé

En adressage indexé, l'adresse directe de l'objet est complétée d'un index : à l'adresse de l'objet est ajouté le contenu de l'index. L'index est défini par un mot interne %MWi. Le nombre de "mots index" n'est pas limité.

Exemple : %MW108[%MW2] : mot d'adresse direct 108 + contenu du mot %MW2.
Si le mot %MW2 a pour contenu la valeur 12, écrire %MW108[%MW2] équivaut donc à écrire %MW120.

Type	Adresse	Taille maximum	Accès en écriture
Mots internes	%MWi[%MWj]	$0 \leq i + \%MWj < 256$	Oui
Mots constants	%KWi[%MWj]	$0 \leq i + \%MWj < 64$	Non

Les mots indexés sont exploitables à l'aide de l'instruction d'affectation := (voir instruction d'affectation ch 3.1-4) et dans les instructions de comparaison (voir instructions de comparaison ch 3.1-5).

Ce type d'adressage permet de parcourir successivement une suite d'objets de même nature (mots internes, mots constants...), en modifiant par programme le contenu du mot index.

• Débordement d'index, bit système %S20

Il y a débordement d'index dès que l'adresse d'un objet indexé dépasse les limites de la zone incluant ce même type d'objet, c'est-à-dire quand :

- adresse objet + contenu de l'index inférieur à la valeur zéro,
- adresse objet + contenu de l'index supérieur à 255 (pour les mots %MWi) ou 63 (pour les mots %KWi).

En cas de débordement d'index, le système provoque la mise à l'état 1 du bit système %S20 et l'affectation de l'objet s'effectue avec une valeur d'index égale à 0.

La surveillance du débordement est à la charge de l'utilisateur : le bit %S20 doit être lu par le programme utilisateur pour traitement éventuel. Sa remise à zéro est à la charge de l'utilisateur.

%S20 (état initial = 0) :

- sur débordement d'index : mise à 1 par le système,
- acquittement débordement : mise à 0 par l'utilisateur, après modification de l'index.

3.1-3 Présentation des instructions numériques

Les instructions numériques portent d'une façon générale sur des mots de 16 bits (voir ch 3.1-1). Elles sont placées entre crochets. Elles sont exécutées si le résultat booléen de l'instruction de test précédant l'instruction numérique est à 1.

Remarque : l'affichage des instructions numériques s'effectue en 2 ou 3 lignes sur terminal FTX117.

3.1-4 Instructions d'affectation

Elles effectuent le chargement d'un opérande Op2 dans un opérande Op1

Syntaxe :

[Op1:=Op2]

<=>

Op2->Op1

Les opérations d'affectation peuvent être réalisées :

- sur chaînes de bits
- sur mots
- sur tableaux de mots

Affectation de chaînes de bits (voir objet chaîne de bits ch 3.1-2)

Les opérations sur chaîne de bits suivantes peuvent être réalisées :

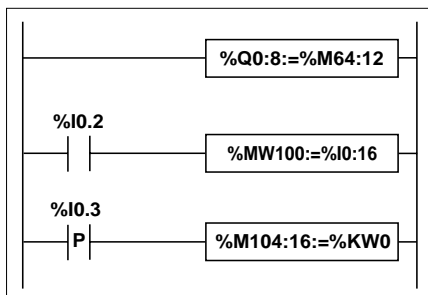
chaîne de bits -> chaîne de bits (ex 1)

chaîne de bits -> mot (ex 2)

mot -> chaîne de bits (ex 3)

valeur immédiate -> chaîne de bits

Exemples



LD 1
[%Q0:8:= %M64:12] (ex 1)

LD %I0.2
[%MW100:= %I0:16] (ex 2)

LDR %I0.3
[%M104:16:=%KW0] (ex 3)

Règles d'utilisation

- cas d'une affectation chaîne de bits -> mot : les bits de la chaîne sont transférés dans le mot en commençant par la droite (premier bit de la chaîne dans le bit 0 du mot), les bits du mot non concernés par le transfert (longueur<16) sont positionnés à 0.
- cas d'une affectation mot -> chaîne de bits : les bits du mot sont transférés à partir de la droite (le bit 0 du mot dans le premier bit de la chaîne).

Syntaxe

Opérateur	Syntaxe	Opérande 1 (OP1)	Opérande 2 (OP2)
:=	[Op1: = Op2] L'opérande 1 (Op1) prend la valeur de l'opérande 2 (Op2)	%MWi,%QWi, %SWi %MWi[MWi], %Mi:L,%Qi:L,%Si:L, %Xi:L	Valeur immédiate, %MWi, %KWi,%IW,%QW,%SWi, %BLK.x,%MWi[MWi], %KWi[MWi], %Mi:L,%Qi:L, %Si:L,%Xi:L, %li:L

Note : l'abréviation %BLK.x (par ex : %C0.P) est utilisée pour désigner tout mot de bloc fonction.

Affectation de mots

Les opérations d'affectation sur mots suivantes peuvent être réalisées :

mot -> mot (ex 1)

mot indexé -> mot

valeur immédiate -> mot (ex 3)

chaîne de bits -> mot

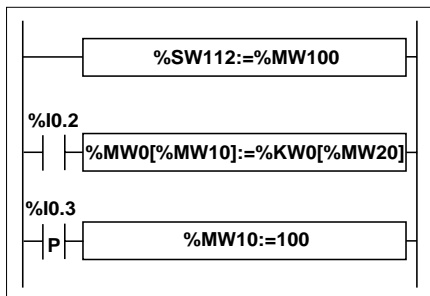
mot -> mot indexé

mot indexé -> mot indexé (ex 2)

valeur immédiate -> mot indexé

mot -> chaîne de bits

Exemples



LD 1
[%SW112 := %MW100] (ex 1)

LD %I0.2
[%MW0[%MW10] :=
%KW0[%MW20]] (ex 2)

LDR %I0.3
[%MW10:=100] (ex 3)

Syntaxe

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérande 2 (Op2)
:=	[Op1: = Op2] L'opérande 1 (Op1) prend la valeur de l'opérande 2 (Op2)	%MWi,%QWi, %SWi %MWi[MWi], %Mi:L,%Qi:L,%Si:L, %Xi:L	Valeur immédiate, %MWi, %KWi, %IW, %QW, %SWi, %BLK.x, %MWi[MWi], %KWi[MWi], %Mi:L,%Qi:L, %Si:L,%Xi:L, %li:L

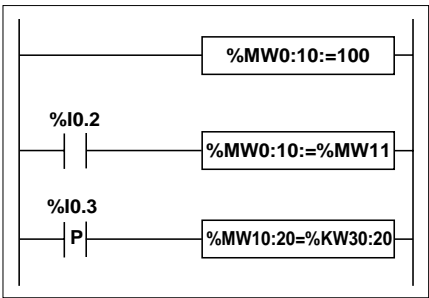
Notes :

- L'abréviation %BLK.x (par ex : R3.I) est utilisée pour désigner tout mot de bloc fonction.
- Pour les chaînes de bits %Mi:L, %Si:L et %Xi:L, l'adresse du premier bit de la chaîne de bit (i) doit être multiple de 8 (0, 8, 16, ..., 96, ...).

Affectation de tableaux de mots (voir objet tableaux de mots ch 3.1-2)

Les opérations d'affectation de tableaux de mots suivantes peuvent être réalisées :
valeur immédiate -> tableau de mots (ex 1)
mot -> tableau de mots (ex 2)
tableau de mots -> tableau de mots (ex 3)

Exemples



```
LD      1
[%MW0 :10:= 100]           (ex 1)

LD      %I0.2
[%MW0:10 := %MW11]        (ex 2)

LDR     %I0.3
[%MW10:20=%KW30:20]       (ex 3)
```

Syntaxe

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérande 2 (Op2)
:=	[Op1: = Op2] L'opérande 1 (Op1) prend la valeur de l'opérande 2 (Op2),	%MWi:L,%SWi:L	%MWi:L,%KWi:L,%SWi:L, Valeur Immédiate, %MWi, %KWi, %IW, %QW, %SWi, %BLK.x

B

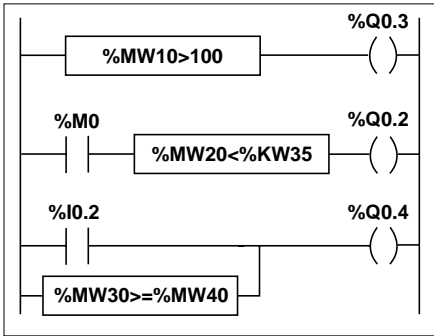
3.1-5 Instructions de comparaison

Les instructions de comparaison permettent de comparer deux opérandes.

- > : teste si l'opérande 1 est supérieur à l'opérande 2
- >= : teste si l'opérande 1 est supérieur ou égal à l'opérande 2.
- < : teste si l'opérande 1 est inférieur à l'opérande 2.
- <= : teste si l'opérande 1 est inférieur ou égal à l'opérande 2.
- = : teste si l'opérande 1 est égal à l'opérande 2.
- <> : teste si l'opérande 1 est différent de l'opérande 2.

Structure

La comparaison est réalisée à l'intérieur de crochets figurant derrière des instructions LD, AND et OR. Le résultat est à 1 lorsque la comparaison demandée est vraie.



```
LD      [%MW10 > 100]
ST      %Q0.3
LD      %M0
AND     [%MW20 < %KW35]
ST      %Q0.2
LD      %I0.2
OR      [%MW30 >= %MW40]
ST      %Q0.4
```

Syntaxe

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérandes 2 (Op2)
>, >=, <, <=	LD[Op1 Opérateur Op2]	%MWi,%KWi,%IW,	Valeur immédiate, %MWi,
=, <>	AND[Op1 Opérateur Op2] OR[Op1 Opérateur Op2]	%QW,%SWi,%BLK.x	%KWi,%IW,%QW,%SWi, %BLK.x,%MWi[%MWi], %KWi[%MWi]

Remarque

Les instructions de comparaison peuvent être utilisées au sein de parenthèses.

Exemple :

```
LD      %M0
AND(    [%MW20 > 10]
OR      %I0.0
)
ST      %Q0.1
```

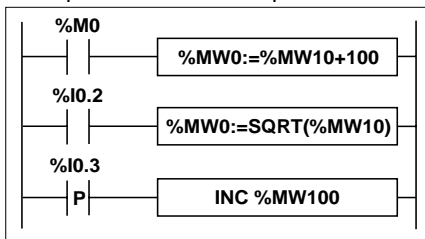

3.1-6 Instructions arithmétiques

Ces instructions permettent de réaliser une opération arithmétique entre deux opérandes ou sur un opérande.

- +** : addition de deux opérandes, **REM** : reste de la division de 2 opérandes,
- : soustraction de deux opérandes, **SQRT** : racine carré d'un opérande,
***** : multiplication de deux opérandes, **INC** : incrémentation d'un opérande,
/ : division de deux opérandes, **DEC** : décrémentation d'un opérande.

Structure

Les opérations arithmétiques sont réalisées de la manière suivante :



```
LD    %M0
[%MW0 := %MW10 + 100]
LD    %I0.2
[%MW0 := SQRT(%MW10)]
LDR   %I0.3
[INC %MW100]
```

Syntaxe

Elle dépend des opérateurs utilisés, voir tableau ci-dessous.

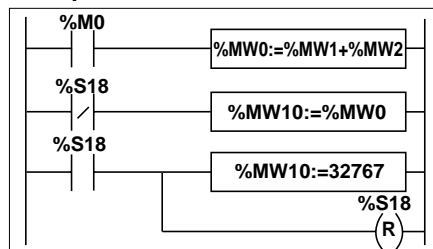
Opérateur	Syntaxe	Opérande 1 (Op1)	Opérandes 2 et 3 (Op2 et3)
+, -, *, /, REM (1)	[Op1 := Op2 Opérateur Op3]	%MWi, %QWi, %SWi	Val. imm. (2), %MWi, %KWi, %IW, %QW, %SWi, %BLK.x
SQRT	[Op1 := SQRT(Op2)]		
INC, DEC	[Opérateur Op1]		

Règles d'utilisation

• Addition : Dépassement de capacité pendant l'opération

Dans le cas où le résultat dépasse les limites -32768 ou +32767, le bit %S18 (overflow) est mis à l'état 1. Le résultat est donc non significatif. La gestion du bit %S18 s'effectue par programme utilisateur.

Exemple :



```
LD    %M0
[%MW0 := %MW1 + %MW2]
LDN   %S18
[%MW10 := %MW0]
LD    %S18
[%MW10 := 32767]
R     %S18
```

Dans le cas où %MW1 = 23241 et %MW2 = 21853, le résultat réel (45094) ne peut pas être exprimé dans un mot de 16 bits, le bit %S18 est mis à l'état 1 et le résultat obtenu (-20442) est erroné. Dans cet exemple lorsque le résultat est supérieur à 32767, sa valeur est fixée égale à 32767.

- (1) avec les TSX 07 version inférieure ou égale à V2.2, le résultat (Op1) de la division (/) ou du reste de la division (REM) est non significatif lorsque le contenu de l'opérande 3 (Op3) est supérieur à 255.
- (2) avec opérateur SQRT, l'opérande Op2 ne peut pas être une valeur immédiate.

Dépassement de la capacité absolue du résultat (arithmétique non signée) :

Lors de certains calculs, il est intéressant d'interpréter un opérande en arithmétique non signée (le bit F représente alors la valeur 32768). La valeur maximale pour un opérande étant de 65535. L'addition de 2 valeurs absolues (non signées) dont le résultat est supérieur à 65535 provoque un débordement. Ce débordement est signalé par la mise à 1 du bit système %S17 (carry) qui représente la valeur 65536.

Exemple 1 : [%MW2:=%MW0 + %MW1] avec %MW0 =65086, %MW1=65333

Le mot %MW2 contient le nombre 64883 et le bit %S17 est mis à l'état 1 et représente la valeur 65536. Le résultat en arithmétique non signée est donc égal à :
65536 + 64883 soit 130419.

Exemple 2 : [%MW2:=%MW0 + %MW1] avec %MW0 =45736 (soit -19800 en valeur signée), %MW1=38336 (soit -27200 en valeur signée).

Les deux bits système %S17 et %S18 sont mis à l'état 1. Le résultat en arithmétique signée (+18536) est erroné. En arithmétique non signée, le résultat (18536 + valeur de %S17 soit 84072) est correct.

- **Soustraction :**

Résultat négatif

Dans le cas où le résultat de la soustraction est inférieur à 0, le bit système %S17 est mis à l'état 1.

- **Multiplication :**

Débordement de capacité pendant l'opération

Dans le cas où le résultat dépasse la capacité du mot de rangement, le bit %S18 (overflow) est mis à l'état 1 et le résultat est non significatif.

- **Division/reste de la division :**

Division par 0

Dans le cas où le diviseur est égal à 0, la division est impossible et le bit système %S18 est mis à l'état 1, le résultat sera donc erroné.

Débordement de capacité pendant l'opération

Dans le cas où le quotient de la division dépasse la capacité du mot de rangement, le bit %S18 est mis à l'état 1.

- **Extraction de la racine carrée :**

L'extraction de racine carrée ne s'effectue que sur des valeurs positives. Le résultat est donc toujours positif. Dans le cas où l'opérande de la racine carrée est négatif, le bit système %S18 est mis à l'état 1 et le résultat est erroné.

Note : La gestion des bits système %S17 et %S18 est à la charge du programme utilisateur. Ils sont mis à 1 par l'automate, ils doivent être remis à zéro par le programme pour pouvoir être ré-utilisés (voir exemple page précédente).

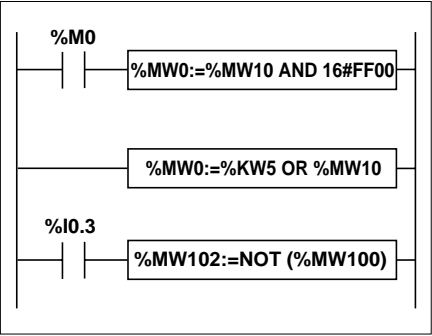
3.1-7 Instructions logiques

Les instructions associées permettent de réaliser une opération logique entre deux opérandes ou sur un opérande.

- AND** : ET (bit à bit) entre deux opérandes,
- OR** : OU logique (bit à bit) entre deux opérandes,
- XOR** : OU exclusif (bit à bit) entre deux opérandes,
- NOT** : complément logique (bit à bit) d'un opérande.

Structure

Les opérations logiques sont réalisées de la manière suivante :



```
LD    %M0
[%MW0 := %MW10 AND 16#FF00]

LD    1
[%MW0 := %KW5 OR %MW10]

LD    %I0.3
[%MW102:= NOT (%MW100)]
```

Syntaxe

Elle dépend des opérateurs utilisés, voir tableau ci-dessous.

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérandes 2 et 3 (Op2 et3)
AND, OR, XOR	[Op1 : = Op2 Opérateur Op3]	%MWi,%QWi,%SWi	Val. imm.(1), %MWi,%KW _i , %IW,%QW,%SW _i ,%BLK.x
NOT	[Op1 : = NOT(Op2)]		

Exemple : [%MW15:=%MW32 AND %MW12]

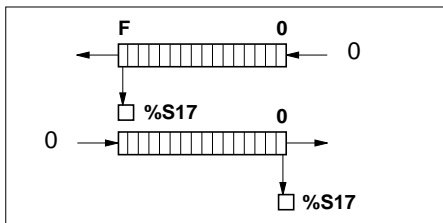
(1) avec l'opérateur NOT, l'opérande Op2 ne peut pas être une valeur immédiate.



3.1-8 Instructions de décalage

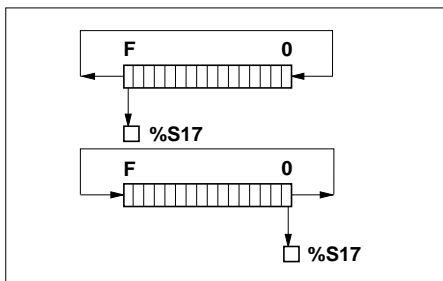
Les instructions de décalage consistent à déplacer les bits d'un opérande d'un certain nombre de positions vers la droite ou vers la gauche.

On distingue deux types de décalages :



- le **décalage logique** :

- **SHL**(op2,i) décalage logique à gauche de i positions.
- **SHR**(op2,i) décalage logique à droite de i positions.



- le **décalage circulaire** :

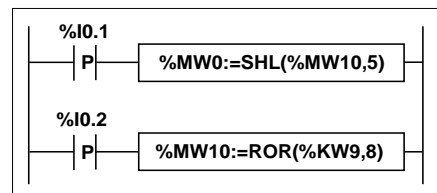
- **ROL**(op2,i) décalage circulaire à gauche de i positions.
- **ROR**(op2,i) décalage circulaire à droite de i positions.

L'opérande à décaler étant un opérande simple longueur, la variable i sera nécessairement comprise entre 1 et 16.

L'état du dernier bit sorti est mémorisé dans le bit %S17.

Structure

Les opérations logiques sont réalisées de la manière suivante :



```
LDR  %I0.1  
[MW0 := SHL(%MW10,5)]
```

```
LDR  %I0.2  
[MW10 := ROR(%KW9,8)]
```

Syntaxe

Elle dépend des opérateurs utilisés, voir tableau ci-dessous.

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérande 2 (Op2)
SHL, SHR ROL, ROR	[Op1: = Opérateur(Op2,i)]	%MWi,%QWi,%SWi	%MWi,%KWi,%IW,%QW, %SWi,%BLK.x

3.1-9 Instructions de conversion

Deux instructions de conversion sont proposées :

- **BTI** : conversion BCD --> Binaire
- **ITB** : conversion Binaire --> BCD

Rappel sur le code BCD

Le code BCD (Binary Coded Decimal) qui signifie Décimal codé binaire permet de représenter un chiffre décimal 0 à 9 par un ensemble de 4 bits. Un objet mot de 16 bits peut ainsi contenir un nombre exprimé sur 4 décades ($0 \leq N \leq 9999$). Lors d'une conversion, dans le cas où la valeur ne correspond pas à une valeur BCD, le bit système %S18 est mis à 1.

Décimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Exemple

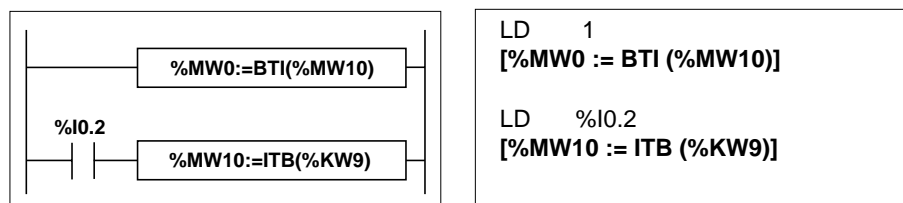
- Mot %MW5 exprimant la valeur BCD "2450" correspondant à la valeur binaire : 0010 0100 0101 0000
- Mot %MW12 exprimant la valeur décimale "2450" correspondant à la valeur binaire : 0000 1001 1001 0010

Le passage du mot %MW5 au mot %MW12 s'effectue par l'instruction BTI.

Le passage du mot %MW12 au mot %MW5 s'effectue par l'instruction ITB.

Structure

Les opérations de conversion sont réalisées de la manière suivante :



Syntaxe

Elle dépend des opérateurs utilisés, voir tableau ci-dessous.

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérande 2 (Op2)
BTI, ITB	[Op1 = Opérateur(Op2)]	%MWi,%QWi,%SWi	%MWi,%KWi,%IW,%QW, %SWi,%BLK.x

Exemples d'applications

L'instruction BTI s'utilise pour traiter une valeur de consigne présente en entrée automate sur des roues codeuses encodés BCD.

L'instruction ITB s'utilise pour afficher des valeurs numériques (ex: résultat de calcul, valeur courante de bloc fonction) sur afficheurs codés BCD.

3.2 Points de réglage analogique

Rappel du chapitre 1.8 intercalaire A :

Les automates TSX Nano disposent en face avant de :

- un potentiomètre de réglage analogique pour les automates TSX Nano 10 E/S,
- deux potentiomètres de réglage analogique pour les automates TSX Nano 16 et 20 E/S.

Programmation

Les valeurs numériques de 0 à 255 correspondantes aux valeurs analogiques fournies par ces potentiomètres sont disponibles dans les mots système :

- %SW112 pour le potentiomètre n°0
- %SW113 pour le potentiomètre n°1

Ces mots sont exploitables à l'aide des opérations arithmétiques. Ils peuvent être utilisés pour tout type de réglage par exemple : présélection de temporisateur, de compteur, réglage de fréquence du générateur d'impulsions, temps de préchauffage de machines...

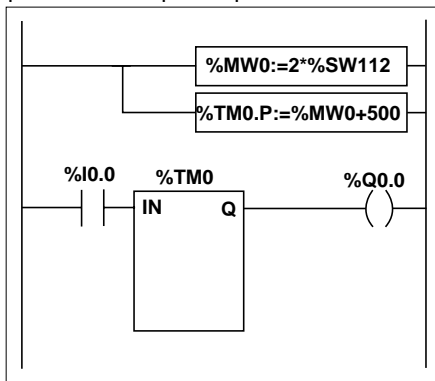
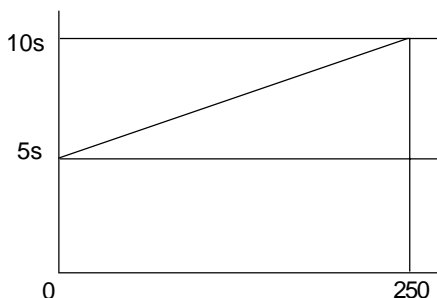
Exemple :

Réglage de la durée d'une temporisation de 5 à 10s à l'aide du potentiomètre n°0. On utilise pratiquement toute l'étendue du réglage possible du potentiomètre de 0 à 250 pour ce réglage.

En configuration les paramètres suivants sont choisis pour le bloc temporisation %TM0 :

- Type TON
- Base de temps BT : 10ms

La valeur de présélection du temporisateur est déduite de la valeur de réglage du potentiomètre par l'équation suivante $\%TM0.P := 2 * \%SW112 + 500$



```
LD    1
[%MW0:=2*%SW112]
[%TM0.P:=%MW0+500]
LD    %I0.0
IN     %TM0
LD     %TM0.Q
ST     %Q0.0
```

3.3 Blocs fonction spécifiques

3.3-1 Objets bits et mots associés aux blocs fonction spécifiques

Les blocs fonction spécifiques mettent en œuvre des objets bits et mots spécifiques du même type que les blocs fonction standards (Voir chapitre 2.2).

Liste des objets bits et mots de blocs fonction accessibles par programme

Blocs fonction spécifiques	Mots et bits associés		Adresses	Accès écriture	Voir Ch.
Sortie modulation de largeur %PWM	Mot	% de l'impulsion à 1 par rapport à la période totale	%PWM.R	oui	3.3-3
		Valeur de présélection de la période	%PWM.P	non	
Générateur d'impulsions %PLS	Mot	Valeur de présélection	%PLS.P	oui	3.3-4
		Nb d'impulsions à générer	%PLS.N	oui	
	Bit	Sortie en cours	%PLS.Q	non	
Sortie génération terminée		%PLS.D	non		
Compteur rapide %FC	Mot	Seuil i (i = 0 ou 1)	%FC.Si	oui	3.3-5
		Valeur courante	%FC.V	non	
		Valeur de présélection	%FC.P	oui	
	Bit	Sortie dépassement	%FC.F	non	
		Sortie dépassement seuil i	%FC.Thi	non	
Envoi de message %MSG	Bit	Sortie défaut liaison	%MSG.E	non	3.3-6
		Sortie liaison disponible	%MSG.D	non	
Registre bit à décalage %SBRi (i=0 à 7)	Bit	Bit du registre j=0 à 15	%SBRi.j	oui	3.3-7
Pas à pas %SCi (i=0 à 7)	Bit	Bit du pas à pas, j=0 à 255	%SCi.j	non	3.3-8

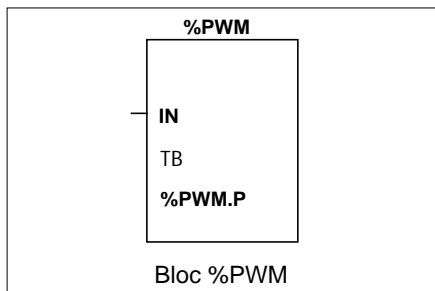
3.3-2 Principes de programmation

Comme les blocs fonction standards, les blocs fonction spécifiques peuvent être programmés de 2 façons différentes :

- de manière non réversible : par instructions spécifiques,
- de manière réversible : en simulant les blocs fonction du langage à contacts.

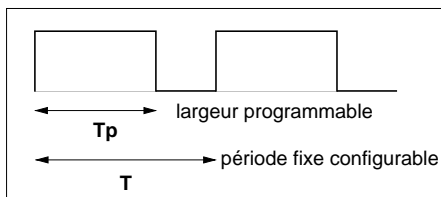
Voir chapitre 2.2-2

3.3-3 Sortie modulation de largeur %PWM



Le bloc fonction %PWM permet de générer un signal rectangulaire, sur la sortie automate %Q0.0, dont on peut faire varier la largeur (rapport cyclique) par programme (voir description ch4.6, int.A).

Cette fonction permet également de piloter un module de sortie analogique câblé sur la sortie %Q0.0 (voir chapitre B-4).



Caractéristiques

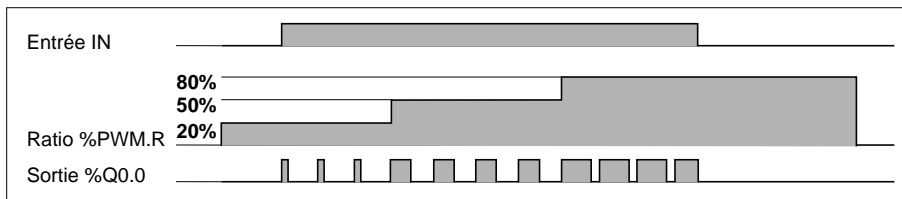
Base de temps	BT	0,1 ms (1), 10ms, 1s (valeur par défaut)
Présélection de la période	%PWM.P	<p>0 < %PMW.P < 32767 si base de temps de 10ms ou 1s.</p> <p>0 < %PMW.P < 255 si base de temps de 0,1ms (0 = fonction inactive)</p> <p>La présélection et la base de temps sont accessibles en écriture en configuration, elles permettent de fixer la période du signal</p> <p>T = %PWM.P x BT</p> <p>%PWM.P doit être choisi d'autant plus grand que les ratios à obtenir sont faibles.</p> <p>Gamme de période obtenue :</p> <ul style="list-style-type: none"> • 0,2 à 26 ms par pas de 0,1 ms, • 20 ms à 5,45 mn par pas de 10ms, • 2 s à 9,1 heures par pas de 1s.
Ratio de la période	%PWM.R	<p>0 ≤ %PMW.R ≤ 100 (2), ce mot donne le pourcentage du signal à l'état 1 sur la période (0 = valeur par défaut).</p> <p>La "largeur" Tp est donc égale à :</p> <p>Tp = T x (%PWM.R/100)</p> <p>Le mot %PWM.R est écrit par programme utilisateur, c'est ce mot qui permet donc d'effectuer la modulation de largeur.</p>
Entrée (ou instruction) génération d'impulsions	IN	<p>Sur état 1, génère le signal modulé en largeur sur la sortie %Q0.0.</p> <p>Sur état 0 met la sortie %Q0.0 à 0.</p>

(1) Cette base de temps est déconseillée pour les TSX Nano à sorties relais.

(2) Les valeurs supérieures à 100 seront prises en compte comme étant égales à 100.

Fonctionnement

La fréquence du signal en sortie %Q0.0 est fixée en configuration par le choix de la base de temps BT et de la présélection %PWM.P. La modulation de largeur du signal est obtenue en modifiant le ratio %PWM.R par programme.



Programmation et configuration

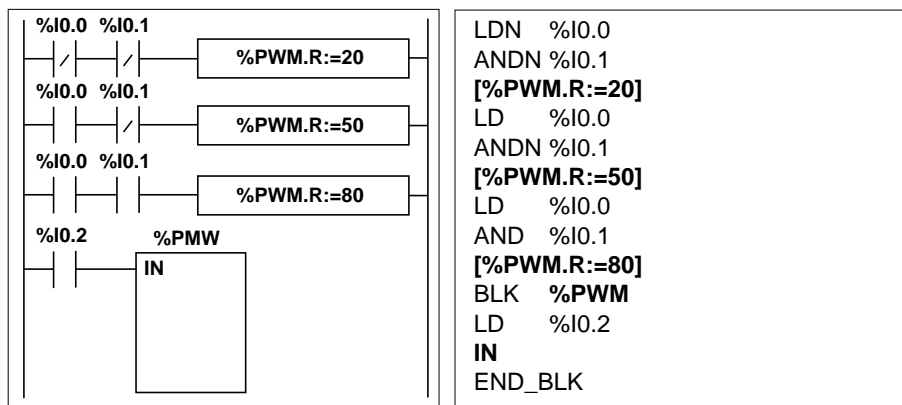
Dans cet exemple, la largeur du signal est modifiée par programme en fonction de l'état des entrées %I0.0 et %I0.1 de l'automate.

La période du signal est fixée à 500 ms en configuration.

Si %I0.0 et %I0.1 à 0, le ratio %PWM.R est fixé à 20%, la durée du signal à l'état 1 est alors de : $20\% \times 500 \text{ ms} = 100 \text{ ms}$.

Si %I0.0 à 0 et %I0.1 à 1, le ratio %PWM.R est fixé à 50% (durée 250 ms).

Si %I0.0 et %I0.1 à 1, le ratio %PWM.R est fixé à 80% (durée 400 ms).



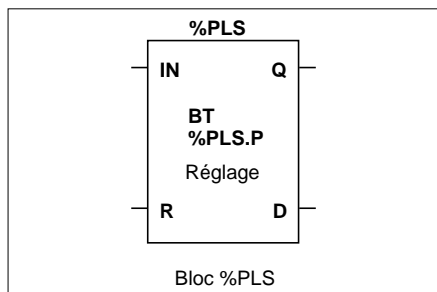
Configuration

Sortie %Q0.0 = sortie %PWM BT= 10ms %PWM.P=50

Cas spécifiques

- **Incidence d'une reprise "à froid"** : (%S0=1) provoque la mise à 0 du ratio %PWM.R
- **Incidence d'une "reprise à chaud"** (%S1=1) pas d'incidence
- **Incidence sur STOP automate** :
La sortie %Q0.0 est mis à 0 quel que soit l'état de bit système %S8. Si la version automate est inférieure ou égale à V2.2, la sortie %Q0.0 est mis à 0 si %S8 = 1 ou maintenue (génération du signal) si %S8 = 0.
- **Avec la base de temps 0,1 ms**, le forçage par un terminal de la sortie %Q0.0 n'arrête pas la génération.

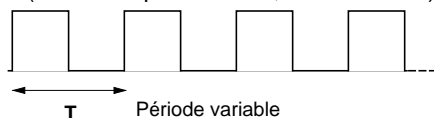
3.3-4 Sortie générateur d'impulsions %PLS



Le bloc fonction %PLS permet de générer un signal carré (rapport cyclique de 50% garanti si %PLS.P a une valeur paire) sur la sortie automate %Q0.0.

Ce signal peut être :

- de durée limitée, le nombre d'impulsions et la période sont écrits par programme (ou en configuration).
- de durée illimitée, la période est écrite par programme ou en configuration. (voir description ch 4.5, Intercalaire A).



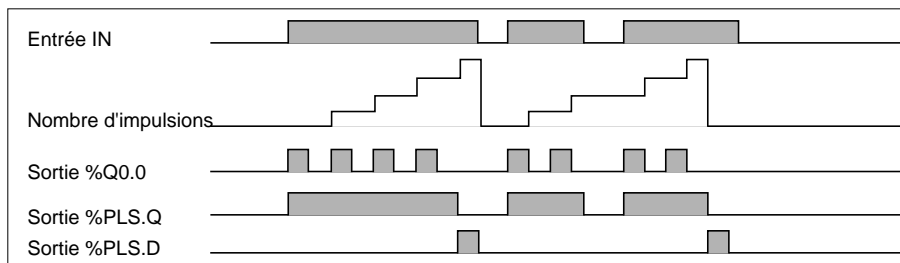
Caractéristiques

Base de temps	BT	0,1 ms (1), 10ms, 1s (valeur par défaut)
Présélection	%PLS.P	<p>0<%PLS.P<32767 si BT=10 ms ou 1s. 0<%PLS.P<255 si BT= 0,1ms (1)(2) (0= sortie à 0, 1=sortie à 1). La présélection permet de moduler la période du signal T = %PLS.PxBT.</p> <p>Gamme de période obtenue :</p> <ul style="list-style-type: none"> • 0,2 à 26 ms par pas de 0,1 ms, • 20 ms à 5,45 mn par pas de 10ms, • 2s à 9,1 heures par pas de 1s. <p>Nota: %PLS.P doit être un nombre pair</p>
Nombre d'impulsions	%PLS.N	<p>0≤%PLS.N≤32767, ce mot donne le nombre d'impulsions du train d'impulsion à générer. 0 = signal carré de durée illimitée (par défaut). %PLS.N est testé et écrit par programme.</p>
Réglage par terminal	O/N	<p>O : possibilité de modification de la valeur de présélection %PLS.P en mode réglage. N : pas d'accès en mode réglage.</p>
Entrée (ou instruction) génération d'impulsions	IN	Sur état 1, génère le signal sur la sortie %Q0.0. Sur état 0 met la sortie %Q0.0 à 0.
Entrée (ou instruction) Ré-initialisation	R	Sur état 1, remise à zéro du nombre d'impulsions, des sorties %PLS.Q et %PLS.D
Sortie génération d'impulsions en cours	%PLS.Q	Etat 1, génération du signal sur la sortie %Q0.0 en cours.
Sortie génération d'impulsions terminée	%PLS.D	Etat 1, génération du signal sur la sortie %Q0.0 terminée.
Comptage PLS (2)		N=non, O=où, cette option permet d'utiliser l'entrée %I0.0 comme entrée de comptage.

(1) Cette base de temps est déconseillée pour les TSX Nano à sorties relais.

(2) L'option comptage %PLS est obligatoire lorsque la base de temps de 0,1ms est sélectionnée. En fonctionnement en train d'impulsions, elle nécessite le rebouclage physique de la sortie %Q0.0 sur l'entrée %I0.0. Dans ce type de fonctionnement, le %PLS.P doit être supérieur ou égal à 6 (freq.max =1,6 KHz) pour garantir un bon fonctionnement de la fonction.

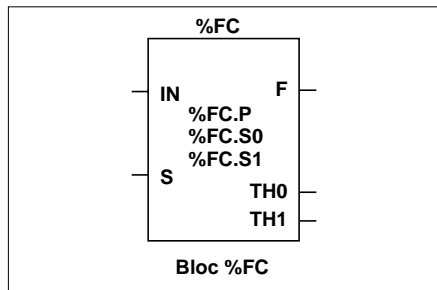
Fonctionnement



Cas spécifiques

- **Incidence d'une reprise "à froid"** : (%S0=1) provoque l'initialisation de la valeur de %PLS.P par celle définie en configuration
- **Incidence d'une "reprise à chaud"** (%S1=1) pas d'incidence
- **Incidence sur STOP automate** : voir Incidence sur Stop automate page 3/17.
- **Incidence d'une modification de la présélection %PLS.P** : la modification de %PLS.P par instruction ou en réglage est prise en compte instantanément.
- **Avec la base de temps 0,1 ms**, le forçage de l'entrée de rebouclage %I0.0 n'arrête pas la génération.

3.3-5 Fonctions comptage rapide, fréquencemètre et compteur/décompteur %FC



Le bloc fonction %FC est utilisé pour réaliser l'une des 3 fonctions suivantes :

- comptage rapide,
- fréquencemètre,
- comptage/décomptage.

(voir description détaillée, ch 4.4, intercalaire A). Le choix de la fonction s'effectue en configuration.

Note : les fonctions de comptage rapide peuvent être réalisées sans aucune programmation, simplement par configuration des entrées/sorties et des paramètres.

Caractéristiques du bloc %FC

Le bloc fonction offre un ensemble de mots, de bits d'entrées et de sorties permettant d'élaborer les 3 fonctions de comptage. Pour comprendre exactement le rôle de chaque objet dans la fonction à réaliser, se reporter à la description détaillée de la fonction.

Valeur courante	%FC.V	Mot incrémenté ou décrémenté en fonction des entrées et de la fonction sélectionnée. Peut être lu, testé mais non écrit.
Valeur de présélection	%FC.P	Utilisé uniquement par la fonction comptage/décomptage : $0 \leq \%FC.P \leq 65535$. Mot pouvant être lu, testé, écrit.
Valeur de seuil S0 (1)	%FC.S0	$0 \leq \%FC.S0 \leq 65535$ mot contenant la valeur du seuil 0, défini en configuration peut être lu et écrit par programme.
Valeur de seuil S1 (1)	%FC.S1	$0 \leq \%FC.S1 \leq 65535$ mot contenant la valeur du seuil 1, défini en configuration peut être lu et écrit par programme.
Entrée (ou instruction) validation	IN	Sur état 1, valide la fonction en cours. Sur état 0, inhibe la fonction en cours.
Entrée (ou instruction) présélection	S	Sur état 1: <ul style="list-style-type: none"> • initialise la valeur courante à la valeur de présélection (fonction comptage/décomptage) ou remet à zéro la valeur courante. • initialise le fonctionnement des sorties réflexes • prend en compte les valeurs de seuils %FC.S0 et %FC.S1 modifiées par programme.
Sortie débordement	%FC.F	Etat 1, lorsque la valeur courante %FC.V dépasse 65535. %FC.F peut être remis à 0 par l'entrée présélection (%I0.1 ou instruction S) ou sur reprise à froid.
Sortie seuil 0 (2)	%FC.TH0	Etat 1, lorsque la valeur courante est supérieure ou égale à la valeur de seuil %FC.S0
Sortie seuil 1 (2)	%FC.TH1	Etat 1, lorsque la valeur courante est supérieure ou égale à la valeur de seuil %FC.S1

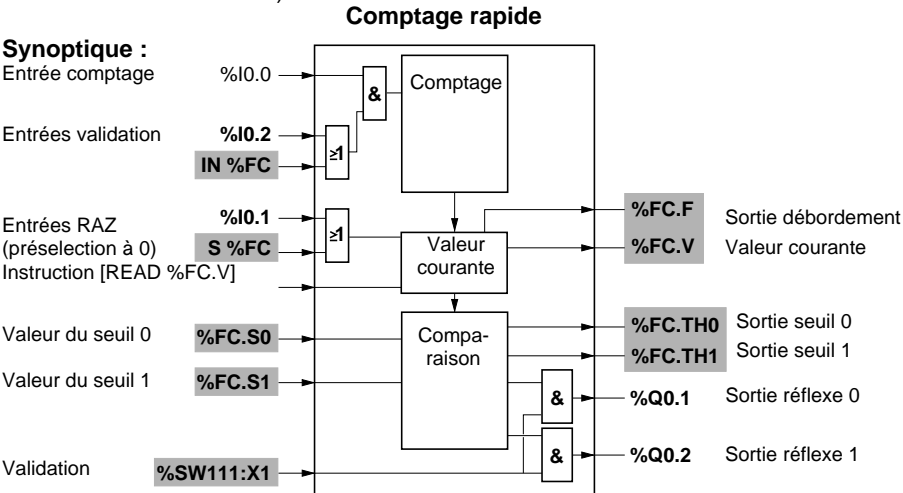
(1) La valeur de seuil %FC.S0 doit être inférieure à la valeur de seuil %FC.S1.

(2) Il est conseillé de ne tester les bits %FC.THx qu'une seule fois dans le programme.

Fonction comptage rapide

La fonction comptage rapide permet un comptage à une fréquence maximum de 10 kHz en mode rapide (ou 5 kHz en mode normal, choix en configuration), avec une capacité de comptage de 65535 points.

Le compteur reçoit les signaux à compter sur l'entrée automate %I0.0. La valeur de comptage (valeur courante %FC.V) est comparée à 2 seuils %FC.S0 et %FC.S1 définis en configuration, et modifiables par programme (modification prise en compte sur activation de l'entrée RAZ).



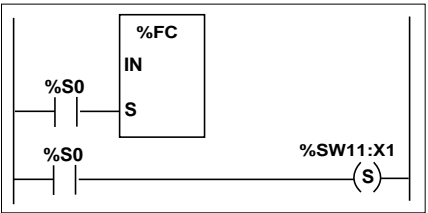
Remarque : Excepté l'entrée comptage %I0.0, les autres entrées/sorties tout ou rien du bloc fonction sont optionnelles (sélection ou pas en configuration), il existe pour chacune d'elles l'équivalent logiciel (repéré par une trame sur le synoptique).

Sorties réflexes : les sorties réflexes sont pilotées directement (n'attendent pas le rafraîchissement des sorties en fin de cycle) par le compteur rapide suivant la matrice d'état à définir en configuration.

Sortie	FC.V < seuil 0 < seuil 1	seuil 0 ≤ FC.V ≤ seuil 1	seuil 0 < seuil 1 < FC.V
%Q0.1	0 ou 1	0 ou 1	0 ou 1
%Q0.2	0 ou 1	0 ou 1	0 ou 1

A l'initialisation, le fonctionnement des sorties réflexes doit être validé impérativement par une commande de préselection du compteur rapide. Exemple de programmation :

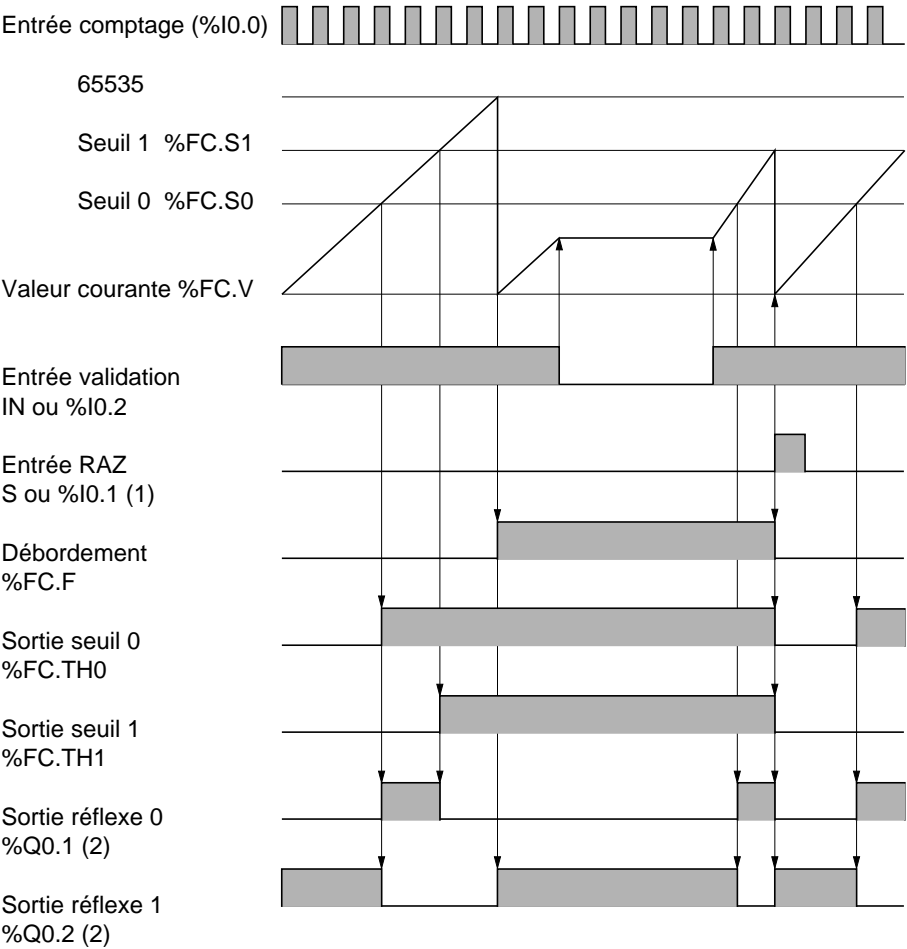
```
BLK %FC
LD %S0
S
END_BLK
LD %S0
S %SW111:X1 (Mise à 1 du bit
validation %SW111:X1)
```



Lecture de la valeur courante

La valeur courante %FC.V est mise à jour à chaque fin de cycle automate.%FC.V peut être également mise à jour par l'instruction READ :

syntaxe : [READ %FC.V] Chronogramme



(1) l'entrée %I0.1 fonctionne sur front montant (voir chronogramme), contrairement à l'entrée S qui fonctionne sur état.

(2) dans ce chronogramme la matrice d'état est la suivante :

Sortie	FC.V< seuil 0<seuil 1	seuil 0≤FC.V≤seuil 1	seuil 0<seuil 1<FC.V
%Q0.1	0	1	0
%Q0.2	1	0	0

Fonction Fréquencemètre

La fonction fréquencemètre permet de mesurer la fréquence en Hz d'un signal périodique sur l'entrée %I0.0. Deux modes rapide (filtrage 10kHz) ou normal (filtrage 5kHz) sont proposés.

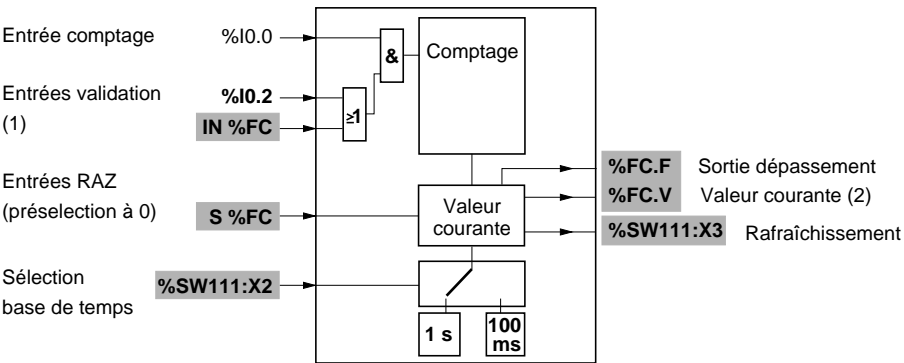
La gamme de fréquence pouvant être mesurée avec une précision admissible s'étend de 10Hz à 10kHz. L'utilisateur a le choix entre 2 bases de temps, la sélection s'effectuant par le bit système %SW111:X2 (1= base de temps de 100 ms, 0=base de temps de 1s).

Base de temps	Gamme de mesure	Précision	Rafraîchissement
100ms	100Hz-10kHz	0,1% pour 10kHz 10% pour 100Hz	10 fois par seconde
1s	10Hz-10kHz	0,01% pour 10kHz 10% pour 10Hz	1 fois par seconde

Le bit système %SW111:X3 est mis à 1 lors d'un rafraîchissement de la valeur courante, sa remise à zéro s'effectue par programme utilisateur.

Fréquencemètre

Synoptique :



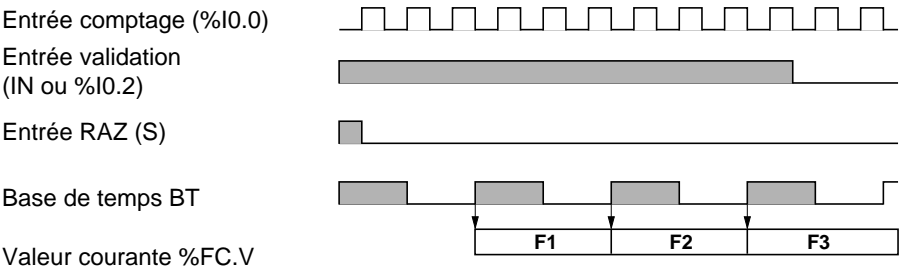
(1) L'entrée %I0.2 est optionnelle, le choix de son utilisation s'effectue en configuration.

(2) La valeur courante FC.V est exprimée en Hz.

Remarque:

Cette fonction permet également d'acquérir la valeur du module analogique câblé sur l'entrée %I0.0 (voir intercalaire B chapitre 4: gestion des E/S analogiques).

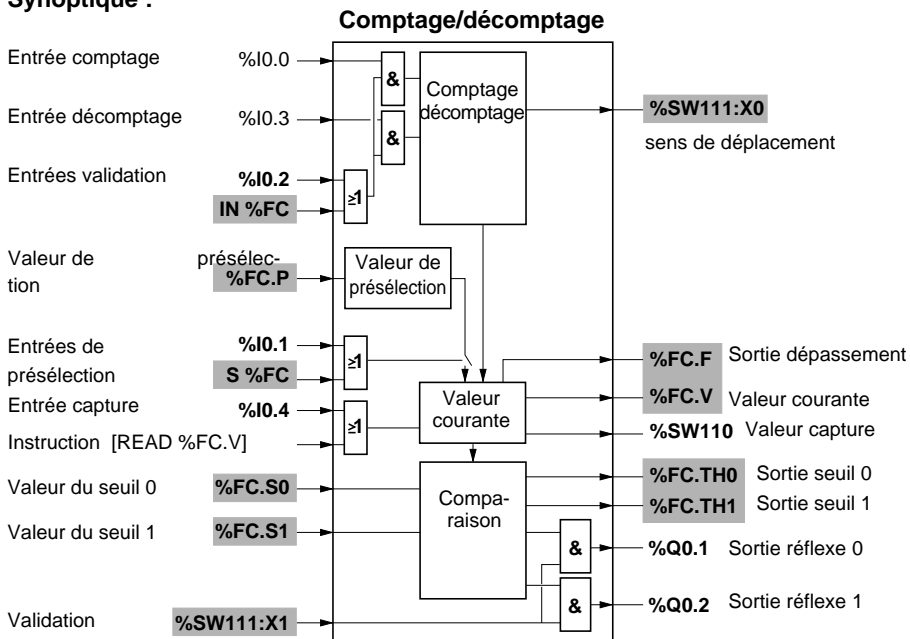
Chronogramme



Fonction comptage/décomptage

La fonction comptage/décomptage permet un comptage/décomptage à une fréquence maximum de 1 kHz, avec une plage de comptage/décomptage de 0 à 65535 points. Le compteur reçoit les signaux à compter sur l'entrée automate %I0.0 et les signaux à décompter sur l'entrée automate %I0.3, il fournit en sortie l'indication de sens de déplacement : si le compteur compte %SW111:X0 = 1 ou décompte %SW111:X0 = 0. La valeur de comptage/décomptage (valeur courante %FC.V) est comparée à 2 seuils %FC.S0 et %FC.S1 définis en configuration et modifiables par programme (modification prise en compte sur activation de l'entrée présélection).

Synoptique :



Excepté les entrées comptage %I0.0 et décomptage %I0.3, les autres entrées/sorties tout ou rien du bloc fonction sont optionnelles (sélection ou pas en configuration), il existe pour chacune d'elles l'équivalent logiciel (repéré par une trame sur le synoptique).

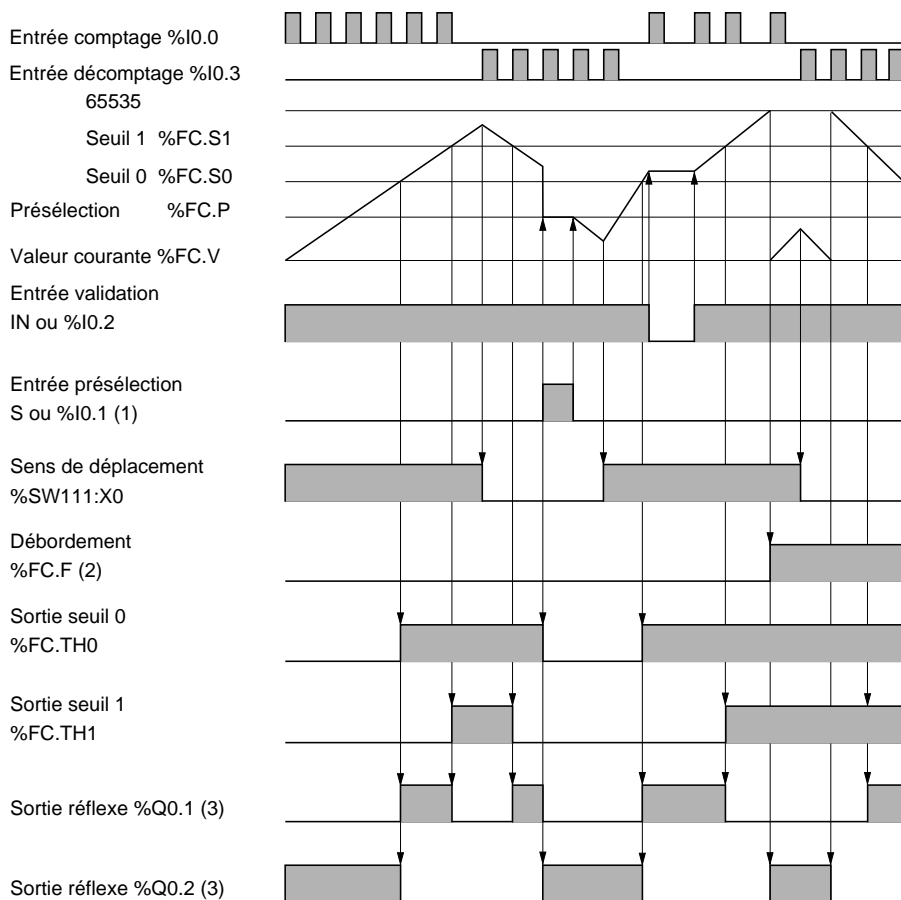
Présélection : la valeur de présélection de 0 à 65535 est définie en configuration et peut être modifiée par programme. La valeur courante est chargée par la valeur de présélection sur front montant sur l'entrée %I0.1 ou sur état 1 de l'entrée S.

Sorties réflexes : voir fonction comptage rapide.

Lecture de la valeur courante

La valeur courante %FC.V est mise à jour en fin de cycle. %FC.V peut être également mise à jour par l'instruction READ, syntaxe : [READ %FC.V]

Un front montant sur l'entrée capture %I0.4 écrit la valeur courante dans le mot système %SW110



- (1) l'entrée %I0.1 fonctionne sur front montant, contrairement à l'entrée S qui fonctionne sur état.
- (2) la sortie %FC.F débordement est remise à zéro après ré-initialisation du compteur
- (3) voir matrice d'état de la fonction comptage rapide.

Important :

Tant que l'une des 4 entrées sur front %I0.0, %I0.1, %I0.3, I0.4 est à l'état 1, aucune action liée à une des 3 autres entrées ne sera exécutée.

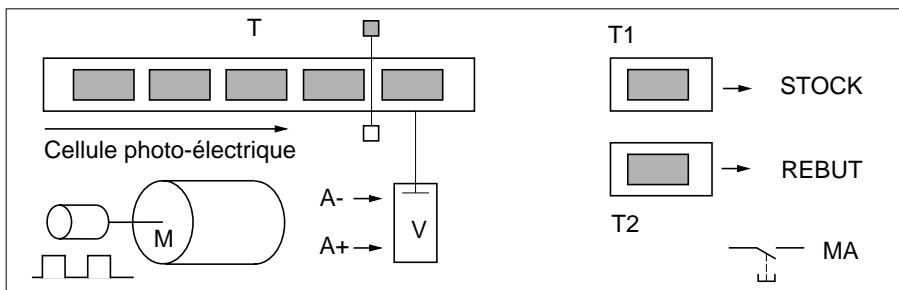
Cas spécifiques (comptage rapide, fréquencemètre et comptage/décomptage)

- **Incidence d'une reprise "à froid"** : (%S0=1) provoque la mise à 0 de la valeur courante, des sorties %FC.F, %FC.TH0, %FC.TH1, du bit validation des sorties réflexes (%SW111:X1) et la recopie des valeurs définies par configuration dans les mots %FC.P,%FC.S0/S1.
- **Incidence d'une "reprise à chaud"** (%S1=1) **et STOP automate** : n'a pas d'incidence sur la valeur courante.

Après une instruction relais maître MCS activée, les sorties réflexes sont mis à 0, après désactivation de MCS, elles ne sont repositionnées qu'après un dépassement de seuil en mode comptage ou une action de comptage/décomptage en mode décomptage.

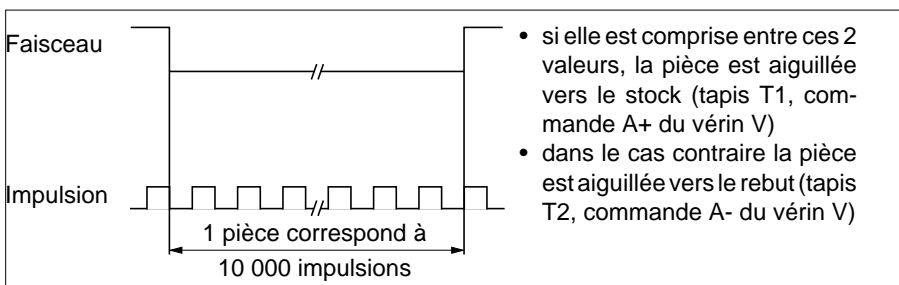
- **Visualisation des valeurs %FC :**
Pour toutes les valeurs issues de la fonction %FC, les valeurs sont visualisées:
 - par le terminal FTX 117 en valeur signée complément à 2 (Décimale) ou directe (Hexadécimale).
 - par PL7-07 en valeur directe(Décimale ou Hexadécimale)Par exemple :

%FC.V	Valeurs visualisées		
	(PL7-07)	(FTX 117)	(PL7-07 - FTX 117)
0	0	0	0000
1	1	1	0001
2	2	2	0002
⋮	⋮	⋮	⋮
•	•	•	•
32766	32766	32766	7FFE
32767	32767	32767	7FFF
32768		-32768	8000
⋮	⋮	⋮	⋮
•	•	•	•
65533	65533	-3	FFFD
65534	65534	-2	FFFE
65535	65535	-1	FFFF
0	0	0	0

Exemple fonction comptage rapide**Description de l'application**

Les pièces à mesurer sont amenées sur un tapis entraîné en permanence sans glisser par un moteur auquel est couplé un codeur incrémental rotatif. La mesure s'effectue en comptabilisant le nombre d'impulsions pendant le temps où la cellule C détecte le passage de la pièce. Un vérin V commande le déplacement latéral du tapis d'amenée T afin de la positionner en face du tapis T1 ou du tapis T2, selon le résultat de la mesure.

Le nombre d'impulsions mesuré est comparé à 2 valeurs extrêmes (tolérances de la mesure de longueur).



Le bouton poussoir MA assure la mise en marche de l'ensemble.

Affectation des entrées/sorties**Entrées**

- %I0.0 entrée comptage raccordée au codeur incrémental
- %I0.1 entrée remise à zéro raccordée à la cellule photoélectrique
- %I0.2 entrée validation raccordée au bouton marche

Sorties

- %Q0.1 sortie commande du vérin A+
- %Q0.2 sortie commande du vérin A-
- %Q0.0 sortie commande du tapis

Traitement de l'application

La fonction comptage rapide peut être traitée sans programmation de l'automate, uniquement par configuration du bloc fonction %FC.

%FC : comptage

Mode : rapide

Entrée comptage : %I0.0

Remise à zéro : %I0.1

Entrée validation : %I0.2

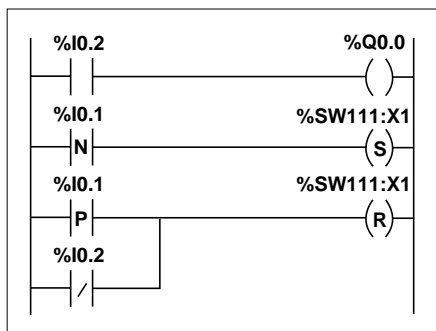
%FC.S0 : 9950 seuil 0 correspondant à la tolérance minimum

%FC.S1 : 10 000 seuil 1 correspondant à la tolérance maximum

Matrice des sorties

Sortie	FC.V < seuil 0 < seuil 1	seuil 0 ≤ FC.V ≤ seuil 1	seuil 1 < FC.V
%Q0.1	0	1	0
%Q0.2	1	0	1

Programme



```
LD    %I0.2
ST    %Q0.0
LDF   %I0.1
S     %SW111:X1
LDR   %I0.1
ORN   %I0.2
R     %SW111:X1
```

Fonctionnement

L'entrée %I0.0 compte le nombre d'impulsions issues du codeur incrémental dès que l'entrée validation %I0.2 (commutateur marche) est activée.

Sur front montant de l'entrée %I0.1, la valeur courante du compteur est remise à 0.

Lorsque la cellule (entrée %I0.1) détecte la fin de passage de la pièce sur le tapis, les sorties %Q0.1 et %Q0.2 sont validées (par le bit %SW111:X1) et prennent l'état 0 ou 1 en fonction de la valeur courante du compteur %FC.V à cet instant (selon matrice des sorties).

La sortie %Q0.1 est mise à 1 lorsque la pièce est dans les tolérances, elle commande la sortie du vérin A de positionnement sur le tapis T1 (voir matrice des sorties).

La sortie %Q0.2 est mise à 1 lorsque la pièce est hors tolérance (rentrée du vérin A vers le tapis T2).

3.3-6 Emission/Réception de message et contrôle des échanges

Le TSX Nano peut communiquer avec un terminal de programmation (FTX117 ou logiciel PL7-07) et avec d'autres équipements UNI-TELWAY connectés sur la prise terminal. Le TSX Nano peut également être configuré pour émettre et/ou recevoir un message en mode caractères (protocole ASCII).

Le langage PL7 offre pour cela :

- l'instruction d'émission / réception de message **EXCH**,
- le bloc fonction de contrôle des échanges **%MSG**.

L'automate TSX Nano détermine le protocole en fonction du brochage du câble utilisé, et fournit cette information dans le bit système %S100 (gestion du /DPT).

Les équipements supportant le protocole UNI-TELWAY et leur configuration sont détaillés dans l'intercalaire F du présent manuel.

Instruction EXCH

L'instruction EXCH permet au TSX Nano d'envoyer et/ou de recevoir des informations vers un équipement UNI-TELWAY ou ASCII. L'utilisateur définit une table de mots (%MWi:L ou %KW i:L) contenant les données à émettre et/ou à recevoir (64 mots de données maximum en émission et/ou en réception). Le format de la table de mots est décrit dans les paragraphes concernant chaque protocole (ASCII et UNITELWAY).

L'échange de messages est effectuée par l'instruction EXCH.

Syntaxe : [EXCH %MWi:L] (1) ou [EXCH %KW i:L]

Notes : Certains équipements supportant le protocole UNI-TELWAY tels que les équipements d'interfaces homme-machine (XBT ou CCX17) ainsi que les systèmes d'identification inductives peuvent communiquer avec un TSX Nano (envoi et/ou réception d'informations) sans programmation spécifique dans le TSX Nano.

Le TSX Nano doit avoir terminé l'échange d'une première instruction EXCH avant d'en lancer une seconde. Le bloc %MSG doit être utilisé lors de l'envoi de plusieurs messages.

(1) L : Nombre de mots de la table de mots.

Les valeurs de la table de mots internes %MWi:L sont telles que $i+L \leq 255$.

Remarque

Le logiciel PL7-07 V1 permettait de faire uniquement de l'émission, à l'aide de l'instruction SEND utilisée sur TSX Nano V1 et V2.

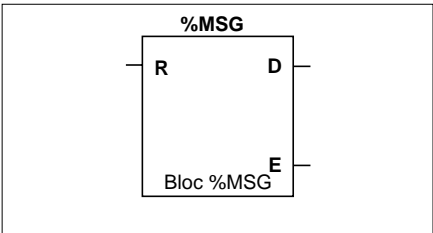
Les automates TSX Nano V3 autorisent l'émission et/ou la réception de messages avec l'instruction EXCH (Avec PL7-07 V1, l'instruction EXCH est appelée SEND).

Bloc fonction %MSG

Ce bloc fonction est utilisé pour gérer les échanges de données (son utilisation est optionnelle). Il dispose de trois fonctions :

- Contrôle d'erreur**
 Cette fonction vérifie que la longueur du tableau de mots de l'instruction EXCH est correctement dimensionnée pour contenir le message à envoyer (comparaison avec la longueur programmée dans l'octet de poids faible du premier mot du tableau). Elle vérifie également qu'un message UNI-TELWAY a été reçu.
- Coordination de l'envoi de plusieurs messages**
 Afin d'assurer la coordination lors de l'émission de plusieurs messages, ce bloc fonction fournit une information indiquant la fin de l'envoi du message précédant.
- Emission de message prioritaire**
 Le bloc fonction %MSG autorise l'arrêt de l'émission en cours afin de permettre l'envoi immédiat d'un message urgent.

Bloc de contrôle des échanges



Le bloc fonction **%MSG** permet de gérer les échanges.
Sa programmation est optionnelle.

Caractéristiques du bloc %MSG

Entrée (ou instruction)	R	Sur état 1, ré-initialise la communication, %MSG.E = 0 et %MSG.D =1.
Ré-initialisation		
Sortie liaison disponible	%MSG.D	Etat 1, liaison disponible si : <ul style="list-style-type: none"> • Fin d'émission si émission , • Fin de réception (caractère de fin reçu), • Erreur, • Ré-initialisation du bloc. Etat 0, demande en cours.
Sortie défaut	%MSG.E	Etat 1 si : <ul style="list-style-type: none"> • Mauvaise commande, • Table mal configurée • Mauvais caractère reçu (vitesse, parité, etc) • Table de réception pleine (non mise à jour) Etat 0, liaison OK.

En cas d'erreur lors de l'utilisation de l'instruction EXCH, les bits %MSG.D et %MSG.E passent à 1 et le mot système %SW69 contient le code d'erreur. Voir chapitre 6.2-2.

Entrée RESET (R) : La mise à 1 de cette entrée provoque l'arrêt immédiat de l'émission en cours, la remise à zéro de la sortie Error et la mise à 1 du bit Done. Un nouveau message peut alors être envoyé.

Sortie ERROR (%MSG.E) : Cette sortie est mise à 1 aussi bien sur une erreur de programmation que sur une erreur de transmission. Elle est également mise à 1 si le nombre d'octets de données définis dans la table de mots associée à l'instruction EXCH (octet de poids faible du premier mot) est supérieure à 128 (80 en hexadécimal).

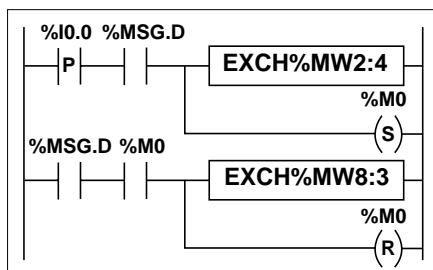
Cette sortie est mise à 1 si un problème est détecté pendant l'échange. Dans ce cas, l'utilisateur doit vérifier le câblage et si l'équipement destinataire supporte bien le protocole UNI-TELWAY.

Sortie DONE (%MSG.D) : Lorsque cette sortie est à 1, le TSX Nano est prêt à envoyer un nouveau message. L'utilisation de ce bit est recommandée lors de l'envoi de plusieurs messages. S'il n'est pas utilisé, des messages peuvent être perdus.

Envoi de plusieurs messages successifs

L'activation d'un bloc message dans le programme application est faite sur exécution de l'instruction EXCH. Le message est émis si le bloc message n'est pas déjà actif (%MSG.D=1). Si l'envoi de plusieurs messages est demandé dans un même cycle, seul le premier message est émis. C'est à l'utilisateur de gérer par programme la transmission de plusieurs messages.

Exemple 4 : envois successifs de 2 messages.

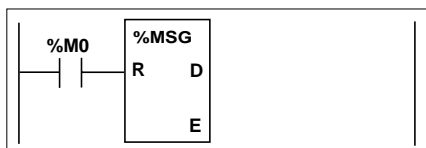


```
LDR %I0.0
AND %MSG.D
[EXCH %MW2:4]
S %M0
LD %MSG.D
AND %M0
[EXCH %MW8:3]
R %M0
```

Ré-initialisation d'échanges

L'annulation d'un échange est obtenue par activation de l'entrée (ou de l'instruction) R. Cette entrée initialise la communication et remet à 0 la sortie %MSG.E et à 1 la sortie %MSG.D. Il est possible de ré-initialiser un échange si un défaut est détecté.

Exemple :
 BLK %MSG
 LD %M0
 R
 END_BLK



Cas spécifiques

- **Incidence d'une reprise "à froid" :** (%S0=1) provoque la ré-initialisation de la communication,

- **Incidence d'une "reprise à chaud"** : (%S1=1) n'a pas d'incidence .
- **Incidence d'un passage en STOP** : si un message est en cours d'émission, l'automate termine son transfert et positionne les sorties des blocs %MSG.D et %MSG.E.

Caractéristiques du mode ASCII:

Le mode de communication chaîne de caractères (ASCII) est sélectionné en reliant plusieurs broches de la prise terminal (pour plus d'informations, se reporter dans l'intercalaire F du présent manuel). Le bit Status %S100 est mis à 1 quand le TSX Nano est en mode ASCII. Il confirme également que le câble est connecté.

Les trois utilisations possibles de cette instruction sont les suivantes:

- **Emission**
- **Emission / Réception**
- **Réception**

La taille maximum des trames émises et/ou reçues est de 128 octets.
 La table de mots associée à l'instruction EXCH est composée des tables d'émission et de réception.

ⁿ	Octet de poids fort	Octet de poids faible	
	Commande	Longueur LgE / LgR	Contrôle
^m	Octet émis 1	Octet émis 2	Table d'emission
	
	...	Octet émis n	
	Octet emis n+1		
	Octet reçu 1	Octet reçu 2	Table de réception
	
	...	Octet reçu p	
	Octet reçu p+1		

L'octet Longueur (LgE) contient la longueur à émettre, puis est écrasé par le nombre de caractères reçus (LgR) en fin de réception.

L'octet de commande doit contenir une des valeurs suivantes:

- 0: Emission
- 1: Emission suivie d'une réception
- 2: Réception

La table ne peut être de type %KWi que dans le cas d'une émission .

La réception s'arrête dès que l'octet de fin de trame est reçu (1). La valeur de cet octet est modifiable par l'utilisateur (poids faible du mot système %SW68). La valeur par défaut de ce mot est H'0D' (retour chariot).

(1) Attention:

Le système ne gère pas de time-out de réception

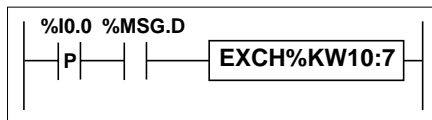
Envoi d'un message vers un équipement ASCII: Emission pure

Le contenu du tableau de mots associé à l'instruction EXCH nécessaire à l'envoi de données en ASCII (vers un écran vidéo, une imprimante, ...) est décrit ci-dessous :

Octet de poids fort	Octet de poids faible
0(émission)	Longueur du message (octets)
Données ASCII	

La longueur maximum du message est de 128 octets.

Exemple 1: afficher le message "DEFAULT 10" sur une imprimante utilisant le protocole ASCII.



```
LDR  %I0.0
AND  %MSG.D
[EXCH %KW10:7]
```

Contenu de la table de mots :

Mot	Contenu	Octet de poids fort	Octet de poids faible
%KW10	12	0	Longueur LgE en octets
%KW11	'DE'	Texte ASCII	
%KW12	'FA'		
%KW13	'UT'		
%KW14	'1'		
%KW15	'0'		
%KW16	16#0A0D	Saut de ligne	Retour chariot

Emission / Réception ASCII

Sur fin d'émission, le TSX Nano commute en attente de réception, puis dès réception de la réponse, la recopie dans la zone de %MWi contigue à la table d'émission si le status de la réception est OK et si la longueur de la question (LgE) et de la réponse (LgR) est inférieure à la zone de %MWi réservée (longueur L). Si ce n'est pas le cas, le bit %MSG.E passe à 1.

La fin de réception est effectuée sur détection de code de fin (H'0D' par défaut mais modifiable dans %SW68) ou sur table de réception pleine.

Il n'y a pas de gestion de time-out de réception.

Remarque

Le TSX Nano V1 ou V2 ne peut pas recevoir de message ASCII.

Le contenu du tableau de mots associé à l'instruction EXCH nécessaire à l'émission / réception de données en ASCII est décrit ci-dessous :

Octet de poids fort	Octet de poids faible	
1 (émission / réception)	Longueur LgE ou LgR	Contrôle
Octet à émettre 1	Octet à émettre 2	Table d'emission
...	...	
...	Octet à émettre n	
Octet à emettre n+1		
Octet reçu 1	Octet reçu 2	Table de réception
...	...	
...	Octet reçu p	
Octet reçu p+1	Code de fin (H'0D')	

Remarques

Les mots de type %KWi sont interdits.

Quand l'échange est fini (caractère de fin de trame reçu), l'octet longueur émission (LgE) contient le nombre de caractères reçus (LgR) .

Il faut donc remettre à jour l'octet longueur émission avant chaque échange.

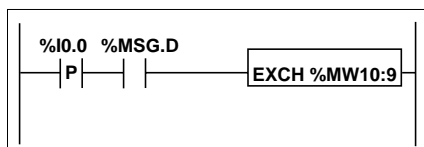
La zone de réception du message est toujours alignée sur le mot suivant la zone d'émission.

Exemple:

Mots	Poids fort	Poids faible
%MW10	16#0001	16#0007
%MW11	'V'	'A'
%MW12	'L'	'U'
%MW13	'E'	' '
%MW14	':'	non utilisé

Programme associé:

```
LDR  %I0.0
AND  %MSG.D
[EXCH %MW10:9]
```



Ce programme transmet la trame suivante: VALUE : soit 7 octets et attend de recevoir une réponse (8 octets maximum).

Les caractères reçus sont accessibles dans les mots %MW15 à %MW18.

Calcul de la longueur de la table de mots

LgE = 7

LgR = 8

$$L = 1 + \frac{LgE + LgE\%2}{2} + \frac{LgR + LgR\%2}{2} = 9$$

Remarque

Les Nano-automates TSX 07 2....., ne peuvent pas recevoir de message ASCII.

Réception d'un message d'un équipement ASCII

Sur l'exécution du bloc EXCH paramétré en réception, le TSX Nano commute en attente de réception et copie la réponse dans la zone de %MWi si le status de la réception est OK et si la longueur de la réponse (LgR) est inférieure à la zone de %MWi réservée (longueur L). Si ce n'est pas le cas, le bit %MSG.E passe à 1.

La fin de réception est effectuée sur détection de code de fin (16#0D par défaut mais modifiable dans %SW68) ou sur table de réception pleine.

Il n'y a pas de gestion de time-out de réception.

Remarque

Le TSX Nano V1 ou V2 ne peut pas recevoir de message ASCII.

Le contenu du tableau de mots associé à l'instruction EXCH nécessaire à la réception de données en ASCII est décrit ci-dessous :

Octet de poids fort	Octet de poids faible	
2 (réception)	0 (1)	Contrôle
Octet reçu 1	Octet reçu 2	Table de réception
...	...	
...	Octet reçu p	
Octet reçu p+1	Code de fin (H'0D')	

Remarque

Les mots de type %KWi sont interdits.

Le type de communication est géré dans le premier mot de la table.

(1) En mode réception, cet octet de poids faible n'est pas pris en compte.

Contrôle des échanges

Le contrôle des échanges est réalisé à l'aide du bloc fonction %MSG et du mot système %SW69.

Après chaque échange, le %SW69 (compte rendu du bloc EXCH) est mis à jour et prend une des valeurs suivantes:

- 0: Echange OK.
- 1: Table d'émission trop grande ($LgE > 128$).
- 2: Table d'émission trop petite ($LgE = 0$).
- 3: Table de mots trop petite (1).
- 7: Mauvaise commande ASCII (octet de commande $\neq 0, 1$ ou 2).
- 8: Non utilisé
- 9: Erreur de réception (problème de format de communication (vitesse, parité)).
- 10: Table %KWi interdite en réception ou émission / réception.

$$(1) L < 1 + \frac{LgE}{2} + \frac{LgE\%2}{2} + \frac{LgR}{2} + \frac{LgR\%2}{2}$$

avec L en mots

LgE et LgR en octets

Dialogue avec un équipement UNI-TELWAY

La table de mots associée à l'instruction EXCH utilisée pour envoyer une requête vers un équipement UNI-TELWAY tels que variateurs de vitesse ATV, équipements d'interface homme machine (CCX 17 ou XBT) est composée des tables d'émission et de réception.

Octet de poids fort	Octet de poids faible	
Adresse destinataire	Longueur LgE / LgR (octets)	Contrôle
Code catégorie	Code requête	Table d'emission
Premier mot (PF)	Premier mot (Pf)	
...	...	
Mot n-1 (PF)	Mot n-1 (Pf)	
Mot n (PF)	Mot n (Pf)	
00 (forcé)	Code réponse reçu	Table de réception
Donnée reçue 2	Donnée reçue 1	
...	...	
Donnée reçue p-1	...	
	Donnée reçue p	

La taille maximum des messages émis et reçus est de 128 octets.
La zone de réception du message est toujours alignée sur le mot suivant la zone d'émission.

Le bloc EXCH lit la longueur à émettre (LgE) dans le champs longueur émission.
Quand l'échange est fini, il y écrit celle du message reçu (LgR).

Il faut donc remettre à jour l'octet longueur émission avant chaque échange.

Le code réponse reçu est inscrit dans le poids faible du premier mot de la table de réception. Le poids fort de ce mot, est forcé à 0. Les éventuelles données suivantes sont alignées sur le prochain mot.

Important

A partir des TSX Nano V3, l'ordre d'émission des données d'une requête UNI-TE devient Poids Fort puis Poids faible.
Les applications fonctionnant avec des TSX Nano V1 ou V2 devront êtres modifiées pour tenir compte de cette inversion si elles sont chargées sur un TSX Nano V3.

Unitelway Maître

Dans ce mode, le TSX Nano gère normalement deux équipements répartis sur 5 adresses esclaves. Il est possible de piloter 2 équipements et un poste de programmation PL7 07, si le terminal de programmation est configuré sur une seule adresse.

Le TSX Nano ne gère pas le routage d'esclave à esclave.

Le TSX Nano Maître peut émettre une requête vers n'importe quel esclave d'adresse 1 à 5, à l'aide du bloc EXCH. Il utilise l'adresse source 0.254.16.

L'adresse destinataire codée dans la table de mots associée au bloc EXCH, doit être une des suivantes:

- 0: Emission d'une requête vers l'esclave 4 (compatibilité TSX07 2.).
- 1 à 5: Emission et réception d'une requête vers un esclave d'adresse 1 à 5.

Si l'adresse du destinataire vaut 0, les caractéristiques du bloc EXCH sont les suivantes:

- Le buffer de réception est inutile,
- La table peut être située en zone %KWi,
- Le bit %MSG.D passe à 1 quand la réponse de l'esclave est arrivée,
- La réponse de l'esclave est ignorée.
- Seules les requêtes Ecriture et Données non sollicitées peuvent être utilisées.

Si l'adresse du destinataire est comprise entre 1 et 5:

- La table de réception est obligatoire (1 mot minimum),
- La table de mots doit être située en zone %MWi,
- Le bit %MSG.D passe à 1 quand la réponse de l'esclave est arrivée,
- La réponse de l'esclave est recopiée dans la table de réception.

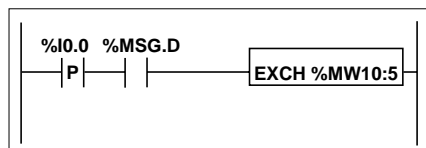
Exemple d'utilisation:

Emission de la requête "Lecture du mot" %MW513 (16#0201) vers un équipement situé à l'adresse 2:

Mots	Poids fort	Poids faible
%MW10	02	04
%MW11	07	04
%MW12	02	01

Programme associé:

```
LDR %I0.0
AND %MSG.D
[EXCH %MW10:5]
```



Le bloc EXCH utilise %MWi:L comme paramètres:

- i indique le numéro du premier mot de la table
- L indique le nombre de mots de la table de mots.

Une fois que le bit %MSG.D = 1 et que le bit %MSG.E = 0, la table contient les données suivantes:

Mots	Poids fort	Poids faible
%MW10	02	04
%MW11	07	04
%MW12	02	01
%MW13	00	34
%MW14	'AB'	'CD'

Les données en **gras** signifient :

- 4 octets reçus,
- code réponse reçu = 16#0034,
- valeur du mot %MW513 = 16#ABCD.

Unitelway Esclave

N'importe quel équipement (local ou distant) peut interroger le serveur système du TSX Nano Esclave en utilisant comme adresse destination Ad0 (serveur).

Un TSX Nano esclave peut émettre (client) une requête vers n'importe quel équipement Maître ou Esclave (adresse 0 à 98) à l'aide du bloc EXCH (lorsque le maître est un automate TSX 37/57).

L'adresse destinataire codée dans la table de mots associée au bloc EXCH, doit être comprise entre 100 et 198 (lorsque le maître est un automate TSX 47/67/87/107).

Les caractéristiques du bloc EXCH sont les suivantes:

- La table de réception est obligatoire (1 mot minimum),
- La table de mots doit être située en zone %MWi,
- Le bit %MSG.D passe à 1 quand la réponse de l'esclave est arrivée,
- La réponse est recopiée dans la table de réception.

Exemple d'utilisation:

Emission de la requête " Lecture du mot " %MW513 vers le Maître:

Mots	Poids fort	Poids faible
%MW10	00	04
%MW11	07	04
%MW12	02	01

Programme associé:

LDR %I0.0

AND %MSG.D

[EXCH %MW10:5]

Le bloc EXCH utilise %MWi:L comme paramètres:

- i indique le numéro du premier mot de la table
- L indique le nombre de mots de la table de mots.

Une fois que le bit %MSG.D = 1 et que le bit %MSG.E = 0, la table contient les données suivantes:

Mots	Poids fort	Poids faible
%MW10	00	04
%MW11	07	04
%MW12	02	01
%MW13	00	34
%MW14	'AB'	'CD'

Les données en **gras** signifient :

- 4 octets reçus,
- code réponse reçu = 16#0034,
- valeur du mot %MW513 = 16#ABCD.

Autre exemple:

Emission de la requête " Lecture du mot " %MW513 vers l'esclave 32:

Mots	Poids fort	Poids faible
%MW10	20	04
%MW11	07	04
%MW12	02	01

Programme associé:

```
LDR  %I0.0
AND  %MSG.D
[EXCH %MW10:5]
```

Le bloc EXCH utilise %MWi:L comme paramètres:

- i indique le numéro du premier mot de la table
- L indique le nombre de mots de la table de mots.

Une fois que le bit %MSG.D = 1 et que le bit %MSG.E = 0, la table contient les données suivantes:

Mots	Poids fort	Poids faible
%MW10	20	04
%MW11	07	04
%MW12	02	01
%MW13	00	34
%MW14	'AB'	'CD'

Les données en **gras** signifient :

- 4 octets reçus,
- code réponse reçu = 16#0034,
- valeur du mot %MW513 = 16#ABCD.

Contrôle des échanges

Le contrôle des échanges est réalisé à l'aide du bloc fonction %MSG et du mot système %SW69.

Le bit %MSG.D passe à 1 dans les cas suivants:

- En fin de réception de la réponse.
- En cas d'erreur de transmission (acquiescement négatif)
- En cas de ré-initialisation du bloc.
- Si la réponse n'arrive pas dans les 7 secondes (time-out applicatif).

Le bit %MSG.E passe à 1 dans les différents cas d'erreur (détaillés dans le mot %SW69):

Après chaque échange, le mot %SW69 (compte rendu du bloc EXCH) est mis à jour et prend une des valeurs suivantes:

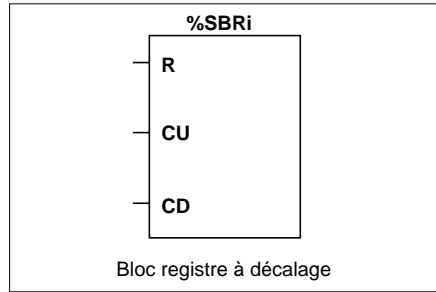
- 0: Echange OK.
- 1: Table d'émission trop grande ($LgE > 128$).
- 2: Table d'émission trop petite ($LgE = 0$).
- 3: Table de mots trop petite (1).
- 4: Mauvaise adresse Unitelway (l'adresse de destination n'appartient pas à [0...98] ou [100...198] en mode UNI-TELWAY Esclave ou à [1...5] en UNI-TELWAY Maître.
- 5: Time-out écoulé.
- 6: Erreur d'émission (le destinataire répond Nach).
- 7: Mauvaise commande ASCII (octet de commande $< > 0, 1$ ou 2).
- 8: Non utilisé
- 9: Erreur de réception (problème de format de communication (vitesse, parité)).
- 10: Table %KWi interdite en réception ou émission / réception.

$$(1) L < 1 + \frac{LgE + LgE\%2}{2} + \frac{LgR + LgR\%2}{2}$$

avec L en mots

LgE et LgR en octets

3.3-7 Blocs fonction registre à décalage bit %SBRi



Un registre à décalage bit permet de rentrer des informations binaires (0 ou 1) et de les faire évoluer dans un sens ou dans un autre.

Caractéristiques

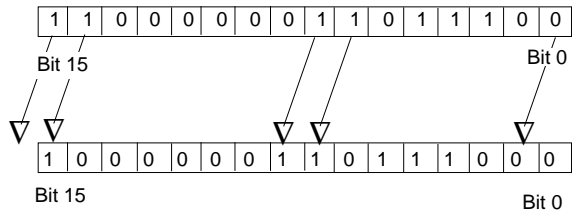
Numéro du registre	%SBRi	0 à 7
Bit du registre	%SBRi.j	bits 0 à 15 (j=0 à 15) du registre à décalage peut être testé par instruction de test et écrit par instruction d'affectation.
Entrée (ou instruction) mise à 0	R	Sur front montant, mise à 0 des bits %SBRi.j du registre.
Entrée (ou instruction) décalage à gauche	CU	Sur front montant, décalage à gauche d'un bit du registre.
Entrée (ou instruction) décalage à droite	CD	Sur front montant, décalage à droite d'un bit du registre.

Fonctionnement

Etat initial

CU %SBRi effectue le décalage à gauche

Le bit 15 est perdu

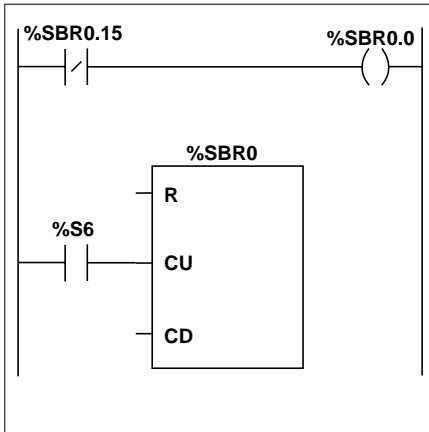


Il en est de même si l'on demande le décalage à droite d'un bit (du bit 15 vers le bit 0) par instruction CD. Le bit 0 est perdu.

Si l'utilisation d'un registre de 16 bits n'est pas suffisante, il est possible par programme de mettre plusieurs registres en cascade.

Programmation

Exemple : faire un décalage à gauche d'un bit toutes les secondes, le bit 0 prenant l'état inverse du bit 15.

**Programmation réversible**

```
LDN  %SBR0.15
ST   %SBR0.0
BLK  %SBR0
LD   %S6
CU
END_BLK
```

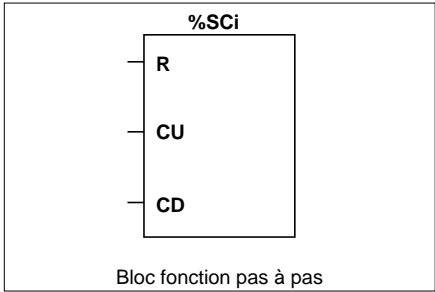
Programmation non réversible

```
LDN  %SBR0.15
ST   %SBR0.0
LD   %S6
CU   %SBR0
```

Cas spécifiques

- **Incidence d'une reprise "à froid" : (%S0=1)**
 - provoque la mise à 0 de tous les bits du mot registre,
- **Incidence d'une "reprise à chaud" : (%S1=1)** n'a pas d'incidence sur les bits du mot registre.

3.3-8 Blocs fonction pas à pas %SCi

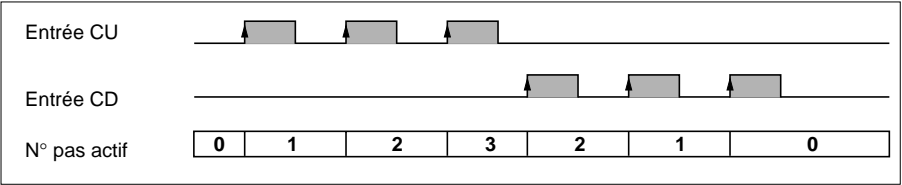


Un pas à pas est une suite de pas auxquels peuvent être associées des actions. Le passage d'un pas à un autre se fait en fonction d'événements externes ou internes. Chaque fois qu'un pas est actif, le bit associé est mis à 1. Un seul pas d'un pas à pas peut être actif.

Caractéristiques

Numéro du pas à pas	%SCi	0 à 7
Bit du pas à pas	%SCi.j	bits 0 à 255 (j=0 à 255) du pas à pas peut être testé par instruction de test LD et écrit par instruction d'affectation.
Entrée (ou instruction) mise à 0	R	Sur front montant, mise à 0 des bits %SCi.j du pas à pas.
Entrée (ou instruction) incrémentation	CU	Sur front montant, incrémentation d'un pas du pas à pas.
Entrée (ou instruction) décrémentation	CD	Sur front montant, décrémentation d'un pas du pas à pas.

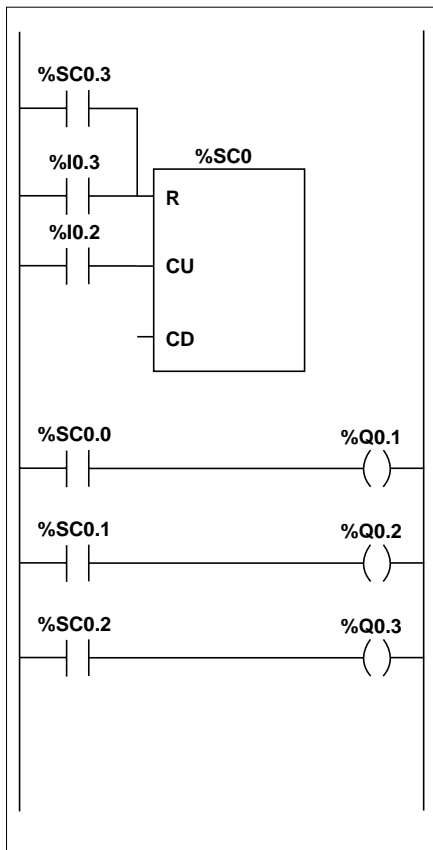
Fonctionnement



Programmation

Exemple : programmer le pas à pas 0 incrémenté par l'entrée %I0.2. Il est remis à 0 par l'entrée %I0.3 ou lorsqu'il arrive au pas 3.

Le pas 0 commande la sortie %Q0.1, le pas 1 la sortie %Q0.2 et le pas 2 la sortie %Q0.3.

**Programmation réversible**

```

BLK  %SC0
LD   %SC0.3
OR   %I0.3
R
LD   %I0.2
CU
END_BLK
LD   %SC0.0
ST   %Q0.1
LD   %SC0.1
ST   %Q0.2
LD   %SC0.2
ST   %Q0.3

```

Programmation non réversible

```

LD   %SC0.3
OR   %I0.3
R    %SC0
LD   %I0.2
CU   %SC0
LD   %SC0.0
ST   %Q0.1
LD   %SC0.1
ST   %Q0.2
LD   %SC0.2
ST   %Q0.3

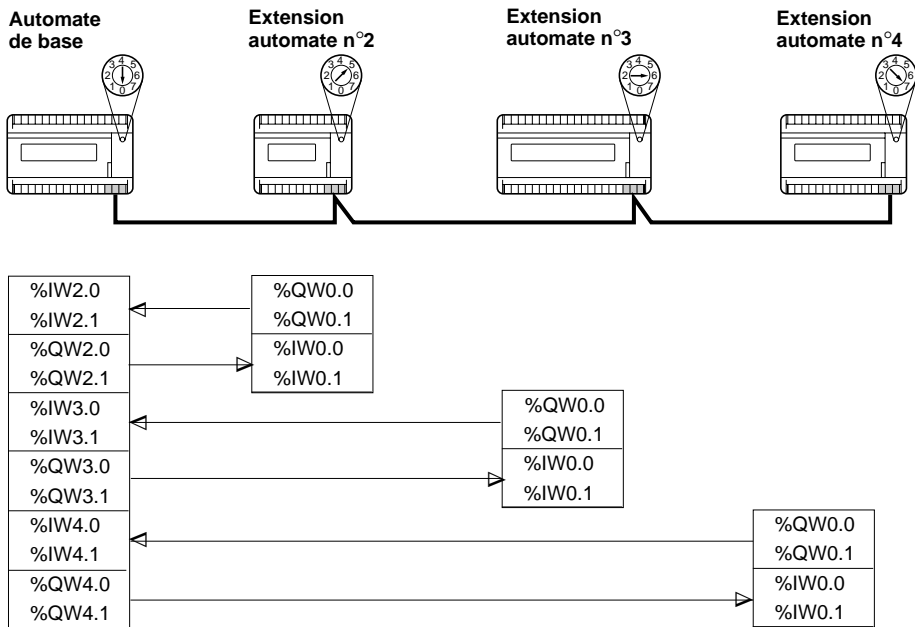
```

Cas spécifiques

- **Incidence d'une reprise "à froid" : (%S0=1)**
 - provoque l'initialisation du pas à pas,
- **Incidence d'une "reprise à chaud" : (%S1=1)** n'a pas d'incidence sur le pas à pas.

3.4 Communication inter-automates

Les mots %IW et %QW permettent l'échange d'informations entre automates. La figure ci-dessous montre pour chaque automate les mots échangés.



La mise à jour de ces mots d'échange s'effectue automatiquement lorsque les automates sont en exécution (RUN). Le programme utilisateur se limite pour chaque automate à :

- écrire dans les mots de sorties %QWi.j
- lire les mots d'entrées %IWi.j

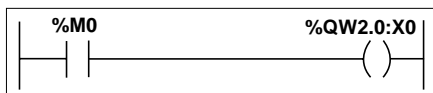
Le cycle de rafraîchissement des mots %IW/%QW est synchrone avec le cycle des automates. Le bit système %S70 est mis à 1 lorsqu'un cycle complet a eu lieu, sa remise à 0 s'effectue par programme ou terminal.

Les bits %S71 / %S72 et le mot %SW71 permettent aussi de contrôler les échanges (voir ch 5).

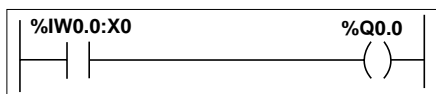
Note : l'adresse de chaque automate est définie en fonction de la position du sélecteur situé en face avant de l'automate, sa position est prise en compte à chaque mise sous tension.

Exemple 1 :

L'automate de base transmet à l'extension automate n°2 une information de type fin de fabrication (bit %M0=1). A la réception de cette information l'extension automate met en marche une machine de manutention par activation de la sortie %Q0.0.

**Programmation base automate**

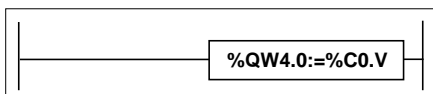
```
LD    %M0
ST    %QW2.0:X0
```

**Programmation extension automate**

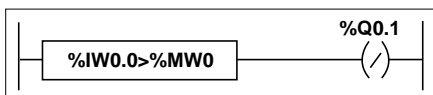
```
LD    %IW0.0:X0
ST    %Q0.0
```

Exemple 2 :

L'automate de base transmet à l'extension automate n°4 la valeur courante du compteur 0. Lorsque cette valeur courante est supérieure au seuil contenu dans le mot %MW0 l'extension automate met en arrêt une machine par désactivation de la sortie %Q0.1.

**Programmation base automate**

```
LD    1
[%QW4.0:=%C0.V]
```

**Programmation extension automate**

```
LD    [%IW0.0>%MW0]
STN   %Q0.1
```

4.1 Présentation

Les automates V3, permettent de gérer des modules d'entrée ou de sortie analogique.

- **Module d'entrée:** ces modules réalisent une conversion tension/fréquence, ce qui nécessite l'utilisation de l'entrée %I0.0 en mode fréquencemètre sur l'automate. Un seul module d'entrée est connectable par automate.
La gestion du module est réalisée depuis le programme application par l'intermédiaire des mots système %SW100 et %SW101.
- **Module de sortie:** ces modules réalisent une conversion PWM/tension, par intégration du signal délivré en sortie %Q0.0 de l'automate (en modulation de largeur d'impulsion). Un seul module est connectable par automate.
La gestion du module est réalisée depuis le programme application par l'intermédiaire des mots système %SW102 et %SW103.

4.2 Modules d'entrée analogique

4.2-1 Configuration des entrées analogiques

L'utilisation de l'entrée %I0.0 en mode fréquencemètre pour le raccordement du module d'entrée analogique nécessite le paramétrage suivant:

- **Compteur rapide:** à positionner sur Fréquence pour utilisation de l'entrée en fréquencemètre
- **Fréquence max. :** à positionner sur 10 KHz

4.2-2 Programmation des entrées analogiques

La programmation du module d'entrée s'effectue par l'utilisation des deux mots système %SW100 et %SW101 et en validant le compteur rapide %FC (voir exemple chapitre 4.2-4).

- **%SW100** : mot de commande des fonctions d'entrée analogique
- **%SW101** : valeur d'entrée analogique acquise

Le choix du mode de fonctionnement est réalisé en écrivant par programme le mot %SW100 et la valeur acquise de l'entrée analogique peut être lue dans le mot %SW101.

Ces deux mots sont remis à zéro par le système lors d'un démarrage à froid.

Le système offre par choix du mode de fonctionnement un service de mise à l'échelle. Cette mise à l'échelle est comprise dans une gamme de 0 à +10 000 pour les modules unipolaires (modules d'entrée 4/20mA et 0/10V) et -10 000 à +10 000 pour les modules bipolaires (-10/+10V).

%SW100	Fonctionnement	Gamme de valeur de %SW101
0	Invalidation du service entrée analogique sur %I0.0	0
1	Fonctionnement sans mise à l'échelle	0...1 000 Période de mesure de 125 ms
2	Mise à l'échelle pour gamme unipolaire (4/20mA, 0/10V)	0...10 000 Période de mesure de 125 ms
3	Mise à l'échelle pour gamme bipolaire (-10/+10V)	-10 000 ... +10 000 Période de mesure de 125 ms
4	Mise à l'échelle pour gamme unipolaire (4/20mA, 0/10V)	0...10 000 Période de mesure de 500 ms
5	Mise à l'échelle pour gamme bipolaire (-10/+10V)	-10 000 ... +10 000 Période de mesure de 500 ms

La valeur analogique brute ou mise à l'échelle est disponible dans %SW101 si %SW100 est écrit à une valeur de 1 à 5. La validité de cette mesure peut être contrôlée à l'aide du bit système %SW111:X3 (mis à 1 par le système si validité de la mesure).

Si mise à zéro du bit système %SW111:X3 par l'application, un service d'acquisition analogique est lancé, les acquisitions de la mesure continuent de se faire automate en STOP.

La mesure de fréquence brute reste disponible dans le mot %FC,V associé à l'entrée %I0.0 mais il s'agit de la mesure fonction de la période de mesure (ex: la pleine échelle 8KHz donnera 1000 pour 125ms et 4000 pour 500ms). Il est donc recommandé, pour simplifier l'applicatif, d'utiliser de préférence le mot système %SW101.

Remarque

La période de mesure peut être modifiée en cours de fonctionnement par réécriture du mot %SW100 mais ce mode d'utilisation n'est pas recommandé car la première mesure après changement de période peut être erronée.

Utilisation des modules d'entrée analogique sur les automates V2

Il est possible d'utiliser les modules d'entrée analogique sur des automates V2, TSX 07 2... , moyennant les règles suivantes:

- utilisation de l'entrée %I0.0 en fréquencesmètre (validation du fonctionnement par instruction IN %FC)
- configuration de la période de mesure par écriture par l'application du bit %SW111:X2.
 %SW111:X2=0 mesure toute les secondes (par défaut)
 %SW111:X2=1 mesure toute les 100 ms
- l'image de la valeur de l'entrée analogique est disponible dans l'objet %FC,V;
 l'interprétation de la valeur se réalise de la manière suivante:

Gamme	Formule
0/10V	$U(V) = 1,25 \times (\%FC,V \times 10^{-3})$
4/20 mA	$I(mA) = 2 \times [(\%FC,V \times 10^{-3}) + 2]$
-10/+10V	$U(V) = 2,5 \times [(\%FC,V \times 10^{-3}) - 4]$

Note

Dans le cas du module 4/20 mA, la fréquence est nulle entre 0 et 4 mA.

4.2-3 Temps de réponse des entrées analogiques

Le temps de réponse TRE en acquisition d'une entrée analogique, entre la variation effective de la grandeur électrique aux bornes du module et la correspondance dans le mot %SW101 de la valeur mesurée dépend essentiellement de la période de mesure choisie (125/500 ms) et dans une moindre proportion du temps de cycle de l'automate. La variation de la grandeur électrique intervient de façon négligeable dans ce temps de réponse.

- Pour une acquisition à la période de 125 ms: TRE est inférieur à 500 ms.
- Pour une acquisition à la période de 500 ms: TRE est inférieur à 1,2 s.

Note

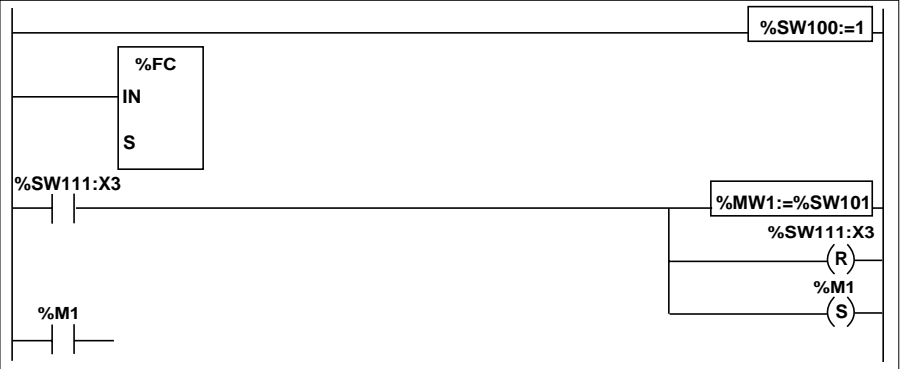
Un changement de mode de fonctionnement (changement de %SW100) est pris en compte à chaque cycle par le système ou immédiatement sur un front montant du "IN %FC". Les mesures sont enchaînées en temps réel les unes à la suite des autres à la période choisie (125 ms ou 500 ms). Le résultat de la dernière mesure effectuée est transmise en début de cycle automate dans %SW101. Ce mot ne change pas de valeur durant le cycle automate.

4.2-4 Exemple de programmation des entrées analogiques

```
(* VALIDATION SERVICES ENTREES ANALOGIQUES *)
LD      1          (* ENTREE ANA MODE 0..1000 SUR 125 MS *)
[%SW100 := 1]      (* PRISE EN COMPTE DU MODE CHOISI *)
BLK %FC
LD 1
IN
END_BLK

(* ACQUISITION MESURE *)
LD      %SW111:X3  (* MESURE VALIDE *)
[%MW1 := %SW101]   (* MEMO MESURE *)
R      %SW111:X3   (* ACQUITTEMENT MESURE *)
S      %M1         (* INDICE MESURE VALIDE *)

(* EXPLOITATION DE LA MESURE, SELON APPLICATION *)
LD      %M1
...
```



4.2-5 Caractéristiques des entrées analogiques

Type	Valeur d'entrée	Valeur %SW101 période 125ms	Résolution (1) /incrément	Valeur %SW101 période 500ms	Résolution(1) /incrément
4/20mA	4mA	0	16µA/10 lsb	0	4µA/2,5 lsb
	12mA	5000		5000	
	20mA	10000		10000	
0/10V	0V	0	10mV/10 lsb	0	2,5mV/ 2,5 lsb
	10V	10000		10000	
-10/+10V	-10V	-10000	20mV/10 lsb	-10000	5mV/ 2,5 lsb
	+10V	10000		10000	

Les valeurs de %SW101 sont données pour le mode de fonctionnement avec mise à l'échelle.
(1) Résolution: Valeur minimum de variation de l'entrée pour obtenir une variation de mesure. La variation de mesure varie par "pas" appelés incrément.

4.3 Modules de sortie analogique

4.3-1 Configuration des sorties analogiques

L'utilisation de la sortie %Q0.0 pour le raccordement du module de sortie analogique nécessite le paramétrage suivant:

- **Sortie %Q0.0 :** à utiliser en modulation de largeur d'impulsion %PWM.
- **Base de temps :** à positionner sur 0,1 ms.
- **Présélection :** à positionner impérativement à 249 pour que le fonctionnement reste valide après un démarrage à chaud.

4.3-2 Programmation des sorties analogiques

La programmation du module de sortie s'effectue par l'utilisation des deux mots système %SW102 et %SW103 et en validant la sortie %PWM (voir exemple chapitre 4.3-4).

- **%SW102 :** mot de commande/status des fonctions de sortie analogique
- **%SW103 :** valeur de sortie analogique à générer

Le choix du mode de fonctionnement est réalisé en écrivant par l'applicatif le mot %SW102 et la valeur à générer sur la sortie analogique doit être écrite dans le mot %SW103.

Ces deux mots sont remis à zéro par le système lors d'un démarrage à froid.

Le système offre par choix du mode de fonctionnement un service de mise à l'échelle. Cette mise à l'échelle est comprise dans une gamme de 0 à +10 000 pour les modules unipolaires (modules d'entrée 4/20mA et 0/10V) et -10 000 à +10 000 pour les modules bipolaires (-10/+10V).

%SW102	Fonctionnement	Gamme de valeur de %SW103
0	Invalidation du service sortie analogique sur %Q0.0	Non utilisé
1	Fonctionnement sans mise à l'échelle	5...249
2	Mise à l'échelle pour gamme unipolaire (4/20mA, 0/10V)	0...10 000
3	Mise à l'échelle pour gamme bipolaire (-10/+10V)	-10 000 ... +10 000

La résolution effective des sorties analogiques est de 245 points.

Dans le cas où la valeur écrite dans %SW103 est inférieure à la valeur minimale (ex: inférieure à 0 en mode unipolaire), c'est la valeur minimale de la gamme qui sera appliquée au module de sortie.

Dans le cas où la valeur écrite dans %SW103 est supérieure à la valeur maximale (ex: supérieure à 10000 en mode unipolaire), c'est la valeur maximale de la gamme qui sera appliquée au module de sortie.

Ces deux types d'erreur de programmation ne sont pas signalés à l'application.

Important

Dans les conditions de repli des sorties TOR, le PWM n'est plus généré et les modules de sorties ne sont plus attaqués par un signal.

De ce fait, les modules bipolaires prennent leur valeur la plus basse (-10V).

Ce mode de repli doit être pris en compte par l'utilisateur.

4.3-3 Temps de réponse des sorties analogiques

Le temps de réponse TRS d'une sortie analogique, entre l'écriture de la consigne dans le mot %SW103 et l'atteinte de la tension (et/ou courant) correspondant aux bornes du module, dépend de l'amplitude de la variation et du temps de cycle de l'automate.

- Pour une variation de la pleine échelle, TRS est inférieur à 500 ms.

Ce temps sera d'autant plus court que le temps de cycle automate sera court et la variation de consigne faible. Pour un temps de cycle de 10 ms et une variation de 1/10^{ème} de la pleine échelle, ce temps de réponse descendra à environ 50 ms.

Note

Un changement de mode de fonctionnement (changement de %SW102) est pris en compte sur front montant de l'entrée IN du PWM (exécution de l'instruction "IN %PWM") ou sur changement de la consigne %SW103. Un changement de consigne (%SW103) est pris en compte à chaque cycle par le système et devient effectif dès le cycle applicatif suivant (délai Max de 3ms).

4.3-4 Exemple de programmation des sorties analogiques

(* VALIDATION SERVICES SORTIES ANALOGIQUES *)
LD 1
[%SW102 := 1] (* SORTIE ANA MODE BRUT 5..249 *)
IN %PWM (* PRISE EN COMPTE DU MODE *)

(* GENERATION CONSIGNE EN PWM *)
.... (* CALCUL CONSIGNE SELON APPLICATION *)
LD 1
[%SW103 := %MW0] (* APPLICATION CONSIGNE A SORTIE *)

%PWM
IN
S

.....

%SW102:=1

%SW103:=%MW0

4.3-5 Caractéristiques des sorties analogiques

Type	Valeur à générer en sortie	Valeur %SW103 à écrire par applicatif	Résolution module de sortie	Résolution lsb (1) %SW103
4/20mA	4mA	0	65µA	40
	12mA	5000		
	20mA	10000		
0/10V	0V	0	40mV	40
	5V	5000		
	10V	10000		
-10/+10V	-10V	-10000	81mV	81
	0V	0		
	+10V	10000		

Les valeurs de %SW103 sont données pour le mode de fonctionnement avec mise à l'échelle.
(1) Résolution LSB: Variation minimum à appliquer à %SW103 pour obtenir une variation du module de sortie égale à sa résolution.

5.1 Présentation

Les automates TSX Nano (16 E/S et 24 E/S) possèdent une horloge, à partir de laquelle peuvent être élaborées trois fonctions :

- **Programmateurs Temporels** qui permet de commander des actions à des horaires prédéfinis ou calculés,
- **Consignateurs Temporels** qui permet la datation d'événements et **mesure de durée**,

La mise à l'heure de l'horodateur du TSX Nano s'effectue soit en mode Configuration, soit par programme. Son fonctionnement, lorsque l'automate est hors tension, est assuré pendant 30 jours si la batterie a été chargée pendant au moins 6 heures consécutives avant l'arrêt de l'automate.

L'horloge a un format 24 heures, elle gère les années bissextiles.

5.2 Programmeur temporel

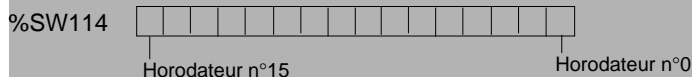
Le programmeur temporel permet de commander des actions à des horaires et des dates prédéfinis ou calculés.

Jusqu'à 16 blocs horodateurs peuvent être utilisés réalisant chacun cette fonction. Ces blocs ne nécessitent aucune saisie de programme, ils sont configurables (voir modes opératoires, intercalaire C).

5.2-1 Caractéristiques

Numéro du Bloc Horodateur	RTC : n	n=0 à 15
Sortie	Q:	Affectation de la sortie activée par l'horodateur: %Mi ou %Qj.k. Cette sortie est mise à l'état 1 lorsque la date et l'heure courante sont comprises entre les consignes de début de période active et les consignes de fin de période active.
Date début	JJ:MMM	mentionne le jour de 1 à 31 et le mois jan,...,dec de début de validation de l'horodateur.
Date fin	JJ:MMM	mentionne le jour de 1 à 31 et le mois jan,...,dec de fin de validation de l'horodateur
Jour	LMMJVSD	mentionne les jours d'activation dans la semaine (L: Lundi,..., D: Dimanche)
Heure début	hh:mm	mentionne en heure (0 à 23) et minute (0 à 59) le début d'activation de l'horodateur.
Heure fin	hh:mm	mentionne en heure (0 à 23) et minute (0 à 59) la fin d'activation de l'horodateur.

Le mot système %SW114 permet de valider par ses bits (à 1) ou inhiber (à 0) le fonctionnement de chacun des blocs..



Ce mot système à tous ses bit à 1 par défaut (ou après reprise à froid), **sa gestion par programme est optionnelle.**

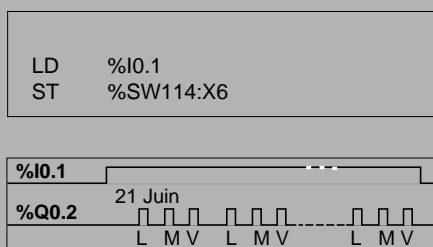
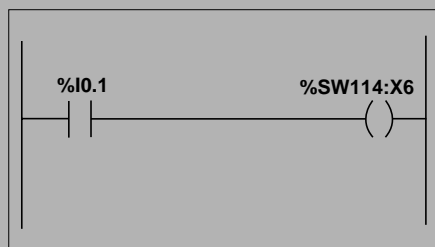
Remarques :

- Si la même sortie (%Mi ou %Qj.k) est affectée par plusieurs blocs, c'est le "OU" des résultats de chacun des blocs qui est finalement affecté à cet objet (permet d'avoir plusieurs "plages de fonctionnement" pour une même sortie).

Exemple : configuration d'un horodateur, arrosage programmé pour les mois d'été.

- **RTC 6** : horodateur n°6,
- **Q : %Q0.2** : sortie activée par l'horodateur
- **21 -Jui -> 21-Sep** : période de validation
- **L•M•V••** : jours de validation (Lundi, Mercredi et Vendredi),
- **21 : 00 - 22 : 00** : plage horaire d'activation

RTC:6 Q: %Q0.2
21-Jul --> 21-Sep
L•M•V••
21 : 00 - 22 : 00



Dans cet exemple, l'utilisateur peut inhiber l'horodateur par un interrupteur ou par un détecteur d'humidité câblé sur l'entrée %I0.1.

Nota:

Il est important de contrôler l'état du bit %S51 qui indique tout défaut de l'horodateur.

5.2-2 Horodatage par programme

Date et heure sont également disponibles dans les mots système %SW50 à %SW53 (voir ch 5.2), il est donc possible de faire de l'horodatage par programme automate en faisant des comparaisons arithmétiques entre la date et heure courante et des valeurs immédiates ou des mots %MWi (ou %KWi) pouvant contenir des consignes.

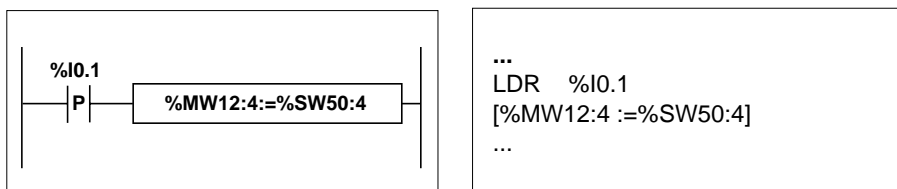
5.3 Consignateur temporel

La fonction consignateur temporel permet de mémoriser la date et heure d'apparition d'un événement.

Les mots système %SW50 à %SW53 (voir ch 6.2) contiennent la date et heure courante, en format BCD, format utile pour l'affichage ou pour l'envoi vers un périphérique.

Pour dater un événement, il suffit d'utiliser les opérations d'affectation, pour transférer le contenu des mots système dans des mots internes et ensuite traiter ces mots internes (par exemple : envoi par instruction EXCH à des afficheurs).

Exemple :



Une fois l'événement détecté, le tableau de mots contient :

Codage :	Octet poids fort	Octet poids faible
%MW12	Seconde	Jour de la semaine(1)
%MW13	Heure	Minute
%MW14	Mois	Jour
%MW15	Siècle	année

Exemple : lundi 19 avril 1994
 En hexa 13H, 40mn, 30s
 3000 30s, 0=Lundi
 1340 13H, 40mn
 0419 4=Avril, 19
 1994 1994

(1) avec 0=Lundi, 1=Mardi, 2=Mercredi, 3=Jeudi, 4=Vendredi, 5=Samedi, 6=Dimanche

Lecture de la date et l'heure du dernier arrêt par mots système

Les mots système %SW54 à %SW57 (voir ch 5.2) contiennent la date et heure du dernier arrêt et le mot %SW58 contient le code mentionnant la cause du dernier arrêt, au format BCD.

5.4 Mise à l'heure de l'horodateur

5.4-1 Mise à jour de la date et heure par terminal

Le mode TSX du terminal de programmation permet un accès simple et rapide à la mise à jour de la date et de l'heure (voir modes opératoires, intercalaire C).

5.4-2 Mise à jour de la date et heure par mots système

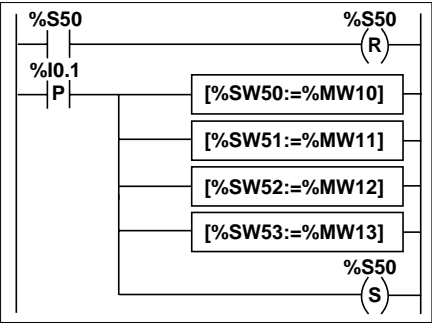
Les mots système offrent 2 autres possibilités de mise à jour de la date et de l'heure :

Mise à jour par les mots système %SW50 à %SW53

(voir ch 6.2)

Pour cela, le bit %S50 doit être mis à 1. Ce bit :

- annule le rafraîchissement des mots %SW50 à %SW53 par l'horloge interne,
- transmet les valeurs écrites dans les mots %SW50 à %SW53 à l'horloge interne.



```
LD    %S50
R      %S50
LDR   %I0.1
[%SW50:=%MW10]
[%SW51:=%MW11]
[%SW52:=%MW12]
[%SW53:=%MW13]
S      %S50
```

Les mots %MW10 à %MW13 doivent contenir la nouvelle date et heure au format BCD et doivent correspondre au codage des mots %SW50 à 53.

Le tableau de mots doit contenir les nouvelles date et heure.

Codage :	Octet poids fort	Octet poids faible
%MW10	Seconde	Jour de la semaine(1)
%MW11	Heure	Minute
%MW12	Mois	Jour
%MW13	Siècle	année

Exemple : Lundi 19 avril 1994
Hexa 13H, 40mn, 30s
3000 30s, 0=Lundi
1340 13H, 40mn
0419 4=Avril, 19
1994 1994

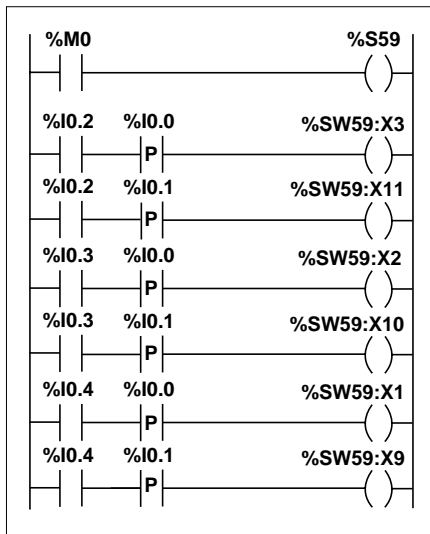
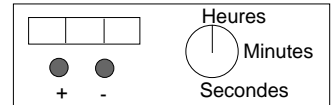
(1) avec 0=Lundi, 1=Mardi, 2=Mercredi, 3=Jeudi, 4=Vendredi, 5=Samedi, 6=Dimanche

Mise à jour par le mot système %SW59

Une autre possibilité de mise à jour est fournie par le bit de validation %S59 et le mot de réglage %SW59.

La mise à 1 du bit %S59 assure la validation du réglage de la date et heure courante par le mot %SW59. Le mot %SW59 est décrit ch 5.2, il permet l'incréméntation ou la décrémentation de chacun des composants de la date et heure sur front montant.

Exemple : une face avant est réalisée pour pouvoir modifier l'heure, les minutes et les secondes de l'horloge interne.



```

LD    %M0
ST    %S59
LD    %I0.2                (heure)
ANDR  %I0.0
ST    %SW59:X3
LD    %I0.2
ANDR  %I0.1
ST    %SW59:X11
LD    %I0.3                (minute)
ANDR  %I0.0
ST    %SW59:X2
LD    %I0.3
ANDR  %I0.1
ST    %SW59:X10
LD    %I0.4                (seconde)
ANDR  %I0.0
ST    %SW59:X1
LD    %I0.4
ANDR  %I0.1
ST    %SW59:X9

```

- les entrées %I0.2, %I0.3 et %I0.4 sont pilotées par le commutateur Heures/Minutes/Secondes,
- l'incréméntation est assurée par l'entrée %I0.0, bouton poussoir +,
- la décrémentation est assurée par l'entrée %I0.1, bouton poussoir -.

6.1 Bits système

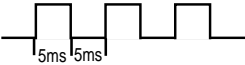
6.1-1 Liste des bits système

Bit	Fonction	Etat init.	Gestion
%S0	1 = démarrage à froid (reprise secteur avec perte des données)	0	S ou U->S
%S1	1 = démarrage à chaud (reprise secteur sans perte de données)	0	S ou U->S
%S4, %S5	Base de temps 10ms, 100ms	-	S
%S6, %S7	Base de temps 1s, 1mn	-	S
%S8	0 = maintien des sorties sur STOP automate	1	U
%S9	1 = mise à zéro des sorties automate en RUN automate	0	U
%S10	0 = défaut entrées/sorties	1	S
%S11	1 = débordement chien de garde	-	S
%S13	1 = premier cycle après mise en RUN	1	S
%S17	1 = débordement sur calcul non signé ou décalage circulaire	0	S->U
%S18	1 = débordement ou erreur arithmétique	0	S->U
%S19	1 = débordement de la période de scrutation	0	S->U
%S20	1 = débordement d'index	0	S->U
%S21	1 = initialisation du Grafcet provoque la mise à 0 des étapes et la mise à 1 des étapes initiales	0	U->S
%S22	1 = remise à zéro Grafcet	0	U->S
%S23	1 = validation du prépositionnement du GRAFCET, maintenu à 1 provoque le figeage du GRAFCET	0	U->S
%S49	1 = demande de réarmement toutes les 10s des sorties statiques déclenchées sur surintensité ou court-circuit	0	U
%S50	1 = mise à l'heure de l'horodateur	0	U
%S51	1 = horodateur non initialisé ou en défaut 0 = date et heure à jour	0	S
%S59	1 = réglage de la date courante	0	U
%S69	1 = visualisation des bits internes	0	U
%S70	1 = rafraîchissement échange %IW/%QW sur extension. Traitement requête Modbus.	0	S
%S71	1 = échange sur liaison d'extension	0	S
%S72	0 = scrutation des automates extension	0	U
%S100	Etat du /DPT	-	S
%S118	1 = défaut automate de base	0	S
%S119	1 = défaut extension d'entrées/sorties	0	S

S = géré par le système, U = géré par l'utilisateur, U->S = mis à 1 par l'utilisateur, remis à 0 par le système, S->U = mis à 1 par le système, remis à 0 par l'utilisateur.

6.1-2 Description détaillée des bits système

Les automates TSX Nano disposent de bits système %Si qui indiquent les états de l'automate ou permettent d'agir sur le fonctionnement de celui-ci. 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 système	Fonction	Désignation
%S0	Reprise à froid	<p>Normalement à l'état 0, est mis à l'état 1 par :</p> <ul style="list-style-type: none"> reprise secteur avec perte des données (défaut batterie) programme utilisateur, terminal (mode Réglage), <p>Ce bit est mis à 1 durant le premier cycle complet. Il est remis à 0 avant le cycle suivant.</p> <p>Fonctionnement : voir chapitre 7.1 intercalaire A.</p>
%S1	Reprise à chaud	<p>Normalement à l'état 0, est mis à l'état 1 par :</p> <ul style="list-style-type: none"> reprise secteur avec sauvegarde des données, programme utilisateur, terminal (mode Réglage). <p>Il est remis à 0 par le système à la fin du premier cycle complet et avant la mise à jour des sorties.</p> <p>Fonctionnement : voir chapitre 7.1 intercalaire A .</p>
%S4 %S5 %S6 %S7	Bases de temps 10ms 100ms 1s 1min	<p>Bits dont le changement d'état est cadencé par une horloge interne. Ils sont asynchrones par rapport au cycle de l'automate.</p> <p>Exemple : %S4</p> 

Bits système	Fonction	Désignation
%S8	Sécurité des sorties	Initialement à l'état 1, peut être mis à l'état 0 par programme ou par le terminal (mode REGLAGE) : <ul style="list-style-type: none"> état 1 : provoque la mise à zéro des sorties de l'automate, en cas de non exécution normale du programme ou sur STOP automate, état 0 : maintient les sorties dans l'état défini en cas de non exécution normale du programme ou sur STOP automate.
%S9	Mise à zéro des sorties	Normalement à l'état 0. Peut être mis à l'état 1 par programme ou par le terminal (mode REGLAGE) : <ul style="list-style-type: none"> état 1 : provoque le forçage à l'état 0 des sorties de l'automate en RUN, 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'entrées/sorties de l'automate de base ou de l'automate extension (configuration non conforme, défaut d'échange, défaut matériel, disjonction des sorties statiques protégées) est détecté. Les bits %S118 et %S119 indiquent l'automate en défaut et les mots %SW118 et %SW119 précisent la nature du défaut (voir ch 5.2) Le bit %S10 est remis à 1 dès la disparition du défaut.
%S11	Débordement du chien de garde	Normalement à l'état 0, est mis à l'état 1 par le système dès que le temps d'exécution du programme devient supérieur au temps de cycle maximum (chien de garde logiciel). Le débordement du chien de garde provoque le passage en STOP de l'automate.
%S13	Premier cycle	Normalement à l'état 0, est mis à l'état 1 par le système durant le premier cycle après la mise en RUN automate.
%S17	Dépassement de capacité (carry)	Normalement à l'état 0, est mis à l'état 1 par le système : <ul style="list-style-type: none"> En cas de dépassement de capacité lors d'une opération en arithmétique non signé (retenue). Lors d'un décalage circulaire, signale la sortie d'un bit à 1. Doit être testé, par le programme utilisateur, après chaque opération où il y a risque de dépassement puis remis à 0 par l'utilisateur en cas de dépassement.

Bits système	Fonction	Désignation
%S18	Débordement ou erreur arithmétique "Overflow"	<p>Normalement à l'état 0. Est mis à l'état 1 en cas de débordement de capacité lors d'une opération sur 16 bits soit :</p> <ul style="list-style-type: none"> • résultat supérieur à + 32767 ou inférieur à - 32768, • division par 0. • racine carrée d'un nombre négatif • conversion BTI ou ITB non significative (valeur BCD hors bornes) <p>Doit être testé, par le programme utilisateur, après chaque opération où il y a risque de débordement puis remis à 0 par l'utilisateur en cas de débordement.</p>
%S19	Débordement période de scrutation (scrutation périodique)	<p>Normalement à l'état 0, ce bit est mis à l'état 1 par le système en cas de dépassement de la période d'exécution (temps d'exécution de la tâche supérieur à la période définie par l'utilisateur en configuration ou programmé dans %SW0).</p> <p>Ce bit est remis à l'état 0 par l'utilisateur.</p>
%S20	Débordement d' index	<p>Normalement à l'état 0, est mis à l'état 1 lorsque l'adresse de l'objet indexé devient inférieure à 0 ou supérieure à 255.</p> <p>Doit être testé, par le programme utilisateur, après chaque opération où il y a risque de débordement, puis remis à 0 en cas de débordement.</p>
%S21	Initialisation GRAFCET	<p>Normalement à l'état 0, est mis à l'état 1 par :</p> <ul style="list-style-type: none"> • reprise à froid, %S0=1, • le programme utilisateur uniquement dans le traitement préliminaire par utilisation de l'instruction S ou bobine Set, • le terminal. <p>A l'état 1, provoque l'initialisation du GRAFCET. Les étapes actives sont désactivées et les étapes initiales sont activées</p> <p>Est remis à 0 par le système après initialisation du Graftet.</p>

Bits système	Fonction	Désignation
%S22	Remise à zéro du GRAFCET	Normalement à l'état 0, ne peut être mis à l'état 1 par programme que dans le traitement préliminaire. A l'état 1, provoque la désactivation des étapes actives sur l'ensemble du GRAFCET. Est remis à l'état 0 par le système en début d'exécution du traitement séquentiel.
%S23	Prépositionnement et gel GRAFCET	Normalement à l'état 0, ne peut être mis à l'état 1 par le programme utilisateur que dans le traitement préliminaire. A l'état 1, il permet de valider le prépositionnement du GRAFCET. Maintenu à l'état 1, provoque le figeage du GRAFCET (gel du graphe). Est remis à 0 par le système en début d'exécution du traitement séquentiel, afin d'assurer l'évolution du GRAFCET à partir de la situation figée.
%S49	Réarmement des sorties statiques	Normalement à l'état 0, est mis à l'état 1 par l'utilisateur pour une demande de réarmement toutes les 10 s à partir de l'apparition du défaut des sorties statiques déclenchées sur sur-intensité ou sur court-circuit.
%S50	Mise à jour de la date et heure par mots %SW50 à 53	Normalement à l'état 0, ce bit peut être mis à 1 ou à 0 par programme ou par le terminal. <ul style="list-style-type: none"> à l'état 0 : accès à la date et à l'heure par lecture des mots système %SW50 à 53. à l'état 1 : mise à jour de la date et l'heure par écriture des mots système %SW50 à 53.
%S51	Date et heure de l'horloge	<ul style="list-style-type: none"> à l'état 0, la date et l'heure sont à jour. à l'état 1, la date et l'heure ne sont pas à jour. Quand ce bit est à 1, la date et l'heure n'ont pas été mises à jour par l'utilisateur ou la batterie est défectueuse.
%S59	Mise à jour de la date et heure par mot %SW59	Normalement à l'état 0, ce bit peut être mis à 1 ou à 0 par programme ou par le terminal. <ul style="list-style-type: none"> à l'état 0 : le système ne gère pas le mot système %SW59 à l'état 1 : le système gère les fronts sur le mot %SW59 pour réglage de la date et l'heure courante.
%S69	Visualisation de bits internes en face avant automate	Normalement à l'état 0, ce bit peut être mis à 1 ou à 0 par programme ou par le terminal. <ul style="list-style-type: none"> à l'état 0 : les états des E/S sont visualisés sur les voyants de l'automate, à l'état 1 : les états de 8 bits internes (TSX 07 10 et 16 E/S) ou 16 bits internes (TSX07 24 E/S) sont visualisés sur les voyants de l'automate (voir ch 1.9, intercalaire A), Le voyant le plus à droite clignote pour signaler que la visualisation des bits internes a été sélectionnée.

Bits système	Fonction	Désignation
%S70	<p>Rafraîchissement des mots d'échange</p> <p>Traitement requête Modbus</p>	<p>Pour l'automate de base, ce bit est mis à l'état 1 dès que celui-ci a effectué un cycle complet d'envoi des mots d'échange %IW/%QW vers les extensions automate.</p> <p>Pour chaque extension automate, ce bit est mis à l'état 1 dès que l'extension a reçu et envoyé les mots d'échange avec l'automate de base.</p> <p>Ce bit est remis à 0 programme ou par terminal.</p> <p>Ce bit est mis à 1 sur traitement d'une requête Modbus.</p> <p>Il peut être exploité par l'utilisateur.</p> <p>Ce bit est remis à zéro par programme ou par terminal.</p>
%S71	Echanges sur la liaison d'extension	<p>Initialement à l'état 0. Est mis à l'état 1 dès qu'un échange sur la liaison d'extension est détecté.</p> <p>Ce bit est mis à l'état 0 quand aucun échange ne s'effectue sur la liaison d'extension. Le mot %SW71 de l'automate de base donne la liste et l'état des extensions présentes.</p>
%S72	Scrutation des extensions automate	<p>Uniquement avec les automates de version inférieure ou égale à V2.2.</p> <p>Normalement à l'état 0. Peut être mis à 1 par programme ou terminal.</p> <ul style="list-style-type: none"> état 0 : scrutation des extensions automates état 1 : inhibition de la scrutation
%S100	Etat du signal /DPT	<p>Indication de l'état du strap INL/DPT sur la prise console :</p> <ul style="list-style-type: none"> Strap absent: protocole UNI-TELWAY maître (%S100 = 0) Strap présent: (/DPT au 0V) protocole défini par la configuration de l'application (%S100 = 1).
%S118	Défaut automate	<p>Normalement à l'état 0. Est mis à l'état 1 quand un défaut d'entrées/sorties ou la disjonction des sorties statiques protégées est détecté sur l'automate de base. Le mot %SW118 permet de déterminer la nature du défaut.</p> <p>Le bit %S118 est remis à 0 dès la disparition du défaut.</p>
%S119	Défaut automate	<p>Normalement à l'état 0. Est mis à l'état 1 quand un défaut d'entrées/sorties ou la disjonction des sorties statiques protégées est détecté sur l'extension d'entrées/sorties. Le mot %SW119 permet de déterminer la nature du défaut.</p> <p>Le bit %S119 est remis à 0 dès la disparition du défaut.</p>

6.2 Mots système

6.2-1 Liste des mots système

Mot	Fonction	Gestion
%SW0	Valeur de la période de scrutation automate (en périodique)	U
%SW11	Durée du chien de garde logiciel	S
%SW14	Time-out UNITELWAY	S et U
%SW30	Temps du dernier cycle de scrutation automate	S
%SW31	Temps de cycle maximum de scrutation automate	S
%SW32	Temps de cycle minimum de scrutation automate	S
%SW50	Fonction Horodateur : mots contenant les valeurs courantes	S et U
%SW51	de la date et l'heure (en BCD)	
%SW52	%SW50 = secondes et jour de la semaine	
%SW53	%SW51= heure et minute %SW52 = mois et jour %SW53= siècle et année	
%SW54	Fonction Horodateur : mots contenant la date et l'heure du	S
%SW55	dernier défaut secteur ou arrêt automate (en BCD)	
%SW56	%SW54 = secondes et code défaut	
%SW57	%SW55 = heure et minute %SW56 = mois et jour %SW57= siècle et année	
%SW58	Code d'identification du dernier arrêt	S
%SW59	Réglage de la date courante	U
%SW67	Valeur du caractère de fin de trame Modbus mode ASCII	U
%SW68	Valeur caractère fin de trame (réception) en mode ASCII (prise TER)	U
%SW69	Code d'erreur du bloc EXCH	S
%SW70	Fonction et type d'automate TSX Nano	S
%SW71	Equipements présents sur liaison d'extension	S
%SW76	Temporisateur 1 ms	S
%SW77	Temporisateur 1 ms	S
%SW78	Temporisateur 1 ms	S
%SW79	Temporisateur 1 ms	S
%SW100	Mot de commande fonction entrée module analogique	U
%SW101	Valeur d'entrée module analogique acquise	S
%SW102	Mot de commande fonction sortie module analogique	U
%SW103	Valeur de sortie module analogique à générer	U
%SW110	Valeur de comptage capturée	S
%SW111	Fonctions de Comptage rapide	S et U
%SW112	Valeur du point de réglage analogique n°0	S
%SW113	Valeur du point de réglage analogique n°1	S

Mot	Fonction	Gestion
%SW114	Validation des blocs horodateurs	U
%SW118	Mot d'état automate de base	S
%SW119	Mot d'état de l'extension d'entrées/sorties	S

S = géré par le système,

U = géré par l'utilisateur.

6.2-2 Description détaillée des mots système

Les automates TSX Nano disposent de mots système décrits ci-après :

Mots système	Fonction	Désignation
%SW0	Période de scrutation	Permet de modifier la période de scrutation automate définie en configuration, par le programme utilisateur ou par le terminal (en mode Réglage).
%SW11	Durée du chien de garde	Permet de lire la durée du chien de garde (150ms).
%SW14	Time-out Unitelway	Permet de modifier la valeur du time-out UNITELWAY, par le programme utilisateur (voir intercalaire F chapitre 1.6)
%SW30	Dernier temps exécution (1)	Indique le temps d'exécution du dernier cycle de scrutation automate (en ms).
%SW31	Temps d'exécution maxi (1)	Indique le temps d'exécution du cycle le plus long de scrutation automate, depuis le dernier démarrage à froid (en ms).
%SW32	Temps d'exécution mini (1)	Indique le temps d'exécution du cycle le plus court de scrutation automate, depuis le dernier démarrage à froid (en ms).
%SW50 %SW51 %SW52 %SW53	Fonction Horodateur	Mots système contenant les valeurs courantes de la date et de l'heure (en BCD) : %SW50 : SSXN Secondes et jour de la semaine avec (N= 0 pour Lundi à 6 pour Dimanche) %SW51 : HHMM Heure et Minute, %SW52 : MMJJ Mois et Jour, %SW53 : SSAA Siècle et Année. Ces mots sont gérés par le système lorsque le bit %S50 est à 0. Ces mots peuvent être écrits par programme utilisateur ou par terminal lorsque le bit %S50 est mis à 1
%SW54 %SW55 %SW56 %SW57	Fonction Horodateur	Mots système contenant la date et l'heure du dernier défaut secteur ou arrêt automate (en BCD) : %SW54 : secondes et jour de la semaine, %SW55 : heure et minute, %SW56 : mois et jour, %SW57 : siècle et année.

(1) Ce temps correspond au temps écoulé entre le début (acquisition des entrées) et la fin (mise à jour des sorties) d'un cycle de scrutation.

Mots système	Fonction	Désignation																											
%SW58	Code du dernier arrêt	Mentionne le code donnant la cause du dernier arrêt : 1= passage de RUN à STOP par le terminal 2= arrêt sur défaut logiciel (débordement de tâche automate) 4= coupure secteur 5= arrêt sur défaut matériel																											
%SW59	Réglage de la date courante	Contient deux séries de 8 bits pour régler la date courante. L'action est toujours réalisée sur front montant du bit. Ce mot est validé par le bit %S59. <table> <thead> <tr> <th>Incrémentation</th><th>Décrémentation</th><th>Paramètre</th></tr> </thead> <tbody> <tr> <td>bit 0</td><td>bit 8</td><td>jour de la semaine</td></tr> <tr> <td>bit 1</td><td>bit 9</td><td>secondes</td></tr> <tr> <td>bit 2</td><td>bit 10</td><td>minutes</td></tr> <tr> <td>bit 3</td><td>bit 11</td><td>heures</td></tr> <tr> <td>bit 4</td><td>bit 12</td><td>jours</td></tr> <tr> <td>bit 5</td><td>bit 13</td><td>mois</td></tr> <tr> <td>bit 6</td><td>bit 14</td><td>années</td></tr> <tr> <td>bit 7</td><td>bit 15</td><td>siècles</td></tr> </tbody> </table>	Incrémentation	Décrémentation	Paramètre	bit 0	bit 8	jour de la semaine	bit 1	bit 9	secondes	bit 2	bit 10	minutes	bit 3	bit 11	heures	bit 4	bit 12	jours	bit 5	bit 13	mois	bit 6	bit 14	années	bit 7	bit 15	siècles
Incrémentation	Décrémentation	Paramètre																											
bit 0	bit 8	jour de la semaine																											
bit 1	bit 9	secondes																											
bit 2	bit 10	minutes																											
bit 3	bit 11	heures																											
bit 4	bit 12	jours																											
bit 5	bit 13	mois																											
bit 6	bit 14	années																											
bit 7	bit 15	siècles																											
%SW67	Fin de trame Modbus	Permet de paramétrer le 'LF' de fin de trame en Modbus mode ASCII. Ce mot est écrit à 16#000A par le système sur initialisation à froid . L'utilisateur peut modifier ce mot par programme ou Réglage quand le Maître utilise un caractère de fin de trame différent de 16#000A.																											
%SW68	Fin de trame Réception Mode ASCII	Permet de paramétrer la valeur de l'octet de fin de trame en mode ASCII. La réception s'arrête dès que celui-ci est reçu. La valeur par défaut est 16#000D.																											
%SW69	Code d'erreur bloc EXCH	En cas d'erreur lors de l'utilisation du bloc EXCH, les bits de sortie %MSG.D et %MSG.E passent à 1. Ce mot système contient le code d'erreur. Les valeurs possibles sont les suivantes: 0: Pas d'erreur, échange correct 1: Buffer d'émission trop grand 2: Buffer d'émission trop petit 3: Table trop petite 4: Mauvaise adresse Unitelway (mode Unitelway uniquement) 5: Time - out écoulé (mode Unitelway uniquement) 6: Erreur d'émission (mode Unitelway uniquement) 7: Mauvaise commande ASCII (mode ASCII uniquement) 8: Non utilisé 9: Erreur de réception (mode ASCII uniquement) 10: Table %KWi interdite. Ce mot est positionné à 0 à chaque utilisation du bloc EXCH.																											

Mots système	Fonction	Désignation
%SW70	Adresse et type d'automate	<p>Contient les informations suivantes :</p> <ul style="list-style-type: none"> • bit 2 : 1=présence horodateur • bit 4 bit 3 Type automate TSX Nano <ul style="list-style-type: none"> 0 0 TSX Nano à 6 entrées/4 sorties (10 E/S) 0 1 TSX Nano à 9 entrées/7 sorties (16 E/S) 1 0 TSX Nano à 14 entrées/10 sorties (24 E/S) 1 1 TSX Nano à entrées alternatives (16 E/S) • bits 7,6 et 5 : adresse de l'automate (recopie du sélecteur de codage d'adresse). <p>Si une extension d'E/S est présente :</p> <ul style="list-style-type: none"> • bits 12 et 11: type d'extension d'E/S (même codage que les bits 3 et 4) • bits 13 : 1 = extension d'E/S présente <p>Les bits inutilisés sont à 0.</p>
%SW71	Equipements présents sur liaison d'extension	<p>Indique l'état de la communication de chaque extension présente avec l'automate de base :</p> <p>bit 1 : extension d'E/S</p> <p>bit 2 : extension automate n°2</p> <p>bit 3 : extension automate n°3</p> <p>bit 4 : extension automate n°4</p> <p>Bit à l'état 0 si extension absente, non alimenté, ou en défaut.</p> <p>Bit à l'état 1 si extension présente et échangeant avec l'automate de base.</p>
%SW76 à %SW78	Mots Décompteurs 1 ms	<p>Ces 4 mots servent de temporisateurs 1 ms.</p> <p>Ils sont décrémentés individuellement par le système toutes les millisecondes si leur valeur est positive. Cela donne 4 décompteurs de temps à la milliseconde, soit une plage d'exploitation de 1 ms à 32767 ms.</p> <p>La mise à 1 du bit 15 permet d'arrêter la décrément.</p>
%SW100	Entrée analogique	<p>Mot de commande des fonctions entrée analogique.</p> <p>Valeur: 0 Entrée analogique non validée</p> <p>Valeur: 1 Fonctionnement sans mise à l'échelle</p> <p>Valeur: 2 Mise à l'échelle gamme unipolaire (période 125ms)</p> <p>Valeur: 3 Mise à l'échelle gamme bipolaire (période 125ms)</p> <p>Valeur: 4 Mise à l'échelle gamme unipolaire (période 500ms)</p> <p>Valeur: 5 Mise à l'échelle gamme bipolaire (période 500ms)</p> <p>L'écriture de ce mot est à la charge de l'applicatif</p>

Mots système	Fonction	Désignation
%SW101	Entrée analogique	<p>Mot contenant la valeur d'entrée analogique acquise. La gamme de valeur est fonction du choix de fonctionnement fait dans %SW100.</p> <p>%SW100=0 %SW101=0</p> <p>%SW100=1 %SW101 varie de 0 à 1000</p> <p>%SW100=2 ou 4 %SW101 varie de 0 à 10000</p> <p>%SW100=3 ou 5 %SW101 varie de -10000 à 10000</p>
%SW102	Sortie analogique	<p>Mot de commande des fonctions sortie analogique.</p> <p>Valeur: 0 Fonctionnement %PWn normal</p> <p>Valeur: 1 Fonctionnement sans mise à l'échelle</p> <p>Valeur: 2 Mise à l'échelle gamme unipolaire</p> <p>Valeur: 3 Mise à l'échelle gamme bipolaire</p> <p>L'écriture de ce mot est à la charge de l'applicatif</p> <p style="text-align: right;">%PWn analogique</p>
%SW103	Sortie analogique	<p>Mot contenant la valeur à appliquer sur la sortie analogique. La gamme de valeur est fonction du choix de fonctionnement fait dans %SW102.</p> <p>%SW102=0 %SW103=0</p> <p>%SW102=1 %SW103 compris entre 5 et 249</p> <p>%SW102=2 %SW103 compris entre 0 et 10000</p> <p>%SW102=3 %SW103 compris entre -10000 et 10000</p> <p>L'écriture de ce mot est à la charge de l'applicatif</p>
%SW110	Comptage/décomptage	Valeur capturée du compteur sur front montant de l'entrée %I0.4.
%SW111	Comptage rapide	<p>bit 0 : sens de déplacement (1=comptage, 0=décomptage)</p> <p>bit 1 : 1= validation des sorties réflexes</p> <p>bit 2 : 1= choix de la base de temps du fréquencemètre (1=100ms, 0=1s)</p> <p>bit 3 : 1= rafraîchissement de %FC en fréquence (indique également la validité de la valeur acquise sur le module d'entrée analogique)</p> <p>ce bit est remis à 0 par l'utilisateur</p>
%SW112	Valeur point de réglage analogique 0	Contient la conversion sur 8 bits (0 à 255) de la position du potentiomètre n° 0.
%SW113	Valeur point de réglage analogique 1	Contient la conversion sur 8 bits (0 à 255) de la position du potentiomètre n°1.
%SW114	Validation horodateur	<p>Permet de valider ou d'inhiber le fonctionnement des horodateurs par programme utilisateur ou par terminal.</p> <p>bit 0 : 1 = validation horodateur n°0</p> <p>.....</p> <p>bit 15 : 1 = validation horodateur n°15</p> <p>Initialement tous les blocs horodateurs sont validés.</p>

Mots système	Fonction	Désignation
%SW118	Etat automate de base	Indique les défauts détectés sur l'automate de base. bit 0 : 0 = disjonction des sorties statiques (1) bit 3 : 0 = défaut alimentation capteur bit 8 : 0 = défaut interne ou défaut matériel TSX Nano bit 9 : 0 = défaut externe ou défaut dialogue bit 11 : 0 = automate en auto-tests bit 13 : 0 = défaut de configuration (extension d'E/S configurée mais absente ou en défaut) Tous les autres bits de ce mot sont à 1 et sont réservés. Ainsi, pour un automate sans défaut , ce mot a pour valeur : 16#FFFF.
%SW119	Etat automate d'extension d'E/S	Indique les défauts détectés sur l'automate d'extension d'E/S (ce mot n'est géré que par l'automate de base). L'affectation des bits de ce mot est identique à celle de %SW118 excepté : <ul style="list-style-type: none"> • bit 13 : non significatif • bit 14 : absence de l'extension alors que cette dernière était présente à l'initialisation.

(1) suite à surcharge ou court-circuit sur l'une des sorties.

7.1 Modes de marche

Le langage PL7 permet la prise en compte des trois grandes familles de modes de marches (1) :

- Vérification,
- Marche ou production,
- Arrêt.

Ces différents modes de marches peuvent être obtenus autour ou à partir du Grafcet par les possibilités suivantes :

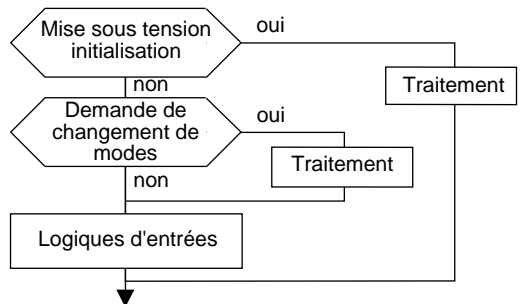
- Initialisation du Grafcet,
- Prépositionnement d'étapes,
- Maintien de situation,
- Gel de graphe.

L'utilisation du traitement préliminaire et des bits système assure la gestion des modes de marches sans compliquer et alourdir le programme utilisateur.

Structuration du traitement préliminaire

Le synoptique ci-dessous montre la structure à donner au traitement préliminaire afin d'effectuer par ordre d'importance chaque traitement en cas de :

- Mise sous tension
- Changement de modes de marches
- Logiques d'entrées



Bits système Grafcet

L'utilisation des bits %S21, %S22 et %S23 est réservée au seul traitement préliminaire. Ces bits sont remis à 0 automatiquement par le système, l'écriture de ces bits doit être faite uniquement par code opération S.

Initialisation du Grafcet, %S21

Causes :

- sur reprise à froid,
- mise à 1 de %S21 par programme ou terminal.

Conséquences : désactivation de toutes les étapes actives et activation de toutes les étapes initiales.

(1) Ces modes de marches sont définies par le GEMMA ("Guide pour l'étude des modes de marches et d'arrêts" proposé par l'ADEPA)

Remise à zéro du Grafcet, %S22

Causes : mise à 1 de %S22 par programme ou terminal.

Conséquences :

- désactivation de toutes les étapes actives,
- arrêt de scrutation du traitement séquentiel.

Prépositionnement du Grafcet, %S22 et %S23

Procédure :

- remise à zéro du Grafcet par mise à 1 de %S22,
- prépositionnement des étapes à activer par une suite d'instructions S Xi,
- validation du prépositionnement par mise à 1 de %S23

Figeage d'une situation

- en situation initiale : par maintien à 1 par programme de %S21,
- en situation "vide" : par maintien à 1 par programme de %S22,
- en situation déterminée par maintien à 1 de %S23.

7.2 Conseils de programmation

Gestion des sauts de programme

Utiliser les sauts de programme avec précaution afin d'éviter des boucles trop longues pouvant augmenter le temps de cycle. Eviter les sauts de programme vers les instructions situées en amont.

Programmation des sorties

Chaque bit de sortie ou bit interne ne doit être piloté qu'une seule fois dans le programme. Dans le cas des bits de sortie, seule la dernière valeur scrutée est prise en compte lors de la mise à jour des sorties.

Prise en compte des sécurités directes

Les capteurs concernant les sécurités directes ne doivent pas être traités par l'automate. Ils doivent agir directement sur les pré-actionneurs correspondants.

Gestion des reprises secteur

Conditionner une reprise secteur à une opération manuelle, un redémarrage automatique de l'installation pouvant être dangereux (utilisation des bits système %S0, %S1 et %S9)

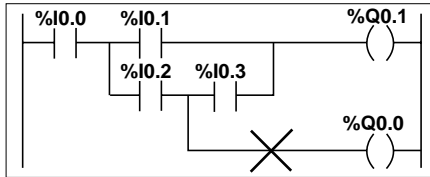
Gestion du temps et du bloc horodateur

Il est conseillé de contrôler l'état du bit %S51 qui indique tout défaut de l'horodateur.

Nota : lors de la saisie du programme, le terminal contrôle la syntaxe des instructions, des opérandes et leur association. La fonction diagnostic du terminal permet de vérifier les erreurs de programmation (voir Annexes intercalaire C).

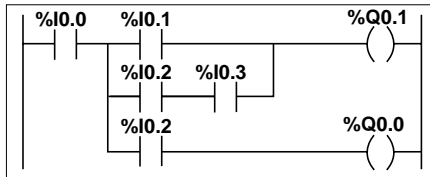
Complément sur l'utilisation des parenthèses

- Les opérations d'affectation ne doivent pas être placées dans des parenthèses.



```
LD    %I0.0
AND   %I0.1
OR(   %I0.2
ST    %Q0.0
AND   %I0.3
)
ST    %Q0.1
```

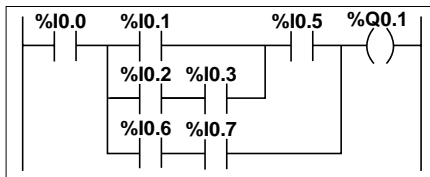
Pour réaliser la même fonction, on programmera les équations suivantes :



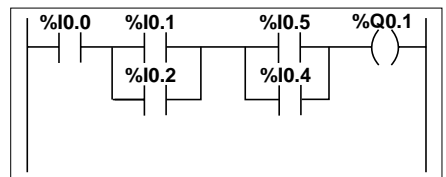
```
LD    %I0.0
MPS
AND(   %I0.1
OR(   %I0.2
AND   %I0.3
)
)
ST    %Q0.1
MPP
AND   %I0.2
ST    %Q0.0
```

- Si plusieurs mises en parallèle de contacts sont effectuées, elles devront être imbriquées les unes dans les autres ou complètement dissociées.

Exemple 1 :

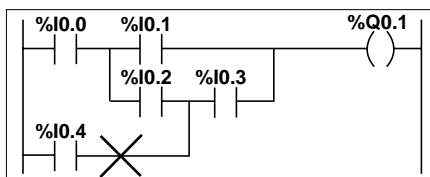


Exemple 2

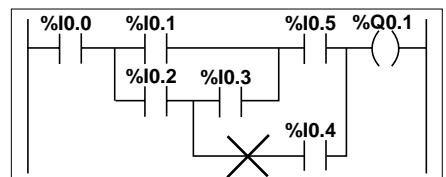


Par contre les schémas suivants ne peuvent pas être programmés

Exemple 3

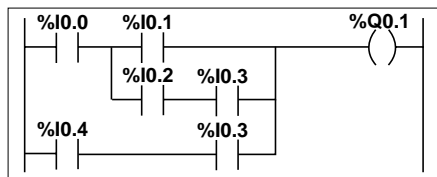


Exemple 4



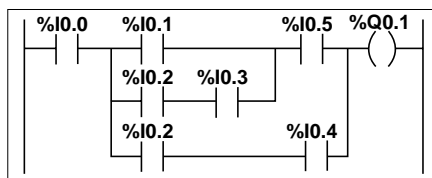
Pour réaliser des schémas équivalents à ceux de la page précédente, il faut les modifier de la façon suivante :

Exemple 5 (voir exemple 3)



```
LD    %I0.0
AND(  %I0.1
OR(   %I0.2
AND   %I0.3
)
)
OR(   %I0.4
AND   %I0.3
)
ST    %Q0.1
```

Exemple 6 (voir exemple 4)



```
LD    %I0.0
AND(  %I0.1
OR(   %I0.2
AND   %I0.3
)
AND   %I0.5
OR(   %I0.2
AND   %I0.4
)
)
ST    %Q0.1
```

7.3 Réarmement des sorties statiques protégées sur TSX 07 ●●●12

Lorsqu'un défaut a provoqué la disjonction des sorties d'un automate (AP de base ou extension d'E/S), il est nécessaire de procéder à leur réarmement. Ce réarmement peut être :

- demandé par un ordre opérateur. C'est le type de réarmement recommandé (voir chapitre A 1.7-2),
- automatique. Ce type de réarmement nécessite au préalable de connaître les conséquences que peut provoquer la réactivation automatique des sorties sur le process.

Ce choix s'effectue à l'aide du bit système %S49.

Disjonction des sorties

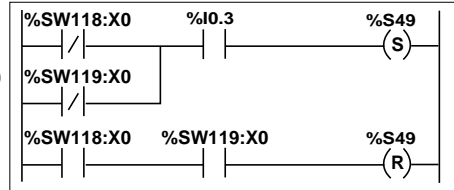
L'apparition d'une surcharge ou d'un court-circuit sur une sortie provoque :

- la limitation en courant de la sortie concernée,
- la disjonction de toutes les sorties du bloc (AP de base ou extension d'E/S),
- l'activation en fixe du (ou des) voyant(s) I/O de l'AP de base (et) de l'extension d'E/S (disjonction des voies de l'extension d'E/S),
- la mise à 0 du bit système Défaut E/S %S10,
- la mise à 1 du bit système %S118 (disjonction des sorties de l'AP de base) ou du bit système %S119 (disjonction des sorties de l'extension d'E/S),
- la mise à 0 du bit du mot système %SW118:X0 (disjonction des sorties de l'AP de base) ou du bit du mot système %SW119:X0 (disjonction des sorties de l'extension d'E/S).

Réarmement manuel des sorties (conditionné à un ordre opérateur)

Le bit système %S49 est mis à 1 par une action extérieure. Le bit %S49 doit être remis à 0 après le réarmement effectif des sorties. Le programme correspondant est :

```
LDN    %SW118:X0
ORN    %SW119:X0
AND    %I0.3 (entrée I0.3 par exemple)
S      %S49
LD     %SW118:X0
AND    %SW119:X0
R      %S49
```



Le temps minimum entre deux réarmements, garanti par le système est de 10s. Si le défaut ayant entraîné la disjonction a disparu :

- les sorties sont de nouveau activées selon l'état défini par le programme,
- les voyants I/O sont éteints,
- les bits système et bits de mots système sont positionnés dans leur état par défaut : %S10, %SW118:X0, %SW119:X0 à l'état 1, %S118 et %S119 à l'état 0 (voir chapitre B 5.2).

Réarmement automatique des sorties

Le bit système %S49 est positionné en permanence à 1 par le programme suivant :

```
LD     1
ST     %S49
```

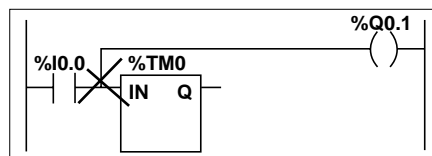
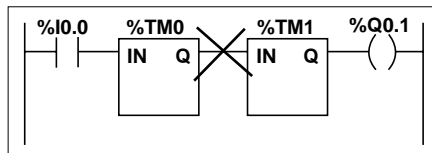


Le réarmement est demandé automatiquement toutes les 10s. La base de temps de 10s est synchrone par rapport à l'apparition du défaut. Lors du réarmement, le comportement des sorties, des voyants et des bits et mots système est identique à celui décrit pour le réarmement manuel.

7.4 Conditions de réversibilité

Les conditions suivantes doivent être vérifiées pour qu'un programme puisse être entièrement réversible (1) :

- les instructions suivantes ne doivent pas être utilisées : XOR, XORN, XORF, XORR, JMPCN, ENDCN, ou N,
- les blocs fonction doivent être programmés de façon réversible (voir ch 2.2-2),
- les blocs fonction ne doivent pas être programmés en cascade,
- les instructions d'affectation sont interdites entre les instructions BLK et OUT_BLK ou BLK et END_BLK (si OUT_BLK n'est pas programmée).



Programmation interdite

```

BLK  %TM0
LD   %I0.0
ST   %Q0.1
IN
END_BLK

```

Programmation autorisée

```

BLK  %TM0
LD   %I0.0
IN
END_BLK
LD   %I0.0
ST   %Q0.1

```

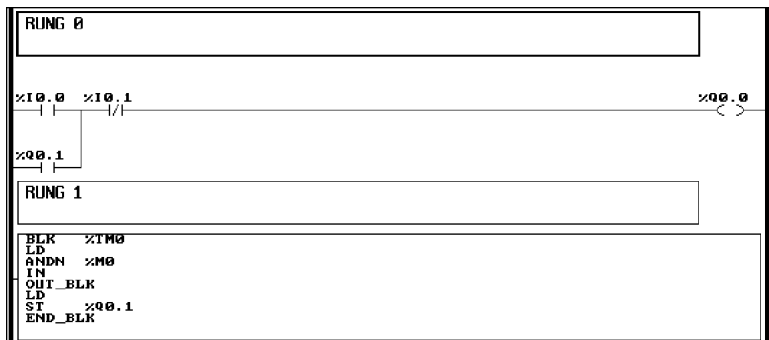
- (1) **Rappel** : lorsqu'une séquence d'instructions n'est pas réversible, elle reste en langage liste d'instructions, alors que la suite du programme réversible est affichée en langage à contacts.

7.5 Règles de réversibilité

- Un réseau complet canonique ne peut avoir une hauteur supérieure à 7 cellules, ni une largeur supérieure à 11 cellules (grille 7x11).
- Une phrase commençant par LD doit finir avec une instruction d'action conditionnelle.
- Les instructions JMPCN, ENDCN, NOP et N ne sont pas réversibles.
- Des instructions d'action à l'intérieur de parenthèses ne sont pas réversibles.
- Des instructions de pile à l'intérieur de parenthèses ne sont pas réversibles.
- Une instruction OR après une instruction d'action n'est pas réversible.
- Les instructions RET, JMP et END sont non conditionnelles. Aucune autre instruction ne peut exister dans le réseau complet.
- Les entrées et sorties de blocs de fonction ne peuvent être accédées que grâce à des instructions de blocs de fonction, standards et réversibles. Les instructions d'accès direct aux blocs fonction ne sont pas réversibles.

- Des instructions après un END_BLK dans une phrase provoquent la non-réversibilité de la phrase.
- Les sorties de blocs fonction utilisées avec des instructions AND et OR ne sont pas réversibles.
- Une instruction OR à l'intérieur d'un réseau complet de sortie de bloc de fonction doit être à l'intérieur de parenthèses.
- Une instruction d'action non-conditionnelle entre un BLK et un END_BLK n'est pas réversible.
- Un OUT_BLK doit être suivi par un LD d'une sortie de bloc de fonction valide ou par un END_BLK.
- Une instruction précédant une instruction MRD ou MPP doit être soit une instruction d'action conditionnelle soit une opération de pile.
- Une instruction OR non incluse dans une paire MPS et MPP n'est pas réversible.
- Une instruction OR après une instruction MPS, MRD, ou MPP n'est pas réversible.
- Une instruction MCS ne peut être utilisée dans un réseau complet avec une autre instruction d'action quelconque.
- Une instruction d'appel de sous-programme ou JMP_C doit être la dernière instruction d'action d'un réseau complet.
- Lorsque vous mettez un titre et des commentaires avant une instruction List, il ne peut y avoir qu'une seule ligne de titre, et que 4 lignes de commentaires au plus. Si une ligne vide est placée n'importe où entre le titre et l'instruction List, une partie de l'en-tête peut ne pas apparaître dans l'en-tête de réseau complet de Ladder correspondant.
- Si plus de 4 lignes de commentaires sont placées avant l'instruction List, la cinquième ligne de commentaire sera interprétée comme une ligne de titre et les lignes de commentaires précédentes n'apparaîtront pas dans l'en-tête de réseau complet de Ladder correspondant.

Quand une séquence d'instructions n'est pas réversible, elle reste dans le langage List, tandis que le reste du programme réversible est converti en diagrammes Ladder, comme montré ci-dessous.



Chapitre	Page
1 Généralités	1/1
1.1 Description du terminal FTX117	1/1
1.2 Raccordement du terminal FTX117	1/2
1.3 Modes de fonctionnement du terminal FTX 117	1/3
1.4 Description de l'écran et du clavier	1/4
1.4-1 Description de l'écran	1/4
1.4-2 Description du clavier	1/4
1.5 Règles générales d'ergonomie	1/6
1.6 Présentation des éditeurs du terminal FTX117	1/7
1.7 Principe de sélection des éditeurs et des fonctions	1/10
1.8 Méthode pour réaliser une application	1/12
2 Première prise en main	2/1
2.1 Mise sous tension du terminal FTX117	2/1
2.2 Mode terminal	2/1
2.3 Choix de la langue	2/2
2.4 Préférences	2/2
3 Configuration de l'application	
3.1 Généralités	3/1
3.2 Configuration générale de l'application	3/2
3.3 Blocs fonction	3/4
3.3-1 Bloc temporisateur %TM	3/3
3.3-2 Bloc compteur %C	3/3
3.3-3 Bloc registre %R	3/4
3.3-4 Bloc programmeur cyclique %DR	3/4

Chapitre	Page
3 Configuration de l'application (suite)	
3.4 Configuration des entrées	3/5
3.4-1 Choix du niveau de filtrage sur les entrées Tout ou Rien de l'automate	3/5
3.4-2 Choix de l'utilisation des entrées à mémorisation d'état	3/5
3.4-3 Fonction compteur rapide et entrées associées	3/6
3.5 Configuration des sorties	3/7
3.5-1 Utilisation en sortie modulation de largeur	3/7
3.5-2 Utilisation en sortie générateur d'impulsions	3/7
3.6 Saisie des constantes	3/8
3.7 Effacement de la configuration	3/8
3.8 Accès direct à la configuration depuis l'éditeur de programme	3/8
3.9 Configuration prise terminal	3/9
4 Programmation	
4.1 Généralités	4/1
4.2 Création d'un programme	4/1
4.3 Modification d'un programme	4/4
4.3-1 Accès à une ligne de programme	4/4
4.3-2 Modification d'une ligne de programme	4/4
4.3-3 Insertion d'une ligne de programme	4/5
4.3-4 Effacement d'une ligne de programme	4/5
4.3-5 Duplication, suppression et déplacement d'un ensemble de lignes	4/6
4.4 Recherche/remplacement	4/7
4.4-1 Recherche d'une ligne de programme, d'une étiquette, sous programme	4/7
4.4-2 Recherche d'une variable dans le programme	4/8
4.4-3 Recherche d'une instruction dans le programme	4/8
4.4-4 Remplacement d'une variable dans le programme	4/9

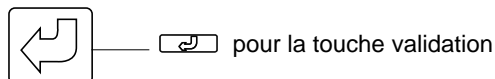
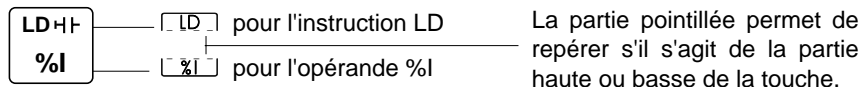
Chapitre	Page
4 Programmation (suite)	
4.5 Diagnostic programme	4/10
4.5-1 Mode autonome ou mode connecté automate en STOP	4/10
4.5-2 Mode connecté automate en RUN	4/10
4.6 Effacement d'une application	4/11
4.7 Information	4/12
4.7-1 Saisie du nom de l'application	4/12
4.7-2 Mot de passe	4/12
4.7-3 Informations visualisées	4/13
5 Configuration des horodateurs	5/1
6 Sauvegarde et transfert de l'application	
6.1 Généralités	6/1
6.2 Principe d'utilisation de la fonction Transfert	6/1
6.3 Sauvegarde/restitution de l'application en mémoire FLASH	6/2
6.4 Sauvegarde/restitution de l'application sur carte mémoire	6/3
6.5 Transfert de l'application Terminal/automate	6/4
6.6 Sauvegarde/restitution de l'application en mémoire EEPROM	6/4
7 Mise en route d'une application	
7.1 Généralités	7/1
7.2 Mise à l'heure de l'horloge interne	7/1
7.3 Initialisation	7/2
7.4 Mise en exécution RUN ou arrêt d'exécution STOP	7/2
7.5 Information	7/2

Chapitre	Page
8 Mise au point	
8.1 Généralités	8/1
8.2 Visualisation dynamique du programme	8/2
8.3 Modification et forçage	8/2
8.4 Modification en RUN du programme	8/4
9 Réglage	
9.1 Généralités	9/1
9.2 Principe d'accès aux objets à régler	9/1
9.3 Objets accessibles par l'éditeur de données	9/2
9.4 Visualisation et modification des variables booléennes	9/3
9.5 Visualisation et modification des variables numériques	9/3
9.6 Visualisation et modification des blocs fonction	9/4
9.7 Visualisation des étapes Grafcet	9/4
9.8 Création et visualisation de tables de données	9/5
9.9 Ré-initialisation des tables de données	9/5
9.10 Transfert de tables de données	9/6

Avant propos

Représentation des touches

Les touches du clavier ont un double marquage, pour des raisons de simplicité dans la représentation et de lisibilité, seul le marquage concerné sera représenté sur la touche.



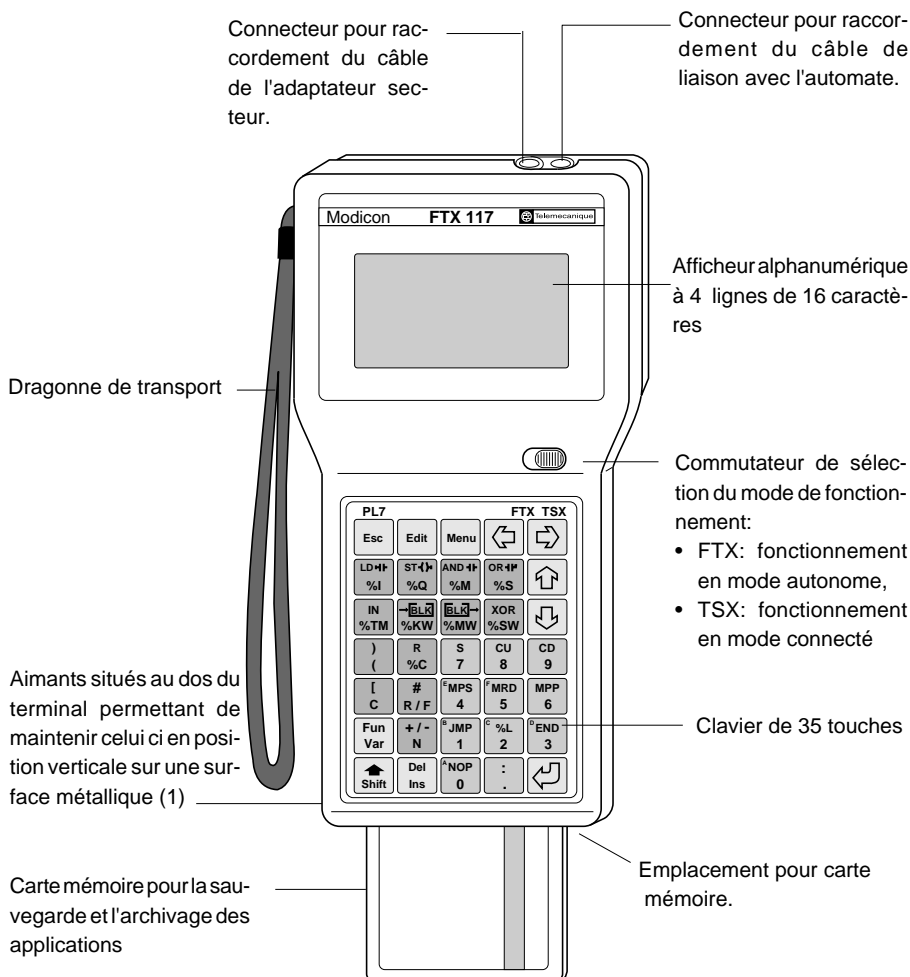
Nouvelles fonctionnalités de la version logicielle supérieure ou égale à V3.0

Les fonctionnalités suivantes ont été rajoutées par les versions logicielles supérieures ou égales à V3.0 (par rapport à la version V2.1) :

- **Fonctions de communication :**
 - protocole Modbus esclave sur port d'extension,
 - protocole ASCII, UNI-TELWAY maître et esclave sur prise terminal.

1.1 Description du terminal FTX 117

Le terminal FTX 117 est l'outil de programmation et de réglage des automates TSX Nano.



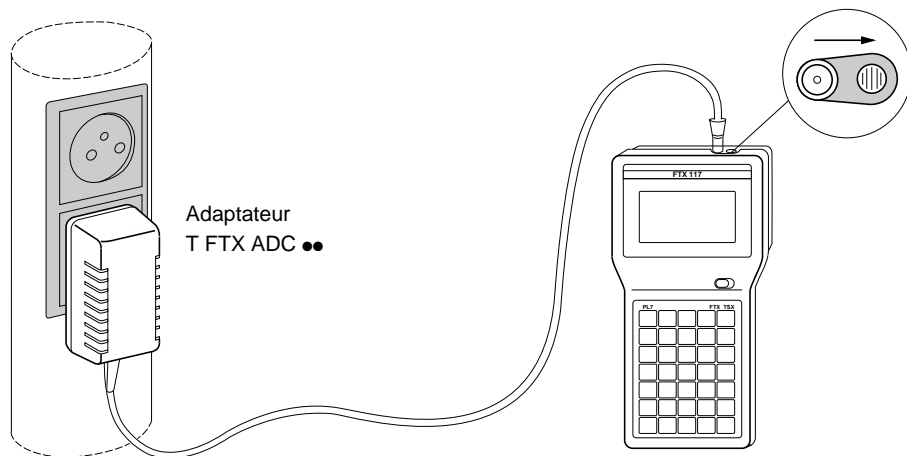
Dimensions : Hauteur : 185 mm, Largeur : 80/95 mm, Epaisseur : 25/30 mm.

(1) veiller à ne pas poser le terminal près de disquettes ou tout autre support magnétique au risque de détruire leur contenu.

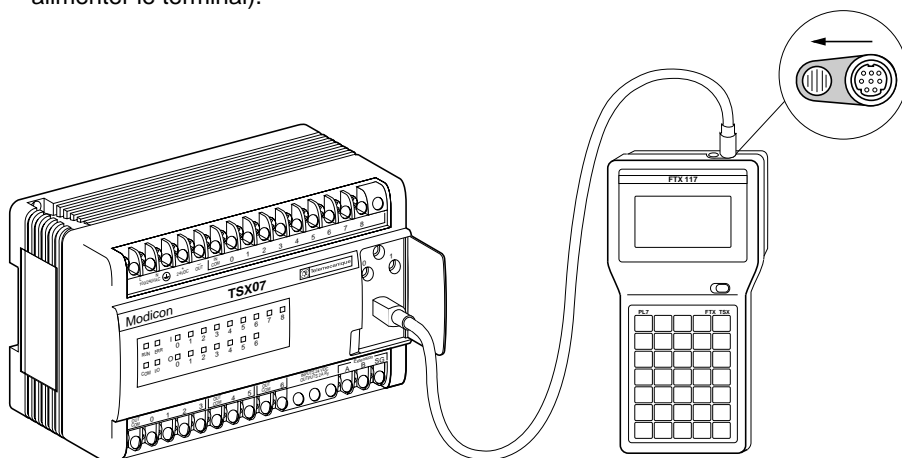
1.2 Raccordement du terminal FTX 117

Le terminal FTX 117 peut être alimenté de deux façons différentes:

- **soit à partir d'une tension secteur 100 à 120V ou 200 à 240V** par l'intermédiaire d'un adaptateur : T FTX ADC 11 (100 à 120V) ou T FTX ADC 12 (200 à 240V). Dans ce cas, le terminal doit être utilisé en **mode autonome** (voir ch1.3).



- **soit à partir de l'automate** par l'intermédiaire du câble de liaison T FTX CB1 020 (longueur : 2 m) ou du câble de liaison T FTX CB1 050 (longueur : 5 m). Dans ce cas le mode privilégié du terminal est le **mode connecté** (voir ch1.3), il peut cependant être utilisé en mode autonome (l'automate sert alors uniquement à alimenter le terminal).



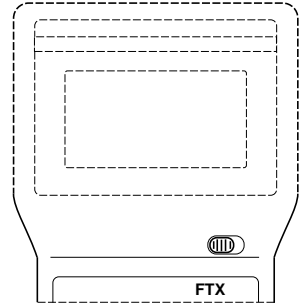
1.3 Modes de fonctionnement du terminal FTX 117

Le terminal FTX 117 propose 2 modes de fonctionnement sélectionnés par commutateur **FTX/TSX** situé sous l'écran :

- **Mode autonome**

Le commutateur **FTX/TSX** doit être sur la position **FTX**. La saisie du programme et de la configuration de l'application s'effectue en mémoire RAM du terminal (1).

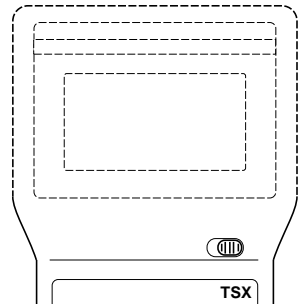
- (1) La mémoire RAM du terminal ayant une autonomie d'une heure lorsque le terminal est mis hors tension, il est conseillé de sauvegarder l'application en mémoire Flash du terminal ou sur carte mémoire (voir ch6).



- **Mode connecté**

Le commutateur **FTX/TSX** doit être sur la position **TSX**. La saisie de la configuration et du programme de l'application s'effectue en mémoire RAM de l'automate (cette mémoire est sauvegardée sur coupure secteur).

Le mode connecté permet d'accéder à toutes les fonctions opératoires notamment aux fonctions de mise au point et de réglage de l'application.



Important

Changement de modes de fonctionnement

Le passage d'un mode de fonctionnement à un autre mode est pris en compte lorsque l'écran de base (écran d'accès aux éditeurs, voir ch 1.6) est affiché.

1.4 Description de l'écran et du clavier

1.4-1 Description de l'écran

L'écran du terminal FTX 117 est constitué par un afficheur rétro-éclairé à cristaux liquides de 4 lignes de 16 caractères.

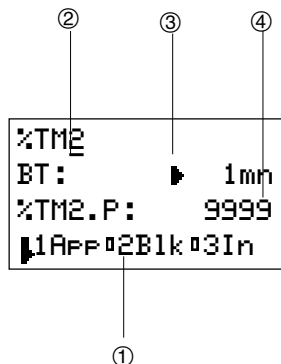
Les éléments caractéristiques des écrans sont les suivants :

- ① Ligne de menu, cette ligne est utilisée dans l'écran de base, et peut être affichée dans tout autre écran par `[Menu]`.

- ② Curseur, permet de pointer l'élément à modifier. Le déplacement du curseur s'effectue par `[←]` `[→]` `[↑]` `[↓]`.

- ③ Le signe `▶` mentionne les paramètres à choix multiples (sans saisie opérateur), la sélection s'effectue par `[←]` `[→]`.

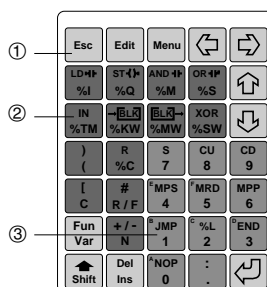
- ④ Zone de saisie numérique.



1.4-2 Description du clavier

Le clavier du terminal FTX 117 comprend 3 zones repérées par 3 couleurs différentes :

- ① zone modes opératoires, touches en bleu clair
- ② zone saisie d'instructions usuelles, touches en bleu foncé, avec :
- en jaune, les instructions
 - en blanc, les opérandes
- ③ zone clavier hexadécimal, touches en gris

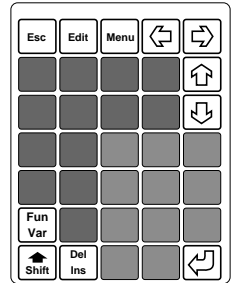


Dans le cas général, le contexte d'utilisation détermine automatiquement la fonction utile des touches à double marquage (inutile d'appuyer sur `[Shift]`).

① Zone modes opératoires (en bleu clair)

Elle comprend :

- les touches d'accès aux éditeurs et aux fonctions à réaliser (Edit et Menu),
- les touches de sélection et déplacement du curseur (↑, ↓, ←, →),
- les touches Fun et Var d'accès aux variables et aux instructions non inscrites au clavier,
- les touches insertion, effacement, validation, annulation et shift (Ins, Del, ↵, Esc, Shift).

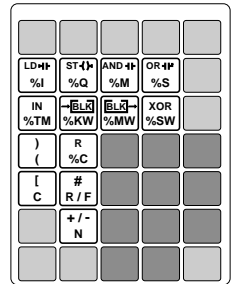


② Zone saisies d'instructions (en bleu foncé)

Ces touches à double marquage permettent suivant le contexte :

- soit la saisie de l'instruction (exemple : %LD) lorsque le curseur est positionnée sur une zone instruction,
- soit la saisie de la variable (exemple : %) lorsque le curseur est positionné sur une zone variable.

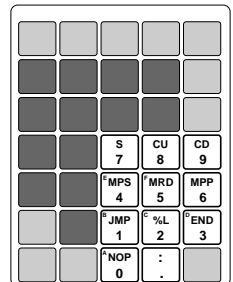
[C], [R], [R/F] appuyées après une instruction peuvent compléter celle-ci : exemple [LD] puis [N] permettent de saisir l'instruction LDN.



③ Zone clavier hexadécimal (en gris)

Ces touches permettent suivant le contexte :



- soit la saisie de valeurs numériques (accès aux lettres A à F, en appuyant d'abord sur Shift)
- soit la saisie d'instructions (sur ces touches sont regroupées les instructions de structuration de programme (JMP, %L, MPS, MRD...))



1.5 Règles générales d'ergonomie

Les quelques règles suivantes assurent une maîtrise rapide du terminal FTX 117.

Sélection d'une fonction

Les fonctions de base pour programmer et mettre au point une application sur un automate TSX Nano sont regroupées dans 5 éditeurs : Automate, Programme, Données, Configuration et Terminal (voir ch 1.6) accessible par . Chaque éditeur offre un menu accessible par . Les principes de sélection sont détaillés chapitre 1.7.

Saisie des instructions et des opérandes

L'accès aux instructions et opérandes mentionnés sur le clavier s'effectue simplement en appuyant sur la touche sans se soucier du double marquage, le terminal effectue lui-même le choix entre instructions et opérandes suivant le contexte.




Accès aux instructions et opérandes non mentionnés sur le clavier :

 et  donnent accès aux instructions et opérandes non accessibles directement au clavier (voir ch4.2).

 donne accès aux instructions numériques.

1	SR	5	U
2	RET	6	MCS
3	I	7	MCR
4	0	8	==

Saisie des paramètres

Les paramètres (exemple : paramètres de configuration) suivis du signe  ne demandent aucune saisie, le choix de la valeur s'effectue par   (exemple, écran ci-contre : BT).



Les autres paramètres sont saisis à l'aide des touches numériques (exemple, écran ci-contre : %TM2.P).




```
%TM2
BT:      1mn
%TM2.P:  9999
1A# 2B1k 3In
```

D'une manière générale les paramètres ou données suivis du signe ":" sont modifiables, ceux suivis du signe "=" ne le sont pas.


Utilisation des touches

Certaines touches ont une utilisation spécifique :

  déplace le curseur vers le haut et le bas, mais aussi d'un champ à l'autre dans ce cas le curseur peut se déplacer de façon horizontale;

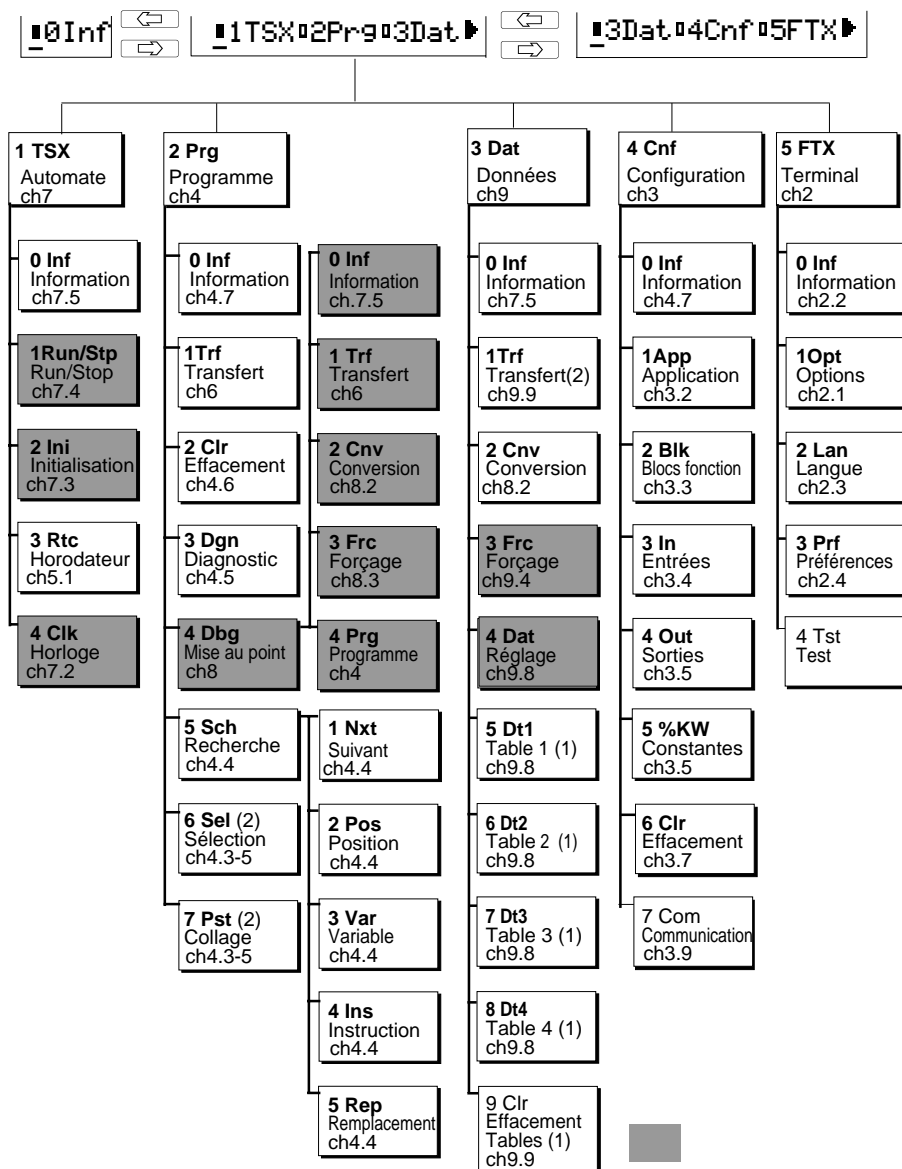
 active la fonction supérieure de la touche, dans le cas où le contexte n'est pas suffisant (ex : , , saisie hexadécimale).

Important

Pour accéder au marquage supérieur de la touche, appuyer d'abord sur , relacher puis appuyer sur la touche désirée.

1.6 Présentation des éditeurs du terminal FTX117

Les éditeurs sont proposés dans le menu de l'écran de base.



(1) non disponible en version logicielle V1

(2) non accessible en connecté en version logicielle V2.

Accès uniquement en mode connecté commutateur FTX/TSX sur TSX

• **Editeur automate (TSX):** visualise l'état de l'automate:

- en marche (RUN), ou à l'arrêt (STOP),
- application exécutable ou non,
- type d'automate connecté,

Le menu affiche les différentes fonctions :

- **0 Inf** : informations sur état automate,
- **1 Run** : mise en marche ou,
Stp arrêt) selon l'état initial de l'automate,
- **2 Ini** : initialisation de la mémoire automate,
- **3 Rtc** : visualisation et paramétrage des horodateurs,
- **4 Clk** : mise à l'heure de l'horloge de l'automate.

```
TSX: RUN
APP: Exec
TSX 07 31-16
_1Stp_2Ini_3Rtc_
```

```
_Ve20 _Dec 1996
      17:26:48
$IB# 2 Jan
      2 17:07:23
```

• **Editeur de programme (Prg):** permet l'accès à l'éditeur de programme liste d'instructions.

[Menu] affiche les différentes fonctions de l'éditeur

- **0 Inf** : informations sur l'application, saisie du nom, mot de passe
- **1 Trf** : transfert ou archivage d'une application,
- **2 Clr** : effacement mémoire (configuration et programme, ou programme uniquement),
- **3 Dgn** : diagnostic programme,
- **4 Dbg** : mise au point,
- **5 Sch** : donne accès à un sous menu proposant les fonctions de recherche/remplacement,.
- **6 Sel** : sélection de lignes de programme,
- **7 Pst** : collage de la sélection effectuée.

```
000 LD      %I0.1
001 AND     %I0.2
002 ST      %Q0.1
003 LD      %M0
```

```
000 LD      %I0.1
001 AND     %I0.2
002 ST      %Q0.1
_1Trf_2Clr_3Dgn_
```

La fonction mise au point **Dbg**, accessible aussi directement par [Shift] et [↔], visualise dynamiquement l'état des variables du programme.

[Menu] affiche les différentes fonctions :

- **0 Inf** : informations sur l'automate,
- **1 Trf** : transfert ou archivage d'une application,
- **2 Cnv** : affichage en hexadécimal, en ASCII ou en décimal des mots,
- **3 Frc** : forçage des bits à 0 ou 1,
- **4 Prg** : retour au menu Programme,

```
%I0.1      0
%I0.2      1
%Q0.1      0
_1Trf_2Cnv_3Frc_
```

• **Editeur de données (Dat):** permet l'accès à toutes les variables pour visualiser leurs états (ou leurs valeurs), les modifier ou les forcer.

```
%I0.5
Err=0 ...f....
      .....f..
%Q0.0
```

Saisir la variable à visualiser

Exemple: visualisation de bits internes

☐ %M ☐ 0

☐ visualise les huit bits internes à partir de l'adresse saisie.

☐ Menu affiche les différentes fonctions :

- 0 Inf : informations sur état automate,
- 1 Trf : transfert ou archivage de tables de données
- 2 Cnv : conversion en hexadécimal, en ASCII, en binaire ou en décimal des mots,
- 3 Frc : forçage des bits à 0 ou 1,
- 4 Dat : retour à la fonction réglage en cours,
- 5 Dt1 : saisie de la table de données n°1,
- 6 Dt2 : saisie de la table de données n°2,
- 7 Dt3 : saisie de la table de données n°3,
- 8 Dt4 : saisie de la table de données n°4,
- 9 Clr : effacement des tables de données n°1 à 4,

```
%M0 0 %M1 0
%M2 0 %M3 0
%M4 0 %M5 0
1Trf 2Cnv 3Frc
```

```
%TM0 %TM0.0=1
TON BT=100ms
2TM0.P:
2TM0.V:
```

- **Editeur de configuration (Cnf):** Il permet l'accès à l'éditeur de configuration de l'application.

☐ Menu affiche les différentes fonctions :

- 0 Inf : informations sur l'application, saisie du nom, mot de passe
- 1 App : configuration niveau application (entrée RUN/STOP, sortie sécurité, scrutation normale ou périodique, extension automates, Modbus),
- 2 Blk : configuration des blocs fonctions (Temporisateur, compteur...)
- 3 In : configuration des entrées (Filtrage, %FC,...)
- 4 Out : configuration des sorties (%PWM, %PLS,...)
- 5 %KW : saisie des valeurs des mots constants
- 6 Clr : effacement de la configuration
- 7 Com : configuration de la prise console (ASCII, UNI-TELWAY maître ou esclave).

```
Run/stop:
Sécurité:
Mode: Nor-
Période: 2ms
%C4
%C4.P: 9999
Réglage: 0
1App 2Blk 3In
```


- **Editeur terminal (FTX):** Il permet de paramétrer le terminal FTX 117





- 0 Inf : informations sur le terminal,
- 1 Opt : options
- 2 Lan : choix de la langue de travail
- 3 Prf : préférences:
 - activation/inhibition du retour son clavier,
 - choix de la durée du rétro-éclairage de l'écran (économiseur d'écran).
- 4 Tst : lance les autotests du terminal (Réservé Schneider Automation)


```
FTX117 - PL7-07
App= (.....(
FTX
1Opt 2Lan 3Prf
```

1.7 Principe de sélection des éditeurs et des fonctions

Ecran de base

Accessible à tout moment par , il permet la sélection d'un éditeur par son menu:


- soit en positionnant le curseur par   sur l'éditeur désiré et en validant par  (le rectangle précédant l'éditeur doit être plein),
- soit en saisissant directement le numéro associé à l'éditeur (exemple ci-contre  éditeur de configuration).

Le menu visualise 3 éditeurs, pour visualiser les autres éditeurs, utiliser  .

Ecran d'un éditeur (ex : configuration)

L'écran de l'éditeur visualisé dépend du contexte, cela peut être :


- soit le 1^{er} écran de l'éditeur,
- soit le dernier écran visualisé dans l'éditeur,
- soit l'écran correspondant à l'objet pointé dans le mode programmation (voir page suivante).

L'accès aux fonctions de l'éditeur s'effectue par son menu affichable à tout moment par  (les éditeurs Automate et Terminal affiche leur menu de façon permanente).

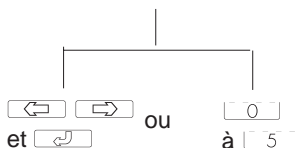
L'accès à une fonction du menu suit le même principe que l'accès d'un éditeur dans le menu de l'écran de base.

 permet de supprimer le menu.

Ecran d'une fonction (ex : Bloc temporisation).

Les valeurs ou les paramètres saisis sont validés par  .

FTX117 - PL7-07
APP= (.....(
TSX 07 31 - 16
02Prg03Dat_4Cnf▶

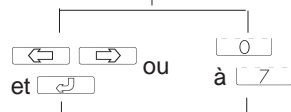


Run/Stop:▶ %I0.1
Securite:▶ %Q0.1
Mode:▶ Normal
Extension:▶ N





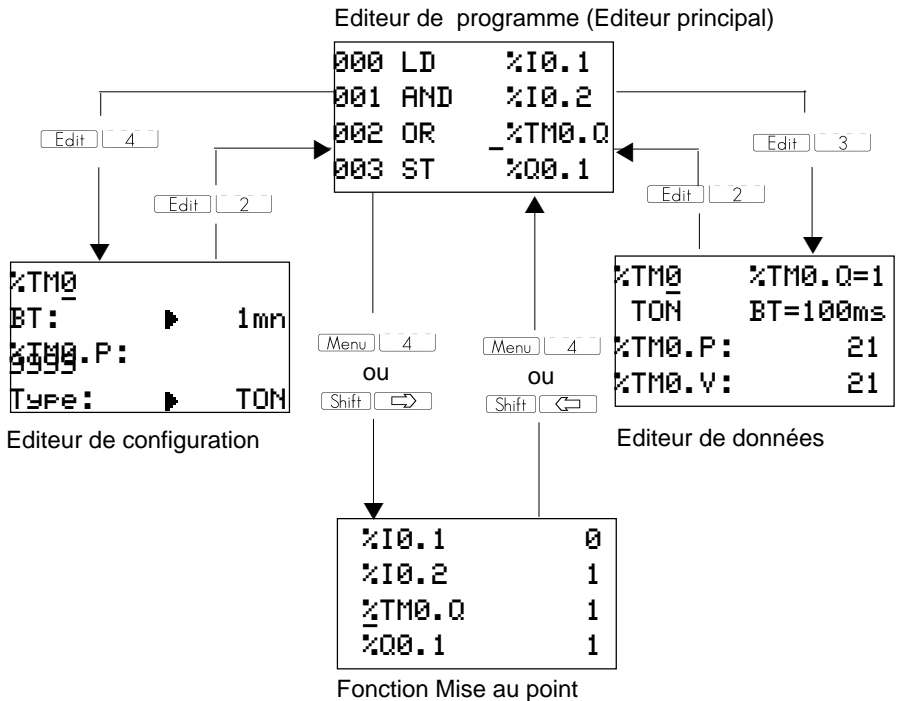
Run/Stop:▶ %I0.1
Securite:▶ %Q0.1
Mode:▶ Normal
01APP_02B1k_03In ▶



%TM2_
BT:▶ 1mn
%TM2.P:
9999
Type:▶ TON

Accès direct entre éditeurs

L'éditeur de programme a été choisi comme éditeur principal, c'est à partir des sélections réalisées dans cet éditeur que peut s'effectuer l'accès direct aux éditeurs de configuration, réglage et à la fonction mise au point.



Exemple ci-dessus : après la sélection (ou saisie) de la temporisation %TM0 dans le programme :

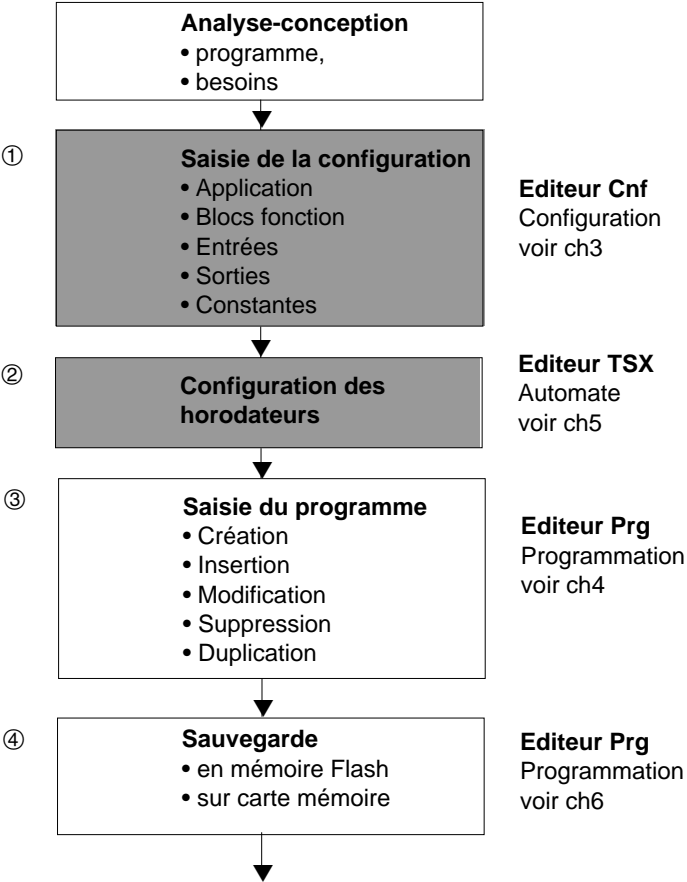
- En phase conception, il est possible d'accéder directement à son écran de configuration pour la paramétrer.
- En phase mise au point, il est possible d'accéder directement à son réglage (éditeur de données) ou à la visualisation de l'état de la sortie (fonction mise au point).

1.8 Méthode pour réaliser une application

L'organigramme suivant propose les différentes étapes pour réaliser une application en langage liste d'instruction à l'aide du terminal FTX 117.

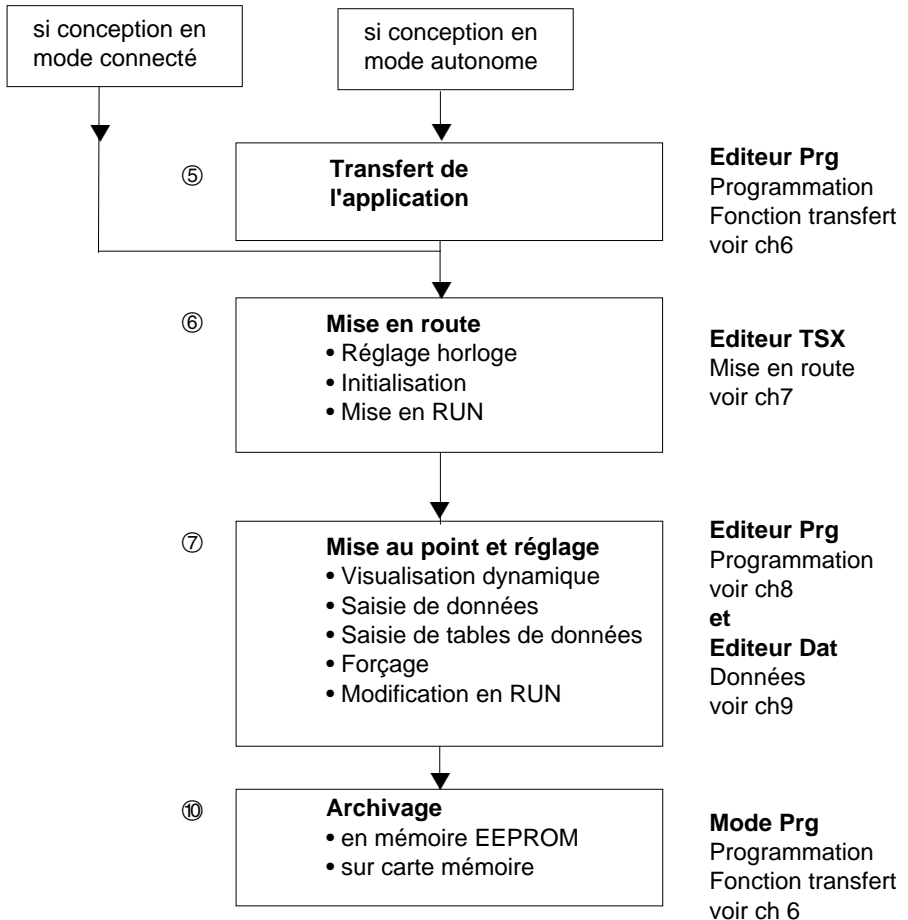
Phase conception , elle peut être effectuée :

- **en mode autonome**, avec (ou sans) automate (commutateur FTX/TSX sur position **FTX**), dans ce cas l'opération de sauvegarde ④ est conseillé, car le terminal a une autonomie de 1heure lorsqu'il n'est plus alimenté.
- **en mode connecté**, avec automate (commutateur FTX/TSX sur position **TSX**), dans ce cas l'opération de sauvegarde ④ est optionnelle.



opération optionnelle si la configuration par défaut convient à l'application à réaliser. Cette opération peut être aussi effectuée en parallèle avec l'opération ③ en utilisant les accès directs (voir ch 1.7).

Phase réglage et mise au point, elle s'effectue : en mode connecté, dans le cas où la phase conception s'est effectuée en connecté, l'opération de transfert ⑤ n'est pas à faire.



Nota : Le terminal offre 2 moyens de protection de l'application lorsque celle-ci a été mise au point :

- une protection totale en lecture et écriture. Elle s'effectue lors du transfert de l'application en mémoire automate , voir ch 6.
- une protection en écriture de l'application, avec accès à la visualisation de l'application et aux réglage des variables. Cette protection s'effectue par mot de passe, voir ch 4.7.

2.1 Mise sous tension du terminal FTX 117

La mise sous tension du terminal s'effectue dès que le raccordement au secteur ou à l'automate est effectué (voir raccordement et modes de fonctionnement ch1.2), un écran d'identification est affiché pendant que s'exécute une série d'auto-tests afin de détecter des dysfonctionnements éventuels.

Après cette série d'auto-tests et si aucun dysfonctionnement n'apparaît, l'écran de base est affiché permettant l'accès à tous les éditeurs du terminal.

- Si le terminal fonctionne en autonome (FTX), l'écran ci-contre est affiché, il mentionne :
 - le type de terminal et le logiciel utilisé,
 - le nom de l'application saisie,
 - le mode de fonctionnement : FTX,
- Si le terminal fonctionne en connecté (TSX), l'écran ci-contre est affiché, il mentionne :
 - le type de terminal et le logiciel utilisé,
 - le nom de l'application saisie,
 - l'automate auquel est connecté le terminal

```
FTX117 V3.0
PL7-07 V3.0
(6) 1994-1996 TE-
```

```
FTX117 - PL7-07
APP: (.....(
FTX
_1TSX_2Pr_3Dat▶
```

```
FTX117 - PL7-07
APP: (.....(
TSX 07 31-16
_1TSX_2Pr_3Dat▶
```

Rappel : le choix du mode de fonctionnement s'effectue à l'aide du commutateur FTX/TSX situé en haut et à droite du clavier.

Nota : lors d'une première mise en route, il est nécessaire de choisir la langue d'utilisation du terminal, l'anglais est sélectionné par défaut (voir ch2.3).

2.2 Mode terminal (5.FTX)

Le mode terminal FTX permet le réglage du terminal FTX 117. L'accès au mode s'effectue par . Il propose 4 fonctions :




Mode Terminal FTX

- • **0 Inf** : Information sur les versions logicielles (voir 1^o écran de cette page)
- • **1 Opt** : Option (implémentation future)
- • **2 Lan** : Langue
- • **3 Prf** : Préférences





```
FTX117 - PL7-07
APP: (.....(
FTX
_10Pt_2Lan_3Prf▶
```


2.3 Choix de la langue (2.Lan)

Menu 2

Cette fonction permet le choix de la langue du terminal.
Le choix s'effectue par  et , il est validé par , il reste mémorisé après coupure secteur.

Choix par défaut : Anglais

 Français
 English
 Deutsch
 Italiano

 Español

2.4 Préférences (3.Prf)

Menu 3

Cette fonction permet :

- de choisir (O) ou pas (N) un signal sonore sur appui des touches clavier.

Le signal sonore a 2 tonalités :

- une tonalité grave indiquant que l'appui sur la touche a été pris en compte,
- une tonalité aigue indiquant l'appui sur une touche non autorisée.

Le choix s'effectue par  ou .

Preferences
Signal sonore:▶0
Retro eclair:▶10

- de choisir la durée au bout de laquelle après aucune action de l'opérateur sur le clavier le rétro-éclairage de l'écran sera rendu inactif.

Ce temps est compris entre 5 et 60mn par pas de 5mn, 0 rend cette fonction inactive.

Le choix s'effectue par  ou .

Cette fonction permet de prolonger la durée de vie du rétro-éclairage (remarque: même éteint, l'écran reste toujours lisible).

Choix par défaut : Signal sonore activé et rétro éclairage fixé à 5mn.

3.1 Généralités

Lors de la réalisation d'une application, l'accès à l'éditeur de configuration n'est pas obligatoire si les valeurs par défaut sont suffisantes (voir tableau ci-dessous). Dans le cas des blocs fonction ou des constantes, l'accès s'effectue de manière plus directe par l'éditeur de programme **Prg** en pointant le bloc ou la constante à l'aide du curseur et en accédant à l'éditeur de configuration (voir accès rapide, ch 3.8).

La configuration de l'application s'effectue par l'éditeur de configuration **Cnf**. L'accès à l'éditeur s'effectue par .

L'éditeur de configuration propose 7 fonctions accessibles à tout instant par :

Editeur de Configuration Cnf

- • **0 Inf** : Information (idem éditeur de programme voir ch4)
- • **1 App** : Application
- • **2 Blk** : Blocs fonctions
- • **3 In** : Entrées
- • **4 Out** : Sorties
- • **5 %KW** : Constantes
- • **6 Clr** : Effacement (mise à la configuration par défaut)
- • **7 Com** : Communication prise terminal

Valeurs par défaut


Fonction	Etat par défaut	Menu	Description de la fonction
Entrée Run/stop	non configurée	1 App	ch4.1 inter A
Sortie Sécurité	non configurée	1 App	ch4.2 inter A
Mode de scrutation	normal	1 App	ch1.3 inter A
Port d'extension	AP	1 App	ch1.10 inter A
Contrôle extension d'E/S	non configuré	1 App	ch1.4 inter A
Filtrage des entrées	12ms	3 In	ch1.7 inter A
Entrées mémorisation d'état	non configurées	3 In	ch4.3 inter A
Compteur rapide	non configuré	3 In	ch4.4 inter A
Sortie %Q0.0	configurée en T.O.R	4 Out	ch1.6 inter A


Nota : les paramètres de configuration ne peuvent pas être modifiés automate en RUN. Après une modification de la configuration la remise en RUN de l'automate provoque une ré-initialisation de l'application (reprise à froid, voir intercalaire A ch 7).


3.2 Configuration générale de l'application (1. App)


Menu

Cette fonction permet :


- de choisir ou pas une entrée de l'automate comme commande de passage **Run/Stop** de l'automate.
 permet le choix d'une entrée de %I0.0 à %I0.5.

- de choisir ou pas une sortie de l'automate comme sortie **Securite** de l'automate.
 permet le choix d'une sortie de %Q0.0 à %Q0.3.

- de sélectionner le mode de scrutation de l'application : **Normal** ou **Periodique** (choix par ). Dans le cas d'une scrutation périodique, une ligne supplémentaire est proposée dans l'écran, elle permet de saisir la période de 1ms à 150ms.

- de sélectionner le type d'extension (choix par )
 - AP (extension automate)
 - Modbus

la sélection AP autorise:

- d'effectuer ou pas le contrôle de l'extension d'entrées/sorties : **Extensions** (choix par ).

Le choix **O** (pour oui) entraîne la détection d'une erreur d'entrées/sorties sur l'automate de base, lorsque :

- l'extension d'entrées/sorties est absente,
- ou lorsqu'un défaut d'entrées/sorties est détecté sur l'extension.

- de choisir la vitesse de transmission 19200 ou 9600 bits/s sur la liaison extension.

- de choisir les équipements à scruter sur la liaison E/S (extension d'entrées/sorties) et AP2 à AP4 automate d'extension.

la sélection Modbus autorise:

- de choisir la vitesse de transmission 1200,2400,4800 9600 ou 19200 bits/s sur le port de communication.

- de sélectionner l'adresse automate (1 à 98).

- de choisir le Time-out (1 à 127 caractères, 3 par défaut).

- de choisir le format (7 ou 8 bits de données, 1 ou 2 bits de stop).

- de choisir le type de parité (paire, impaire ou sans).

Le passage d'une ligne à l'autre de l'écran s'effectue à l'aide de  ou .

 valide les choix effectués.

```
Run/Stop: ▶ %I0.1
Securite: ▶ %Q0.1
Mode: ▶ Normal
Port Ext: ▶ AP
```

```
Run/Stop: ▶ N
Securite: ▶ N
Mode: ▶ Periodique
Periode: 10ms
```

```
Bits/s: ▶ 19200
E/S: ▶ 0
AP2: ▶ 0
AP3: ▶ N
```

```
E/S: ▶ 0
AP2: ▶ 0
AP3: ▶ N
AP4: ▶ N
```

```
Run/Stop: ▶ %I0.1
Securite: ▶ %Q0.1
Mode: ▶ Normal
Port Ext: ▶ Modbus
```

```
Port Ext: ▶ Modbus
Bits/s: ▶ 4800
Adr Esclave: 8
Time Out : 3
```

```
Time Out : 3
Data Bit: ▶ 8
Parite: ▶ Paire
Stop Bit : ▶ 1
```

3.3 Blocs fonction (2. Blk)

Menu 2

Cette fonction permet la configuration des blocs fonction standards :

Bloc fonction	Accès	Description
Temporisateur %TM,	%TM	voir ch 3.2-2
Compteur %C,	%C	voir ch 3.2-3
Registre %R,	Var 0	voir ch 3.2-4
Programmateurs cyclique %DR	Var	voir ch 3.2-5

3.3-1 Bloc temporisateur %TM

La configuration d'un bloc fonction temporisateur s'effectue de la manière suivante :

- 1 %TM donne accès au bloc fonction temporisateur.
- 2 saisir le numéro de bloc (de 0 à 31) à l'aide des touches numériques, ou utiliser et .
- 3 accéder à la base de temps BT par et sélectionner la valeur de la base de temps 1ms (valeur proposée uniquement pour %TM0 et %TM1), 10ms, 100ms, 1s ou 1mn à l'aide de et .
- 4 accéder à la présélection %Tmi.P par et à l'aide des touches numériques, saisir la valeur de présélection de 0 à 9999.
- 5 accéder au Type de temporisateur par et sélectionner le type TON (retard à l'enclenchement), TOF (retard au déclenchement), ou TP (monostable) par les touches et .
- 6 accéder à l'option Reglage par et sélectionner O (pour Oui) ou N (pour Non) par et .
- 7 valider les choix effectués par .

```
%TM2
BT:      1mn
%TM2.P:  9999
Type:    TON
```

```
BT:      1mn
%TM2.P:  9999
Type:    TON
Reglage: 0
```

3.3-2 Bloc compteur %C

La configuration d'un bloc fonction compteur s'effectue de la manière suivante :

- 1 %C donne accès au bloc fonction compteur.
- 2 saisir le numéro de bloc (de 0 à 15) à l'aide des touches numériques, ou utiliser et .
- 3 accéder à la présélection %Ci.P par et à l'aide des touches numériques, saisir la valeur de présélection de 0 à 9999.

```
%C4
%C4.P:  9999
Reglage: 0
```

C

④ accéder à l'option **Reglage** par et sélectionner **O** (pour Oui) ou **N** (pour Non) par et .

⑤ valider les choix effectués par .

3.3-3 Bloc registre %R

La configuration d'un bloc fonction registre s'effectue de la manière suivante :

①

donne accès au bloc fonction registre.

② saisir le numéro de bloc (0 à 3) à l'aide des touches numériques, ou utiliser et .

③ accéder au **Type** de registre par et sélectionner : **FIFO** (premier entré, premier sorti) ou **LIFO** (dernier entré, premier sorti) par et .

④ valider les choix effectués par .

```
%R0
Type:      ▶FIFO
```

3.3-4 Bloc programmeur cyclique %DR

La configuration d'un bloc fonction programmeur cyclique s'effectue de la manière suivante :

①

donne accès au bloc fonction programmeur cyclique.

② saisir le numéro de bloc (0 à 3) à l'aide des touches numériques, ou utiliser et .

③ accéder au nombre de pas **Nb Pas:** par et saisir le nombre de pas configuré de 1 à 8.

④ accéder à chaque **Pas** par et saisir la valeur de chaque bit d'ordre (16 valeurs) à l'aide des touches 0 ou 1, pour chaque pas. L'accès aux différents bits s'effectue par et , le pas est repéré par 2 nombres : **Pas i.j** i est le numéro du pas, j est le bit pointé par le curseur.

⑤ accéder à chaque bit d'ordres par la touche et sélectionner pour chacun d'eux, le type d'objet par ou et son numéro par les touches numériques.

annule un bit d'ordre.

⑥ valider les choix effectués par .

```
%DR0  Nb Pas:8
Pas0.0: 00000000
Pas1.2: 00100000
Pas2.0: 00000000
```

```
Bits d'ordre:
0:%Q0.1 1:%Q1.0
2:%M10  3: ...
4: ...  5: ...
```

```
6: ...  7: ...
8: ...  9: ...
A: ...  B: ...
C: ...  D: ...
```

```
E: ...  F: ...
```


3.4 Configuration des entrées (3 In)






Menu 3

Cette fonction permet de choisir :

- le niveau de filtrage effectué sur les entrées tout ou rien de l'automate,
- les entrées utilisées en entrées mémorisation d'état,
- la fonction comptage rapide.

La validation de ces choix s'effectue par .

3.4-1 Choix du niveau de filtrage sur les entrées Tout ou Rien de l'automate







- le premier choix **Filtrage** permet de valider (choix O) ou d'inhiber (choix N) le filtrage sur les entrées Tout ou Rien de l'automate par  et .
- lorsque le choix Filtrage est validé :  donne accès successivement à 3 groupes d'entrées %I0.0- %I0.3, %I0.4-%I0.7 et %I0.8-%I0.13 pour chacun d'entre-eux, il est possible de sélectionner un filtrage de 3ms, de 12ms ou pas de filtrage (choix N) par  et .

```

Filtrage      ►0
%I0.0-3:     ►12ms
%I0.4-7:     ►12ms
%I0.8-13:    ►12ms
  
```

Choix par défaut : filtrage à 12 ms.

3.4-2 Choix d'utilisation d'entrées à mémorisation d'état

- le choix **Memor.etat** permet de valider (choix O) ou d'inhiber (choix N) l'utilisation d'entrées à mémorisation d'état par  et .
- lorsque le choix mémorisation d'état est validé :  donne accès à 6 entrées de %I0.0 à %I0.5. Pour choisir les entrées, déplacer le curseur successivement sur les entrées désirées par  et  et choisir **O** (oui) par .

```

%I0.8-13:    ►12ms
Memor.etat:  ►0
%I0.0 1 2 3 4 5
      ►N►N►N►N►N►N
  
```

Choix par défaut : entrées sans mémorisation d'état.

3.4-3 Fonction compteur rapide et entrées associées

- **%FC** permet de choisir la fonction à réaliser : **Comptage**, **Fréquence** (fréquencemètre), **Decompt** (comptage/décomptage) ou **N** (fonction non sélectionnée) par et .

%FC: ▶ Comptage
Max Freq.: ▶ 5kHz
Entrée:
Preset : ▶ %I0.1

Une fois la fonction de comptage sélectionnée, les autres choix sont affichés et leur sélection s'effectue en appuyant successivement sur .

- **Max Freq.** : choix de la fréquence maximale **5kHz** ou **10 kHz** de comptage par et .
- **Entrée** : rappelle que l'entrée de comptage est l'entrée **%I0.0** (cette ligne n'offre aucune sélection).
- **Preset** : choix de l'entrée **%I0.1** comme entrée de présélection (ou remise à zéro) par et . **N**= pas d'entrée présélection sélectionnée.
- **Valid** : choix de l'entrée **%I0.2** comme entrée de validation par et . **N**= pas d'entrée validation sélectionnée.
- **Decomptage** : rappelle que l'entrée de décomptage est l'entrée **%I0.3** (cette ligne n'offre aucune sélection). (uniquement lorsque la fonction comptage/décomptage est sélectionnée).
- **Capture** : choix de l'entrée **%I0.4** comme entrée de capture par et . **N**= pas d'entrée validation sélectionnée.
- **%FC.S0** : saisie de la valeur du seuil 0 de 0 à 65535.
- **%FC.S1** : saisie de la valeur du seuil 1 de 0 à 65535.
- **Reflex** : utilisation ou non des sorties réflexes **%Q0.1** et **%Q0.2**. (**N** = pas de sorties réflexes).

Valid: ▶ N
%FC.S0: 65535
%FC.S1: 65535
Reflex: ▶ N

- Les 3 lignes proposées lorsque les sorties réflexes sont sélectionnées correspondent au tableau complet suivant :

Reflex: ▶ %Q0.1-2
%FC.V<S0 >S0 >S1
%Q0.1: 1 0 0
%Q0.2: 1 0 0

	%FC.V<%FC.S0	%FC.S0<%FC.V<%FC.S1	%FC.S1<%FC.V
Etat %Q0.1	1	0	0
Etat %Q0.2	1	0	0

L'accès à la configuration de l'état de chaque sortie s'effectue par et et le choix des états 0 ou 1 par et .

3.5 Configuration des sorties (4. Out)


Menu 4




Cette fonction permet de choisir l'utilisation de la sortie %Q0.0 de l'automate:

- en sortie Tout ou Rien : Normal (choix par défaut),
- en sortie modulation de largeur %PWM,
- en sortie générateur d'impulsion %PLS.

```
%Q0.0:  Normal
```


3.5-1 Utilisation en sortie modulation de largeur

Le choix %PWM obtenu par , sélectionne la sortie %Q0.0 en sortie modulation de largeur et donne accès à la configuration du bloc fonction PWM


- ① accéder à la base de temps BT du bloc %PWM par  et sélectionner la base de temps 0.2ms, 10ms, 1s par  et .




```
%Q0.0:  %PWM
BT:      1s
%PWM.P:  0
```

- ② accéder à la présélection %PWM.P par  et à l'aide des touches numériques, saisir la valeur de présélection de 0 à 32767.


- ③ valider les choix effectués par .

3.5-2 Utilisation en sortie générateur d'impulsions




Le choix %PLS obtenu par , sélectionne la sortie %Q0.0 en sortie générateur d'impulsions et donne accès à la configuration du bloc fonction générateur d'impulsions.




- ① accéder à la base de temps BT du bloc %PLS par  et sélectionner la base de temps 0.2ms, 10ms, 1s ou par  et .

```
%Q0.0:  %PLS
BT:      1s
%PLS.P:  32767
Comptage PLS:  N
```

- ② accéder à la présélection %PLS.P par  et à l'aide des touches numériques, saisir la valeur de présélection de 0 à 32767.

```
BT:      1s
%PLS.P:  32767
Comptage PLS:  N
Reglage:  N
```

- ③ accéder à l'option **Comptage PLS** (dans le cas d'un rebouclage de la sortie %Q0.0 sur l'entrée %I0.0) par  et sélectionner **O** (pour Oui) ou **N** (pour Non) par  et .



- ④ accéder à l'option **Reglage** par  et sélectionner **O** (pour Oui) ou **N** (pour Non) par  et .

- ⑤ valider les choix effectués par .

3.6 Saisie des constantes (5 %KW)

[Menu] [5]

L'accès aux constantes désirées, s'effectue :

- soit par  et 
- soit en appuyant sur la touche [%KW], en saisissant le numéro de la constante de 0 à 63 et en validant par .

%KW30	-	0
%KW31		0
%KW32		0
%KW33		0

La saisie de la valeur s'effectue par le clavier numérique, une valeur peut être saisie en décimal, ou en hexadécimal par [#] (ex: [#] [F] [5] [0] [0] pour 16#F500, le 16 est rajouté automatiquement par le terminal).

```
%KW_  
  
Saisir variable
```

Chaque valeur saisie doit être validée par .

3.7 Effacement de la configuration (6 Clr)

[Menu] [6]

[Menu] [6] donne accès à l'écran de confirmation d'effacement de la configuration de l'application, cette fonction remet tous les paramètres de configuration aux valeurs par défaut.

```
Effacer  
Conf?  
0:Enter N:Esc
```

 exécute l'effacement.

[Esc] annule la demande d'effacement.

3.8 Accès direct à la configuration depuis l'éditeur de programme

Cas des blocs fonction

Lorsqu'un bloc fonction est pointé par le curseur dans l'éditeur de programme, l'accès à l'éditeur de configuration s'effectue directement sur l'écran de configuration de ce bloc. La configuration du bloc est ensuite identique à celle décrite ch3.3, les opérations ① et ② ne sont plus nécessaires.

Cas des constantes

Lorsqu'une constante est pointée par le curseur dans l'éditeur de programme, l'accès à l'éditeur de configuration s'effectue directement sur l'écran de configuration des constantes. La saisie des constantes est ensuite identique à celle décrite ch3.6.

3.9 Configuration prise terminal (7 Com)

L'accès aux modes de fonctionnement de la prise terminal s'effectue en sélectionnant

  de l'éditeur Cnf.

Mode ASCII (choix par)

Cet écran permet:

- de choisir la vitesse de transmission 1200, 2400, 4800, 9600 bits/s sur la liaison terminal.
- de choisir le format (7 ou 8 bits de données, 1 ou 2 bits de stop).
- de choisir le Time-Out de l'équipement maître (30 à 255 caractères).
Ce champ est spécifique au mode UNI-TELWAY.
- de choisir le type de parité (paire, impaire ou sans).

```
Port COM: ▶ ASCII
Bits/s: ▶ 9600
Time Out: ▶ 50
Data Bit: ▶ 8
```

```
Time Out: ▶ 50
Data Bit: ▶ 8
Parite: ▶ Sans
Stop Bit: ▶ 2
```

Mode UNI-TELWAY maître

L'écran de configuration est celui utilisé par le mode ASCII.



Mode UNI-TELWAY esclave


Cet écran permet:

- de choisir la vitesse de transmission 1200, 2400, 4800, 9600 bits/s sur la liaison terminal.
- de saisir le numéro d'adresse de l'équipement esclave (1 à 97, 4 par défaut).
- de choisir le format (7 ou 8 bits de données, 1 ou 2 bits de stop).
- de renseigner le Time-Out de l'équipement maître (30 à 255 caractères).
- de choisir le type de parité (paire, impaire ou sans).

```
Port COM: ▶ UTLW
Bits/s: ▶ 9600
Adr Esclave: 4
Time Out: ▶ 50
```

```
Time Out: ▶ 50
Data Bit: ▶ 8
Parite: ▶ Sans
Stop Bit: ▶ 2
```

Le passage d'une ligne à l'autre de l'écran s'effectue à l'aide de  ou .

 valide les choix effectués.

C

4.1 Généralités

La saisie et la modification du programme s'effectuent par l'éditeur de programme. L'accès à l'éditeur se fait par .

Le programme est validé et mémorisé à chaque validation d'une ligne par .

Le menu, accessible par , offre des fonctions supplémentaires d'aide à la programmation.

Editeur de Programme Prg

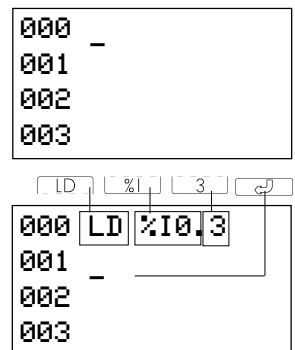
- • **0 Inf** : Saisie du nom, du mot de passe et informations sur l'application (voir ch4.7)
- • **1 Trf** : Transfert (voir ch6)
- • **2 Clr** : Effacement du programme et de la configuration (voir ch 4.6)
- • **3 Dgn** : Diagnostic, contrôle de cohérence du programme (voir ch 4.5)
- • **5 Sch** : Recherche/remplacement (voir ch 4.4)
- • **6 Sel** : Sélection de lignes de programme (voir ch 4.3)
- • **7 Pst** : Collage de lignes de programme (voir ch 4.3)

4.2 Création d'un programme

Lors de l'accès à l'éditeur de programme, le terminal propose un écran vierge comportant uniquement les numéros de ligne.

La saisie d'une instruction et de son opérande s'effectue simplement en appuyant successivement :

- sur la touche de l'instruction (ex :) ,
 - sur la touche de l'opérande (ex :) ,
 - et en saisissant le numéro (ou l'adresse) de l'opérande (ex :) ,
 - en validant la ligne de programme par .
- Le curseur passe alors à la ligne suivante.



Nota : les instructions et les opérandes sont regroupées sur les mêmes touches, les instructions sont en jaune et les opérandes en blanc. Le choix entre instruction ou opérande s'effectue automatiquement par le terminal suivant que le curseur pointe une zone instruction ou une zone opérande.

Cas d'instructions utilisant un modificateur

Les instructions LDN, LDR, JMP... sont composées d'une instruction élémentaire (ex: LD) suivie d'un modificateur (ex : N). Une telle instruction se programme en appuyant sur la touche du modificateur après avoir appuyé sur la touche de l'instruction.

Exemples (voir écran ci-contre) :

[LD] [R/F] [%I] [2] [↵]
[AND] [N] [%M] [1] [↵]
[AND] [] [N] [%I] [3] [↵]
[OR] [R/F] [R/F] [%I] [4] [↵]

```
000 LDR %IO.2
001 ANDN %M1
002 AND(N %IO.3
003 ORF IO.4_
```

Cas d'instructions ou de variables non mentionnées sur le clavier

Ces instructions ou variables sont accessibles par [Fun] pour les instructions ou [Var] pour les variables. Fun et Var sont sur la même touche, le choix s'effectue automatiquement par le terminal suivant que le curseur pointe une zone instruction ou une zone opérande.

Principe d'accès :

- accès direct : [Fun] ou [Var] + Numéro + [↵]
 - accès indirect : [Fun] ou [Var] + [↵] donne accès aux écrans d'instructions ou variables (voir écrans ci-dessous). 2 possibilités :
 - choix du Numéro,
 - déplacement du curseur sur l'instruction ou la variable choisie puis [↵],
- [Shift] [↵] et [Shift] [↵] permet la visualisation des premiers ou derniers éléments de la liste.

Instructions booléennes

```
1 SR      5 U
2 RET     6 MCS
3 I       7 MCR
4 O       8 ==
```

```
9  --
10 #D
11 ==P
```

Instructions de comparaison

```
1 =      5 <=
2 <>     6 >=
3 <
4 >
```

Instructions arithmétiques et logiques

```
1 :=      5 /
2 +       6 REM
3 -       7 INC
4 *       8 DEC
```

```
9 AND     13 SHL
10 OR      14 SHR
11 XOR     15 ROL
12 NOT     16 ROR
```

```
17 BTI    21 READ
18 ITB
19 SORT
20 EXCH
```

Variables

```
1 %IW     5 %FC
2 %QW     6 %PLS
3 %SBR    7 %PWM
4 %SC     8 %MSG
```

```
9 %X      13 D
10 %R      14 F
11 %DR     15 Q
12 E       16 R
```

```
13 D      17 TH0
14 F      18 TH1
15 Q
16 R
```


Autre saisie possible

L'accès à l'instruction ou à la variable désirée peut s'effectuer directement en saisissant son numéro après avoir appuyé sur **Fun** ou **Var**. La liste des numéros est donnée en Annexes, elle apparait aussi dans les écrans de la page précédente.

Exemple : **Fun**

permet d'accéder directement à l'instruction SR.

```
000 Fun 1
001
```

```
000 SR _
001
```

Cas d'instructions numériques

Ces instructions se programment de la façon suivante :

- ① Appuyer sur (ou sur l'instruction booléenne LD, AND, ou OR puis sur pour les instructions de comparaison), Fun est alors affiché automatiquement,
- ② Appuyer sur et choisir l'instruction sur les écrans proposés (ou saisir directement le numéro de l'instruction puis sur), l'instruction est alors visualisée et le curseur pointe sur le premier champ opérande.
- ③ Saisir le premier opérande à l'aide des touches opérandes ou de la fonction Var, appuyer sur pour accéder aux opérandes suivants et les saisir ,
- ④ Valider la saisie en appuyant sur

```
000 [
      Fun _
001
002
```

1	:=	5	/
2	+	6	REM
3	-	7	INC
4	*	8	DEC

```
000 [  _...
      := ...
]    +  ...
002
```

Exemple (voir écran ci-contre) :

μ

μ

```
000 [  %MW10
      := %MW0
1    +  100
002  _
```

4.3 Modification d'un programme

Nota : dans le cas où un mot de passe a été défini, les modifications ne pourront être effectuées que si ce dernier a été préalablement saisi (voir ch 4.7).

4.3-1 Accès à une ligne de programme

- déplacer le curseur sur la ligne à atteindre à l'aide des touches :

- , pour un déplacement ligne par ligne,
- pour atteindre le début du programme,
- pour atteindre la fin du programme,
- **<n°ligne>**

pour accéder à un numéro de ligne donné (voir ch 4.4).

4.3-2 Modification d'une ligne de programme

- déplacer le curseur sur la ligne à atteindre,
- positionner le curseur sur l'élément à modifier par et .

- la modification du numéro d'un opérande ne modifie ni l'instruction ni l'opérande,
- la modification de l'opérande ne modifie pas l'instruction,
- par contre la modification de l'instruction modifie la ligne complète.

- la modification est prise en compte en validant la ligne de programme par .

Le curseur passe alors à la ligne suivante.

permet d'annuler la modification en cours (avant validation par).

Exemple (voir écrans ci-contre) : remplacer dans la ligne 001 l'opérande %I0.2 par l'opérande %I0.3 :

- pour atteindre le début du programme,
- pour un déplacement sur la ligne 001,
- 5 fois pour se positionner sur l'adresse 2,
- pour modifier l'adresse.
- pour valider.

```
010 END_BLK
011 LD    %M10
012 AND   %I0.4
013 ANDF  %I0.6
```

```
000 BLK   %TM0
001 LDN   %I0.2
002 AND(N %I0.5
003 ORF   %I0.6
```

et 5 fois

```
000 BLK   %TM0
001 LDN   %I0.2
002 AND(N %I0.5
003 ORF   %I0.6
```

```
000 BLK   %TM0
001 LDN   %I0.3
002 AND(N %I0.5
003 ORF   %I0.6
```

4.3-3 Insertion d'une ligne de programme

- déplacer le curseur sur la ligne située après la ligne à insérer,
- insère une ligne blanche, dans laquelle il est alors nécessaire de saisir une instruction et son opérande.
- valide la nouvelle ligne de programme et insère une nouvelle ligne blanche dans laquelle il est possible de saisir une nouvelle instruction.
- termine l'opération d'insertion.

```
000 BLK  %TM0
001 LDN  %I0.2
002 AND(N %I0.5
003 ORF  %I0.6
```

```
000 BLK  %TM0
001 LDN  %I0.2
002
003 AND(N %I0.5
```

Exemple (voir écrans ci-contre) : insérer à la ligne 002 l'instruction AND %M0.

Le curseur est positionné au départ sur la ligne 000.

- 2 fois pour atteindre la ligne n°2,
- pour insérer la ligne,
- saisie de l'instruction,
- pour valider,
- pour terminer l'opération d'insertion.

```
001 LDN  %I0.2
002 AND  %M0
003
004 AND(N %I0.5
```

```
001 LDN  %I0.2
002 AND  %M0
003 AND(N %I0.5
004 ORF  %I0.6
```

4.3-4 Effacement d'une ligne de programme

- déplacer le curseur sur la ligne à effacer,
- supprime la ligne, toutes les lignes de programmation sont alors décalées vers le haut.
- annule cette opération, la ligne effacée est alors restituée.

4.3-5 Duplication, suppression et déplacement d'un ensemble de lignes

Ces opérations sont réalisées à partir des 2 fonctions de bases décrites en bas de page:

- sélection d'un ensemble de lignes : fonction **Sel** Menu 6
- collage d'un ensemble de lignes : fonction **Pst** Menu 7

Ces fonctions ne sont accessibles qu'en mode autonome.

Duplication d'un ensemble de lignes :

- ① sélection des lignes à dupliquer,
- ② validation par Enter,
- ③ positionnement sur le numéro de ligne où doivent être collées les lignes,
- ④ collage des lignes.

Suppression d'un ensemble de lignes :

- ① sélection des lignes à supprimer,
- ② suppression par Shift Del,

Déplacement d'un ensemble de lignes :

- ① sélection des lignes à dupliquer,
- ② suppression par Shift Del,
- ③ positionnement sur le numéro de ligne où doivent être collées les lignes,
- ④ collage des lignes.

Sélection d'un ensemble de lignes de programme ①

- déplacer le curseur sur la première ligne à sélectionner,
- Menu 6 lance la fonction **Sel** (sélection des lignes de programme),
- déplacer le curseur sur la dernière ligne à sélectionner, chaque ligne sélectionnée est repérée par une astérisque.

```
000 LDR %I0.2
001*AND %I0.3
002*ORF %I0.4
003*AND %I0.5
```

Collage d'un ensemble de lignes de programme ②

- Menu 7 lance la fonction **Pst** : collage des lignes de programme précédemment sélectionnées, la suite du programme est décalée d'autant vers le bas.

4.4 Recherche/remplacement (5 Sch)

Menu 5

Les opérations de recherche/remplacement sont accessibles à partir de la fonction Recherche **5.Sch** du menu par Menu 5. Cette fonction propose le sous menu suivant :

Fonction : Recherche

Menu 5

- **1 Nxt** : poursuite d'une recherche en cours
- **2 Pos** : recherche d'une ligne, d'une étiquette (label) ou d'un sous programme
- **3 Var** : recherche d'une variable
- **4 Ins** : recherche d'une instruction
- **5 Rep** : remplacement

4.4-1 Recherche d'une ligne de programme, d'une étiquette (label) ou d'un sous programme : 2 Pos

Menu 5 2

La première ligne de l'écran donne accès à la saisie du numéro de ligne, de label ou de sous programme à atteindre (suivant la fonction Chercher).

↕ donne accès au type de recherche :

- Ligne,
- Label (étiquette),
- SR (sous programme),

Choisir le type de recherche par → et ←.

↪ lance la recherche. Le curseur se positionne alors sur la ligne recherchée.

Si la ligne demandée ne contient aucune instruction, un écran prévient que la ligne est non trouvée.

↪ donne de nouveau accès à l'écran de recherche,

Esc retourne à l'écran de visualisation du programme.

```
Num: 3
Chercher: ▶ Ligne
```


```
Num: 5
Chercher: ▶ Label
```

```
Num: 5
Chercher: ▶ SR
```

4.4-2 Recherche d'une variable dans le programme : (3 Var)

Menu 5 3

Saisir la variable recherchée,

 lance la recherche. Le curseur se positionne alors sur la variable recherchée.

Lorsque la variable n'est pas présente dans le programme, le message NON TROUVE est affiché.

Menu 5 1 poursuit la recherche sur la prochaine variable, cette recherche peut se poursuivre jusqu'à la fin du programme.

Les "X" sous le nom de la variable signifient que le caractère au-dessous est pris en compte dans la recherche. L'absence de X signifie que le caractère n'est pas pris en compte.

```
Chercher Var
%00.1_
XXXXXXXXXXXX
```

```
Chercher Var
%00.1
XXXX_XXXXX
```

Exemple de l'écran ci-contre : la recherche s'effectue sur toutes les sorties de l'automate de base .


Pour supprimer ou remplacer un caractère "X", utiliser

Ins Del.

4.4-3 Recherche d'une instruction dans le programme : (4 Ins)

Menu 5 4

Se positionner dans le programme sur la ligne à chercher, (la saisir au besoin).

 lance la recherche. Le curseur se positionne alors sur l'instruction recherchée.

Menu 5 1 poursuit la recherche sur la prochaine instruction, cette recherche peut se poursuivre jusqu'à la fin du programme.

Les "X" sous le nom de l'instruction signifient que le caractère au-dessous est pris en compte dans la recherche. L'absence de X signifie que le caractère n'est pas pris en compte.

```
Chercher Inst
ST %00.1
XXXXXXXXXXXXXXXXX
```

```
Chercher Inst
ST %00.1
XXXXXXXXX_
```

Exemple de l'écran ci-contre : la recherche s'effectue sur toutes les instructions ST.

Pour supprimer ou remplacer un caractère "X", utiliser

Ins Del.

4.4.4 Remplacement d'une variable dans le programme : (5 Rep)

Saisir la variable à remplacer (la dernière variable recherchée apparaît par défaut),

Saisir la variable de remplacement (accès par) ,

Sélectionner l'option de remplacement :

- **Partout** pour un remplacement dans tout le programme,
- **Suivant** pour rechercher la variable suivante et la remplacer.

Accès par et pour faire le choix.

lance le remplacement sur la première variable recherchée rencontrée ou dans tout le programme suivant le cas.

```
Remplacer:
      %I0.3
Par:
▶Suivant
```

```
Remplacer:
      %I0.3
Par: %I0.5_
▶Partout
```

Nota :

- cette fonction n'est pas accessible lorsque l'automate est en exécution RUN.
- le remplacement ne s'effectue que sur des variables de même type.

4.5 Diagnostic Programme (3 Dgn)

Menu 3

4.5-1 Mode autonome ou mode connecté automate en STOP

Le diagnostic est lancé sur :

- demande de l'utilisateur Menu 3 (Dgn)
- demande de transfert de l'application Menu 1 (Trf)
- demande de sortie de l'éditeur programme Edit

permettant ainsi d'effectuer en une seule fois la vérification du programme.

Note : Avec un terminal FTX 117 version inférieure ou égale à V2.1, le diagnostic est lancé uniquement sur demande de l'utilisateur Menu 3 (Dgn).

Le diagnostic est lancé comme indiqué précédemment, le message "Diagnostic_" est visualisé sur la 4^o ligne de l'écran.

En fin de diagnostic :

Aucune erreur n'est détectée, l'écran se positionne sur l'écran demandé. Dans le cas où le diagnostic est lancé par **3 Dgn**, le message "Diagnostic OK" est visualisé.

Une erreur est détectée :

- le message "Diagnostic n" est visualisé où n indique le numéro d'erreur (voir liste en annexe A4 intercalaire G),
- la ligne programme incriminée est visualisée sur la 1^o de l'écran
 pour continuer l'action initiale demandée soit :
- si demande initiale **3 Dgn** retour à l'éditeur de programme,
- si demande initiale **1 Trf 1 Trf** accès à la sauvegarde et au transfert d'application,
- si demande initiale **Edit** retour au choix des éditeurs.

Note : Dans le cas où le diagnostic est lancé par **1 Trf** ou **Edit**, Esc permet d'abandonner l'action demandée et redonne l'accès à l'éditeur de programme pour correction éventuelle.

```
000 LDR %IO.2
001 ANDCN %IO.3
002 ORF %IO.4
Diagnostic_
```

```
000 LDR %IO.2
001 ANDCN %IO.3
002 ORF %IO.4
Diagnostic OK
```

```
010 ANDN %IO.3
011 AND %M5
012>
Diagnostic 7
```

4.5-2 Mode connecté automate en RUN



Le programme est exécutable. Les modifications de programme autorisées en RUN ne pouvant pas entraîner un programme non exécutable, le diagnostic est de ce fait toujours correct.


4.6 Effacement d'une application (2 Clr)

Menu 2

Cette opération efface dans la mémoire terminal (mode FTX) ou dans la mémoire automate (mode TSX) :

- soit le programme seul,
- soit le programme et la configuration.

 et  permettent de faire le choix entre l'effacement du programme "Prog" et l'effacement du programme et de la configuration : "Prog + Conf" .

 donne accès à l'utilisation des instructions relais maître MCR/MCS

 et  permettent de faire le choix :



- O avec utilisation des instructions MCR/MCS
- N sans utilisation des instructions MCR/MCS


Nota :

- le choix N (sans instructions MCS/MCR) permet d'avoir des temps d'exécution de 1,5 à 4 fois plus rapides (voir Annexe G A.5).

 demande l'effacement.

Un écran de confirmation est proposé :

 et  permettent de faire le choix entre **O** (Oui) et **N** (Non).

 lance l'effacement et provoque le retour à la visualisation du programme.

Nota : cette fonction est valide uniquement automate en STOP.

```
Effacer:
Prog + Conf
TSX07 30/31
MCR: 0
```

```
Effacer:
Prog + Conf
Enter N:Esc
```

4.7 Information (0 Inf)

Menu 0

Cet écran donne accès :

- à la saisie du nom de l'application,
- à la création d'un mot de passe et à sa saisie,
- à la visualisation d'informations sur l'application.

```
Pass:(          (
APP:(.....(
Exec:N    MST:N
Libre 6170
```

4.7-1 Saisie du nom de l'application

Ce nom permet d'identifier l'application, il a 8 caractères maximum. Il se saisit à partir du clavier, avec les touches:

- 0 à 9
- A à F (accès à l'aide de Shift)
- et , +/ -, . , :

```
Pass:(          (
APP:(....FAB5(
Exec:N    MST:N
Libre 6172
```

4.7-2 Mot de passe

Rôle du mot de passe :

Le mot de passe permet de protéger une application afin que celle -ci ne soit pas modifiée ou détruite. Quand une application contient un mot de passe, l'utilisateur du terminal qui n'a pas saisi ce mot de passe, se voit limiter les services offerts uniquement en consultation de l'application et en réglages des données.

Le mot de passe est demandé à chaque initialisation du terminal et à chaque changement de mode FTX/TSX.

Création d'un mot de passe :

Lorsque aucun mot de passe n'a été créé, le champ mot de passe (Pass:) est vide.

donne accès à sa création.


Il a 8 caractères maximum. Il se saisit à partir du clavier, avec les touches:

- 0 à 9
- A à F (accès à l'aide de Shift).

```
Pass:(ABC_    (
APP:(....FAB5(
Exec:N    MST:N
Libre 6172
```

Saisie du mot de passe :

Lorsque le terminal est connecté, initialisé ou après un changement de mode local/connecté. Le terminal demande de façon automatique la saisie du mot de passe.

L'utilisateur a alors le choix de le saisir ou pas. Dans les 2 cas  permet d'accéder à l'écran de base.


```
Pass:(?????????(<
App:(....FAB5(<
Exec:N    MST:N
Libre 6172
```

Toute saisie ultérieure peut être effectuée à partir de l'écran d'informations.

Remarque : la saisie d'un mot de passe erroné est équivalente à l'absence de saisie de mot de passe. Lorsque le mot de passe correct est saisi et validé, il doit ré-apparaître en clair à chaque nouvel accès à l'écran d'informations.

```
Pass:(ABC    (<
App:(....FAB5(<
Exec:N    MST:N
Libre 6172
```

Modification du mot de passe :

Pour modifier un mot de passe, il suffit de saisir l'ancien mot de passe, le valider par  . Puis d'accéder de nouveau à l'écran d'informations (l'ancien mot de passe doit apparaître en clair) et de saisir le nouveau mot de passe.

4.7-3 Informations visualisées

Informations visualisées :

- 3° ligne de l'écran :
 - Exec :
 - O = application exécutable
 - N = application non exécutable
 - MST : indique si le programme a été enregistré en mémoire EEPROM de l'automate avec l'option "chargement automatique" (voir ch6).
- 4° ligne de l'écran :
 - indique la taille de la mémoire application libre en octets.

```
Pass:(?????????(<
App:(....FAB5(<
Exec:N    MST:N
Libre 6172
```


5.1 Configuration des horodateurs

L'accès à la configuration des horodateurs s'effectue par les touches :

Edit accès au mode TSX

Menu accès à la fonction Horodateur

Procédure de configuration d'un horodateur

- ① Saisir le numéro d'horodateur à configurer (0 à 15),
- ② donne accès à la sortie à activer, cette sortie peut être :
 - une sortie Tout ou Rien : appuyer sur et saisir l'adresse,
 - un bit interne : appuyer sur et saisir le numéro.
- ③ donne accès à la saisie du jour (1 à 31) de début d'activation.
- ④ donne accès au mois de début d'activation, choix par ou .
- ⑤ donne accès à la saisie du jour (1 à 31) de fin d'activation.
- ⑥ donne accès au choix du mois de fin d'activation, choix par ou .
- ⑦ donne accès au choix des jours d'activation dans la semaine, permettent de choisir ou pas le jour (L = Lundi, M=Mardi...,D=Dimanche), ou permettent le déplacement du curseur.
- ⑧ donne accès à la saisie de l'heure de début d'activation.
- ⑨ donne accès à la saisie de l'heure de fin d'activation.
- ⑩ valide la configuration du bloc.

```
RTC: 5   Q:
1 ▶Jan->31 ▶Dec
   LMMJVSD
   00:00->23:59
```

```
RTC:0051
4 ▶Avr->30 ▶Mai
   LMMJV..
   10:00->13:00
```

Exemple :

La sortie %Q0.1 est activée de 10h à 13h les lundi, mardi, mercredi, jeudi, vendredi du 4 Avril au 30 Mai.

Après mise à l'heure de l'horloge , il suffit de mettre l'automate en RUN pour que les horodateurs remplissent leur fonction.

6.1 Généralités

L'accès à la sauvegarde et au transfert d'application s'effectue par :

- accès à l'éditeur de programme,
- accès à la fonction Transfert.

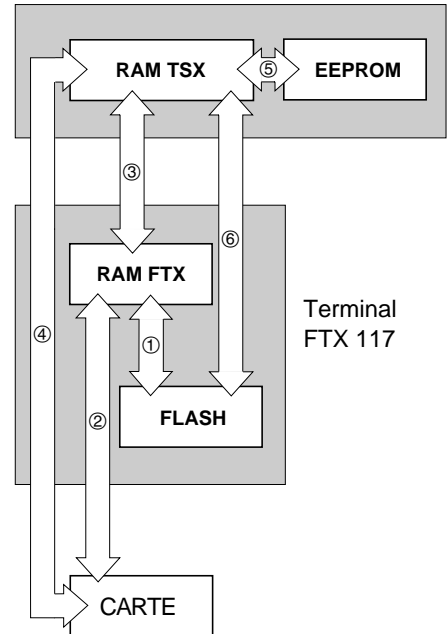
```

▶Transférer App
Source:▶TSX
Dest:▶FTX
    
```

Description des zones mémoire :

- **RAM du terminal FTX**, correspond à la mémoire de travail en mode local, elle est sauvegardée pendant une heure lorsque le terminal est mis hors tension.
- **FLASH**, zone mémoire permanente du terminal où peut être sauvegardée l'application en cours.
- **CARTE**, carte mémoire où peuvent être sauvegardées ou archivées plusieurs applications.
- **RAM de l'automate TSX**, correspond à la zone de travail en mode connecté et à la zone dans laquelle sera exécuté le programme, cette zone est sauvegardée par une pile, elle peut néanmoins être altérée (usure de la pile, mauvais transfert...).
- **EEPROM**, mémoire permanente dans laquelle peut être sauvegardée l'application. Une option permet un rechargement automatique par le contenu de l'EEPROM de la RAM automate lorsque celle-ci est différente du contenu de la mémoire EEPROM.

Automate TSX Nano












Rôle des échanges :

- ① Sauvegarde (restitution) de l'application en cours de conception, en mode local, voir ch6.3,
- ② Sauvegarde (restitution) de l'application en cours de conception, en mode connecté, voir ch6.4,
- ③ Transfert de l'application pour mise au point en mode connecté voir ch6.5,
- ④ Archivage (restitution) de l'application après mise au point, voir ch6.4,
- ⑤ Archivage de l'application après mise au point avec possibilité de chargement automatique, voir ch6.6,
- ⑥ Sauvegarde (restitution) de l'application en cours de conception, voir ch6.3.

6.2 Principe d'utilisation de la fonction Transfert

Quel que soit le transfert à effectuer le principe d'utilisation de la fonction transfert est toujours identique :

- ①  ou  permettent de choisir entre la fonction transfert et la fonction comparaison,
- ②  donne accès au choix de la source (mémoire où se trouve le programme),
 ou  permettent de choisir la source.
- ③  donne accès au choix de la destination (mémoire où le programme sera transféré),
 ou  permettent de choisir la destination.
- ④  lance le transfert.

```
►Transférer APP  
Source:►TSX  
Dest:►FTX
```

Suivant le type de transfert effectué, des options peuvent être proposées (voir pour cela le sous chapitre correspondant).

Nota :

- Tous les transferts vers l'automate (destination TSX) ne peuvent s'effectuer que si le terminal est en mode connecté (commutateur FTX/TSX sur TSX) et l'automate est en STOP.
- En cas d'erreur de communication pendant le transfert , il est nécessaire d'effacer la mémoire automate et d'effectuer un nouveau transfert, l'automate ne pourra pas être mis en RUN tant que ce transfert n'a pas été de nouveau effectué..

Fonction comparaison : cette fonction permet de vérifier que les applications situées sur 2 supports différents (Source et Destination) sont identiques. L'opération 4 lance la comparaison, si les 2 applications sont identiques le message "Comparer OK" est affiché sinon le message "Comparer NOK" est affiché.

```
►Comparer APP  
Source:►TSX  
Dest:►FTX  
Comparer OK
```

6.3 Sauvegarde/restitution de l'application en mémoire FLASH

Utilisation principale : sauvegarde de l'application en cours de saisie lorsque le terminal est en mode local.

- Pour une sauvegarde :
Source : FTX (1)
Destination : FLASH
- Pour une restitution :
Source : FLASH
Destination : FTX (1)

```
►Transférer APP  
Source:►FTX  
Dest:►FLASH
```

(1) Le transfert TSX <--> mémoire FLASH est aussi possible.








6.4 Sauvegarde/restitution de l'application sur carte mémoire

Utilisation principale : sauvegarde ou archivage de l'application lorsque le terminal est en mode local ou connecté.


- Pour une sauvegarde :
Source : FTX ou TSX
Destination : Carte
- Pour une restitution :
Source : Carte
Destination : FTX ou TSX

```
►Transférer App
Source:►FTX
Dest:►Carte
ARCH:
```

Après avoir choisi la destination (voir ch 6.2) :

- ④  donne accès au répertoire de la carte (1), choix par , ,  ou  du numéro d'archive.
- ⑤  valide le choix.
- ⑥  lance le transfert.

```
ARCH0 P ARCH4 P
ARCH1 P ARCH5
ARCH2 P ARCH6
ARCH3 P ARCH7
```

ou saisir directement le numéro d'archive,  lance le transfert


Le "p" à coté de chaque nom d'archive indique qu'une application a déjà été sauvegardée. La sauvegarde d'une application sur une archive déjà utilisée détruit cette dernière.

```
►Transférer App
Source:►FTX
Dest:►Carte
Erreur Carte 3
```

(1) un message "Erreur Carte" suivi d'un code, peut apparaître dans les cas suivants :

Code	Cause
1	aucune carte n'est insérée dans le terminal
2	le type de carte n'est pas reconnu
3	carte non formatée (voir effacement ci après)
4	la carte n'est pas compatible
5	la carte est protégée en écriture
6	la carte insérée n'est pas une carte Telemecanique
7	recharger la batterie
53	carte non formatée (voir effacement ci après)
54	erreur de traitement

Effacement d'une carte

Après l'affichage du code erreur 3 ou 53,  donne accès à l'écran de confirmation d'effacement.

 lance l'effacement,

 annule l'opération d'effacement.

```
Effacer Carte?
O:Enter N:Esc
```

Remarque : le nombre d'archives offertes par une carte dépend de sa taille (en Kmot de 16 bits), elle est donnée par la formule suivante :

Nombre d'archives = (Taille/4) - 2





6.5 Transfert de l'application Terminal/automate

Utilisation principale : après avoir saisi le programme en mode local sur le terminal, permet de transférer celui-ci vers l'automate.



- Pour un transfert Terminal -> Automate :
Source : FTX
Destination : TSX
- Pour un transfert Automate -> Terminal :
Source : TSX
Destination : FTX

```
►Transférer App  
Source:►FTX  
Dest:►TSX  
Prot:►N
```

Dans le cas d'un transfert FTX vers TSX après le choix de la destination : (voir ch 6.2)

- ④  donne accès à l'option de protection en lecture du programme par  ou  .
- ⑤  lance le transfert.

Lorsque l'application est protégée en lecture, les éditeurs de programme et de configuration ne sont plus accessibles et l'écran ci-contre est proposé.

-  efface l'application après deux confirmations
-  retourne à l'écran de base.

```
Acces refuse  
Effacer App?  
0:Enter N:Esc
```








6.6 Sauvegarde/restitution de l'application en mémoire EEPROM

Utilisation principale : sauvegarde de l'application mise au point (avec possibilité de chargement automatique en mémoire RAM TSX).

- Pour un transfert RAM automate -> EEPROM :
Source : TSX
Destination : EEPROM
- Pour un transfert EEPROM -> RAM automate :
Source : EEPROM
Destination : TSX

```
►Transférer App  
Source:►TSX  
Dest:►EEPROM  
Prot:►N MST:►0
```



Après le choix de la destination : (voir ch 6.2)

- ④  donne accès au choix de la protection (Prot) en lecture de l'application par  ou  .
- ⑤  donne accès à l'option de chargement automatique (MST) de la mémoire RAM automate par le contenu de la mémoire EEPROM. Ce chargement s'effectue lorsqu'une différence est détectée à la mise sous tension, entre les 2 mémoires avec mise en RUN automatique de l'automate (excepté si l'entrée RUN/STOP est configurée et, est en position STOP).
Le choix s'effectue par  ou  .
- ⑥  lance le transfert.

Nota : lors d'un transfert TSX --> EEPROM, les sorties automate ne doivent pas être activées.

7.1 Généralités

L'éditeur Automate TSX permet la mise en route de l'application dans l'automate, le programme doit donc être en mémoire automate, et le terminal doit être en mode connecté (commutateur sur TSX).

L'accès au mode s'effectue par  .

Il propose les fonctions pour la mise en route :

Editeur AutomateTSX

- **0 Inf** Information sur l'état de l'automate
- **1 Run/Stop**
- **2 Ini** Initialisation de l'application
- **3 Rtc** programmation des horodateurs (voir ch5)
- **4 Clk** Mise à l'heure de l'horloge interne.

Etat automate :
RUN ou STOP

Référence de
l'automate
connecté

```
TSX: STOP
App=Exec
TSX 07 31-16
_1Run_2Ini_3Rtc_
```










Etat application :
Exécutable, non
exécutable ou in-
connu.

7.2 Mise à l'heure de l'horloge interne (4 Clk)

 Menu

 4

Cette fonction permet de mettre à l'heure l'horloge interne de l'automate. Cette opération doit être exécutée lors de la première mise en route de l'automate ou après une mise hors tension prolongée.

- sélectionner le jour de la semaine par  ou ,
- accéder au choix du jour par , et saisir le jour à l'aide du clavier numérique,
- accéder au choix du mois par , et sélectionner le mois par  ou ,
- accéder au choix de l'année par , et saisir celle-ci à l'aide du clavier numérique,
- accéder au choix de l'heure par , et saisir celle-ci à l'aide du clavier numérique,
- valider le réglage par .

```
_Ve 10 _Mar1994
17:26:48
STP= 2 Jan 1996
2 17:07:23
```

Nota : les 2 lignes inférieures de l'écran visualisent la date et l'heure de la dernière mise en "Stop" de l'automate. Un code indique la cause de la mise en Stop :

- 1= passage de RUN en STOP par le terminal,
- 2= arrêt sur défaut logiciel (débordement de tâche automate),
- 4= coupure secteur,
- 5= arrêt sur défaut matériel.

7.3 Initialisation (2 Ini)

Menu 2

Cette commande initialise l'application de l'automate, elle doit être lancée lors de la première mise en exécution (RUN) de l'application ou après chaque modification de l'application, elle provoque une reprise à froid (voir chapitre A7.1).

Menu 2 donne accès à l'écran de confirmation d'initialisation.

↻ lance l'initialisation.

```
TSX->Ini?  
0:Enter  N:Esc
```

7.4 Mise en exécution RUN ou arrêt STOP (1 Run)

Menu 1

Menu 1 donne accès à l'écran de confirmation de mise en RUN (ou STOP suivant l'état initial) de l'automate.

↻ lance la mise en RUN (ou STOP) de l'automate.

Vérifier, sur l'écran principal du mode que l'automate est bien dans l'état demandé.

D'autre part, il est possible de vérifier l'état de fonctionnement de l'automate, sur sa face avant :

- le voyant RUN clignote lorsque l'automate est en STOP
- le voyant RUN est allumé sans clignotement lorsque l'automate est en RUN.

```
TSX->RUN?  
0:Enter  N:Esc
```

```
TSX : RUN  
APP : Exec  
TSX 07 31-16  
1Run2Ini3Rter
```

Nota : si la mise en RUN ou STOP n'est pas prise en compte, vérifier qu'une entrée n'a pas été configurée pour piloter la commande RUN/STOP de l'automate, dans ce cas celle-ci est prioritaire par rapport au terminal.

7.5 Information (0 Inf)

Menu 0

Cet écran donne en plus des informations de l'écran de base de l'éditeur TSX, une information sur l'état de l'automate :

- OK : si aucun défaut n'est détecté
- ! : si un défaut est détecté

```
TSX : RUN  
APP : Exec  
TSX 07 31-16  
Mod0=OK
```

8.1 Généralités

Le terminal FTX 117 offre un ensemble de fonctions permettant la mise au point des applications en langage liste d'instructions :

- visualisation dynamique de l'état des entrées/sorties, des bits internes et des valeurs des variables suivant la base désirée (décimal, hexadécimal ou ASCII),
- modification des valeurs des variables et de l'état des bits,
- forçage des bits d'entrées/sorties,
- mise en place d'un point d'arrêt et exécution cycle par cycle,
- modification du programme en RUN.

L'éditeur de programme **Prg** donne accès à la fonction de mise au point du programme **Dgb** par Menu 4 ou Shift ⇨ .

Elle propose des fonctions d'aide à la mise au point accessibles par menu :

Fonction Mise au point Dgb

- **0 Inf** : Information (voir ch 7.5)
- **1 Trf** : Transfert de programme (voir ch 6)
- **2 Cnv** : Choix de la base d'affichage
- **3 Frc** : Forçage des entrées/sorties
- **4 Prg** : Retour à l'écran de programmation

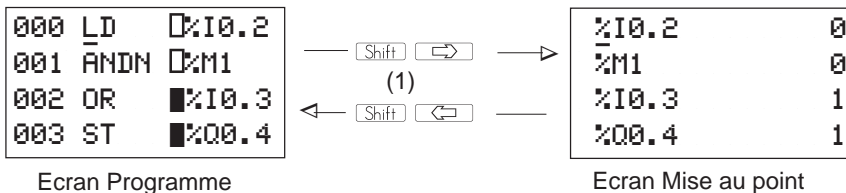
8.2 Visualisation dynamique du programme

Lorsque l'automate est en RUN, l'écran programme visualise directement l'état des variables booléennes par un rectangle placé devant chaque une d'elle :

- rectangle vide = état 0,
- rectangle plein = état 1.

000	LD	<input type="checkbox"/> %I0.2
001	ANDN	<input type="checkbox"/> M1
002	OR	<input checked="" type="checkbox"/> %I0.3
003	ST	<input checked="" type="checkbox"/> %Q0.4

L'écran de mise au point visualise directement l'état des variables booléennes et les valeurs numériques contenues dans chaque ligne de programme. Il se présente comme une extension à droite de l'écran de programme.



Dans le cas des variables numériques, la fonction conversion Menu 2 permet de choisir la base d'affichage (Hexadécimale, Decimale, Binaire ou ASCII). La visualisation Binaire n'est pas possible en visualisation dynamique de programme par contre elle est accessible avec l'éditeur de données

Hex	16#346
Dec	838
Bin	00000011
Ascii	(#F(

↶ et ↷ permettent de sélectionner la base et ↵ valide le choix.

%I0.3	1
%Q0.4	1
%MW0	16#346
%KW10	16#F00

Les variables numériques de l'écran sont alors affichées dans la base choisie.

(1) ou Menu 4

8.3 Modification et forçage

L'écran mise au point permet de modifier :

- l'état des bits,
- les valeurs des mots.

Nota: le programme reste prioritaire ainsi si une instruction commande l'état du bit ou écrit la variable, les modifications effectuées à l'aide du terminal ne sont pas prises en comptes.

%I0.3	1
%Q0.4	1
%MW0	100_
%KW10	200

Pour modifier une variable, à partir de l'écran de mise au point :

↵ donne accès à la saisie de la valeur,
↵ valide la saisie.

Forçage des Entrées/sorties

[Menu] [3] donne accès à l'écran de forçage des entrées/sorties (%I et %Q).

- choisir l'option forçage **O** (oui) par [→] ou [←], lorsque l'option O est choisie, l'écran propose le forçage de chacune des entrées/sorties.

```
Forçage      ►0
              01234567
%I0.0....►
```

- choisir les entrées ou les sorties à l'aide de [↑] et [↓], et pointer le curseur sur l'adresse de l'objet à forcer par [→] ou [←]. L'adresse complète de l'objet est affichée sur la gauche. Le forçage de l'objet s'effectue en saisissant la valeur 0 ou 1. La suppression d'un forçage s'effectue en appuyant sur [Del].

```
Forçage      ►0
              01234567
%I0.0  .1.0....►
%Q0.0  .1.0....►
```

- [↻] exécute les forçages. Les bits d'entrées/sorties forcés sont repérés par la lettre "f" devant la variable dans l'écran de mise au point.

```
f%I0.3      1
f%Q0.4      1
%MW0       100
%KW10      200
```

Nota :

- les forçages sont prioritaires sur l'écriture des entrées/sorties par le programme.
- le forçage de tous les bits est supprimé sur une reprise à froid.

8.4 Modification en RUN du programme

L'éditeur de programme autorise la modification du programme automate en RUN.


Avertissement

Pour des raisons de sécurité évidentes, il est conseillé d'effectuer la programmation automate à l'arrêt (STOP).

Cependant la programmation automate en exécution (RUN) est possible afin de pouvoir exécuter les modifications de programmes ne nécessitant pas l'arrêt de l'application mais ces dernières restent sous la responsabilité de l'utilisateur.

La programmation d'un automate en exécution est soumise à certaines conditions. Avant toute modification, il est indispensable d'en connaître les conséquences sur l'application et de prendre toutes les dispositions nécessaires à cet effet.

Accès à la modification en RUN

Lors de la première modification, un écran prévient l'utilisateur que l'automate est en RUN,  valide la modification.

```
TSX en RUN
O:Enter  N:ESC
```



Restrictions

Le tableau suivant donne les restrictions d'utilisation et de modification lorsque l'automate est en RUN :

Mode/fonction	Accès
Configuration	Accès uniquement en visualisation
Programmation	<p>Toutes les modifications/insertion/suppression d'instructions sont autorisées mises à part les instructions pouvant modifier la structure du programme :</p> <ul style="list-style-type: none">• parenthèses,• instructions Grafcet,• étiquettes %Li: et SRi:,• instructions LD %●● suivant une instruction %Li:, SRi:, -*i, *=i,• saut : JMP, et appel à sous programme SR,• bloc BLK, OUT_BLK, END_BLK• instructions MPS, MPP, MRD <p>Les fonctions suivantes ne sont pas actives :</p> <ul style="list-style-type: none">• effacement du programme,• remplacement

9.1 Généralités

L'éditeur de données assure le réglage des données du programme : valeurs des variables mots, des bits, des blocs fonction et le forçage des bits d'entrées/sortie.

L'accès à cet éditeur s'effectue par   à partir de l'écran de base. Le réglage est accessible dès l'entrée dans l'éditeur, les objets visualisés par défaut sont ceux pointés dans le programme (accès contextuel).

















L'éditeur de données propose des fonctions d'aide au réglage accessibles par menu :

Editeur de Données Dat

- **0 Inf** : Information (voir ch4.7)
- **1 Trf** : Transfert de tables de données
- **2 Cnv** : Choix de la base d'affichage
- **3 Frc** : Forçage (voir Fonction mise au point ch8)
- **4 Dat** : Retour à l'écran de réglage en cours
- **5 Dt1** : Création/visualisation de la table de données n°1
- **6 Dt2** : Création/visualisation de la table de données n°2
- **7 Dt3** : Création/visualisation de la table de données n°3
- **8 Dt4** : Création/visualisation de la table de données n°4
- **9 Clr** : Ré-initialisation des tables de données

9.2 Principe d'accès aux objets à régler

L'accès à l'objet à régler s'effectue de la façon suivante :

- ① choix du type d'objet par les touches du clavier (ex:                

9.3 Objets accessibles par l'éditeur de données

Les différentes notations utilisées sont : **R** pour lecture, **W** pour écriture, **F** pour forçage.

Type		Désignation	Touche	Possibilités
Objets bits	bits d'entrées	%I.j	<input type="text" value="%I"/>	R, W, F
	bits de sorties	%Qi.j	<input type="text" value="%Q"/>	R, W, F
	bits internes	%Mi	<input type="text" value="%M"/>	R, W
	bits système	%Si	<input type="text" value="%S"/>	R, W (1)
	bits d'étape Grafcet	%Xi	<input type="text" value="Var"/> <input type="text" value="9"/>	R
	bits extraits de mots			aucune
Objets mots	mots internes	%MWi	<input type="text" value="μ"/>	R, W
	mots constants	%KWi	<input type="text" value="%KW"/>	R
	mots système	%SWi	<input type="text" value="%SW"/>	R, W (1)
	mots d'entrées	%IWi.j	<input type="text" value="Var"/> <input type="text" value="1"/>	R, W
	mots de sorties	%QWi.j	<input type="text" value="Var"/> <input type="text" value="2"/>	R, W
Objets blocs fonction standards	Temporisateur	%TMi	<input type="text" value="%TM"/>	
	• présélection	%TMi.P		R, W (2)
	• valeur courante	%TMi.V		R, W
	• sortie	%TMi.Q		R
	Compteur	%Ci	<input type="text" value="%C"/>	
	• présélection	%Ci.P		R, W (2)
	• valeur courante	%Ci.V		R, W
	• sortie compteur vide	%Ci.E		R
	• sortie présélection	%Ci.D		R
	• sortie compteur plein	%Ci.F		R
	Registre	%Ri	<input type="text" value="Var"/> <input type="text" value="1"/> <input type="text" value="0"/>	
	• mot d'entrée	%Ri.I		R, W
	• mot de sortie	%Ri.O		R, W
	• sortie registre vide	%Ri.E		R
	• sortie registre plein	%Ri.F		R
	Programmateurs	%DRi	<input type="text" value="Var"/> <input type="text" value="1"/> <input type="text" value="1"/>	aucune (3)
Objets blocs fonction spécifiques	Modulation de largeur	%PWM	<input type="text" value="Var"/> <input type="text" value="7"/>	
	• présélection	%PWM.P		R
	• pourcentage	%PWM.R		R, W
	Générateur d'impulsions	%PLS	<input type="text" value="Var"/> <input type="text" value="6"/>	
	• présélection	%PLS.P		R, W (2)
	• nb impulsions	%PLS.N		R, W
	• sortie en cours	%PLS.Q		R
	• sortie terminée	%PLS.D		R
	Compteur rapide	%FC	<input type="text" value="Var"/> <input type="text" value="5"/>	
	• présélection	%FC.P		R, W
	• valeur courante	%FC.V		R, W
	• valeurs de seuil 0 et 1	%FC.S0 et 1		R, W
	• sortie dépassement	%FC.F		R
	• sortie dépassement seuil	%FC.TH0 et 1		R
	Registre à décalage	%SBRi	<input type="text" value="Var"/> <input type="text" value="3"/>	aucune (3)
	Pas à pas	%SCi	<input type="text" value="Var"/> <input type="text" value="4"/>	aucune (3)
	Message	%MSG	<input type="text" value="Var"/> <input type="text" value="8"/>	aucune (3)

(1) Ecriture possible de certains bits et mots système,

(2) Ecriture possible si l'option réglage est sélectionnée en configuration.

(3) Accès possible en création et visualisation de tables de données (fonctions Dt1 à Dt4).

9.4 Visualisation et modification des variables booléennes

Entrées/sorties :

L'écran visualise dans la partie supérieure l'état des entrées et dans la partie inférieure l'état des sorties :

- un rectangle blanc = état 0
- un rectangle noir = état 1

Les lignes du milieu donnent des informations sur chaque entrée/sortie :



- ! = défaut sur une voie
- f= forçage



Exemple : sortie %Q0.2 en défaut, entrée %I0.3 et sortie %Q0.6 forcées.

		%I0.0%I0.8	
%I0.5	0		
Err=0	...	f.....	
	f..	
%Q0.0	0		
		%Q0.0%Q0.8	





Les adresses de l'entrée et de la sortie visualisées sur la gauche correspondent à celles pointées par le curseur.

Les entrées non affichées sont accessibles par .



 et  permettent de choisir entre entrées et sorties.

 et  permettent de choisir l'adresse des entrées ou des sorties.

L'information Err indique lorsqu'elle est à 1 que l'automate ou l'extension d'entrées/sorties est en défaut.

 et  permettent d'écrire les entrées/sorties. Le forçage des entrées/sorties est accessible par   (voir ch8.3).

Autres variables booléennes



Dans le cas des autres variables booléennes (exemple : bits internes, bits système...), l'écran visualise une liste de 8 variables. Il est possible de visualiser les autres variables de la liste par  et .


 et  permettent d'écrire les variables.



%M0	0	%M1	0
%M2	0	%M3	0
%M4	0	%M5	0
%M6	0	%M7	0

9.5 Visualisation et modification des variables numériques

Dans le cas des variables numériques (exemple : mots internes, mots système...)

L'écran visualise une liste de 4 variables. Il est possible de visualiser les autres variables de la liste par  et .

Pour modifier la valeur d'une variable, il faut la sélectionner, saisir la nouvelle valeur et valider par .



  donne accès au choix de la base de visualisation (Décimal, hexadécimal, binaire, ASCII) (voir ch8.2). En visualisation binaire, les valeurs sont représentées sur 2 lignes : octet poids fort sur la 1° ligne, octet poids faible sur 2° ligne.


%MW10	-	100
%MW11		100
%MW12		200
%MW13		200

%MW10	00000000	01100100
%MW11	00000000	01100100

9.6 Visualisation et modification des blocs fonction

L'écran visualise les paramètres et les sorties du bloc fonction choisi.

L'écran visualise une liste de 4 variables. Il est possible de visualiser les autres variables de la liste par  et .

Pour modifier la valeur d'une variable, il faut la sélectionner, saisir la nouvelle valeur et valider par . Les variables suivies de ":" sont modifiables, celles suivies de "=" ne le sont pas.

Rappel: les valeurs de présélection ne sont modifiables que si l'option Réglage est choisie en configuration du bloc.

```
%C5
%C5.P:-      100
%C5.V:       0
%C5.E=       0
```

```
%C5.V:-      100
%C5.E=       0
%C5.D=       1
%C5.F=       0
```



9.7 Visualisation des étapes Grafcet

L'écran visualise les étapes actives. Elles sont repérées par "X". Les 62 étapes sont représentées sous la forme d'un tableau :

- la première ligne de l'écran repère les unités.
- la première colonne de l'écran repère les dizaines.

Ainsi l'écran ci-contre visualise les étapes 0 à 29 et parmi ces étapes, les étapes 1, 2, et 16 sont actives.

Le nombre situé en haut et à droite de l'écran (%X43 dans cet écran) repère l'étape pointée par le curseur.

 et  permettent d'accéder aux autres lignes du tableau.

```
0123456789 %X24
0 XX
1      X
2      _
```

```
0123456789 %X43
3
4  XX
5  _
```

9.8 Création et visualisation de tables de données (5 Dt1 à 8 Dt4)

Les fonctions tables de données Dt1 à Dt4 permettent de visualiser les valeurs de variables d'une application. Elles permettent de créer jusqu'à 4 listes de 16 variables de type différent avec modification possible des valeurs associées à ces variables. Ces variables sont animées lorsque le terminal FTX117 est en mode connecté.

Ces listes peuvent être sauvegardées en mémoire Flash interne au terminal ou sur carte mémoire (voir ch 9.10).

Ces fonctions sont accessibles en mode connecté (TSX) et en mode autonome (FTX). En mode autonome, il est aussi possible de donner des valeurs aux variables. En mode connecté, ces valeurs peuvent être affectées aux variables du programme automate.

[Menu] [5] ([6] [7] ou [8]) donne accès à la saisie de la liste 1 (2, 3, ou 4).

La saisie des variables s'effectue par les touches clavier (voir annexes intercalaire G) et est validée par [↵].

Pour modifier la valeur d'une variable, il faut la sélectionner, pointer avec le curseur le champ variable et saisir la nouvelle valeur et la valider par [↵].

L'écran visualise 4 variables maximum de la liste. Il est possible d'accéder aux autres variables de la liste par [↑] et [↓], ou par [Shift] [↑] et [Shift] [↓] pour atteindre le début et la fin de la liste.

[Menu] [2] donne accès au choix de la base de visualisation (Décimal, hexadécimal, binaire, ASCII) (voir ch8.2) pour la variable numérique sélectionnée.

En visualisation binaire, les valeurs sont représentées sur 2 lignes : octet poids fort sur la 1^{re} ligne, octet poids faible sur 2^e ligne.

[Menu] [4] (fonction **Dat**) redonne accès à l'écran de réglage en cours.

Liste 1	
%I0.2	1
%C1.F	1
%M13	0

Liste 1	
%I0.2	1
%C1.F	1
%M13	1_



champs variable

%SW118	00000000
	01100100
%MW10	16#64
%MW20	(AB(

9.9 Ré-initialisation des tables de données (9 Clr)












[Menu] [9] (fonction **Clr**) ré-initialise les 4 tables de données après confirmation par [↵].

9.10 Transfert de tables de données (1 Trf)

  donne accès à l'achivage et à la restitution des données de l'application.

Les données archivées correspondent aux tables de données créées par les fonctions **Dt1** à **Dt4** précédentes.












Archivage :

- ① Choisir la source : FTX ou TSX par  ou ,
FTX : effectue le transfert des tables de données avec les valeurs définies par l'utilisateur.
TSX : effectue le transfert des tables de données et avec les valeurs correspondantes de l'application.
- ② Choisir la destination Carte mémoire (ou FLASH) par  ou 
- ③ Appuyer sur  et choisir par , ,  ou  le numéro d'archive (1).
- ④  valide le choix.
- ⑤  lance le transfert (2).

```
Transférer liste
Source:►FTX
Dest:►CARTE
ARCH:
```

```
ARCH0 p ARCH4 d
ARCH1 p ARCH5
ARCH2 d ARCH6
ARCH3 d ARCH7
```

Restitution :

- ① Choisir la source : Carte mémoire (ou FLASH) par  ou ,
- ② Choisir la destination FTX ou TSX par  ou 
FTX : effectue uniquement le transfert des tables de données (sans les valeurs associées).
TSX : effectue le transfert des tables de données et remplace les données correspondantes de l'application en cours par celles sauvegardées.
- ③ Appuyer sur  et choisir par , ,  ou  le numéro d'archive (1).
- ④  valide le choix.
- ⑤  lance le transfert(2).

```
Transférer liste
Source:►CARTE
Dest:►TSX
ARCH:
```

- (1) le "d" à coté de chaque nom d'archive indique que des données ont déjà été sauvegardées (rappel : le "p" indique qu'un programme a été sauvegardé). La sauvegarde des données sur une archive déjà utilisée détruit cette dernière.
- (2) un message "Erreur Carte" suivi d'un code, peut apparaître dans les cas décrits chapitre 6.4.

Chapitre	Page
1 Recherche et analyse des défauts	1/1
1.1 Recherche des défauts à partir des voyants d'état automate	
1.1-1 Sur automate de base ou extension automate	1/1
1.1-2 Sur extension d'entrées/sorties	1/2
1.1-3 Sur automate de base, extension automate ou extension d'E/S	1/3
1.2 Analyse des défauts à partir des bits et mots système	1/4
1.2-1 Bits système	1/4
1.2-2 Mots système	1/5












1.1 Recherche des défauts à partir des voyants d'état automate

L'utilisateur est renseigné sur le mode marche et les éventuels défauts de fonctionnement de l'automate par des voyants d'état situés en face avant.

Note :

A chaque mise sous tension de l'automate, tous les voyants sont allumés pendant une durée d'environ 1 seconde correspondant à la phase d'autotests. Les sorties ne sont pas pour autant activées.

1.1-1 Sur automate de base ou extension automate

Etats des voyants		Signification	Cause probable
Voyant RUN		Automate hors tension ou application non exécutable	Automate non alimenté ou application invalide
		Automate en STOP	Etat demandé par le terminal (commande par entrée configurée RUN/STOP %I0.1 ou par le terminal) ou état entraîné par un défaut d'exécution (débordement chien de garde logiciel, appel à une fonction non implémentée ou interdite)
		Automate en RUN	Etat normal
Voyant ERR		Fonctionnement OK	Etat normal
		Application non exécutable	Application absente, invalide, défaut checksum, débordement chien de garde logiciel.
		Défauts internes	<ul style="list-style-type: none"> • débordement chien de garde hardware • auto-tests mauvais (Pb accès RAM, EEPROM ou horodateur) • inversion câblage de liaison extension
Voyant I/O		Fonctionnement OK	Etat normal
		Défaut d'entrées/sorties	Défauts d'entrées/sorties <ul style="list-style-type: none"> • sorties statiques disjonctées • défaut alimentation capteurs • défaut configuration
Voyant COM		Pas d'échange sur la liaison extension	
		Echanges en cours sur la liaison extension	
		Echanges en cours sur la liaison Modbus	



voyant éteint












voyant clignotant





voyant allumé fixe

1.1-2 Sur extension d'entrées/sorties





Etat des voyants		Signification	Causes probable
Voyant RUN		Extension hors tension ou non connectée sur liaison extension.	Extension non alimentée, non connectée ou erreur de connexion sur câble liaison extension.
		Automate de base en STOP (image du voyant RUN/STOP de l'automate de base).	Identiques à celles de l'automate de base.
		Automate de base en RUN (image du voyant de l'automate de base)	Etat normal.
Voyant ERR		Fonctionnement OK.	Etat normal.
		Défauts internes.	<ul style="list-style-type: none"> • débordement chien de garde hardware, • auto-tests mauvais. • inversion de câblage de la liaison extension
Voyant I/O		Fonctionnement OK	Etat normal,
		Défaut d'entrées/sorties	Défauts d'entrées/sorties : <ul style="list-style-type: none"> • sorties statiques disjonctées • défaut alimentation capteurs.
Voyant COM		Pas d'échange sur la liaison extension.	
		Echanges en cours sur liaison extension.	


 voyant éteint


 voyant clignotant

 voyant allumé fixe

1.1-3 Sur automate de base, extension automate ou extension d'E/S

Etat des voyants		Signification	Causes probable
Voyants I0.....I13		Entrée inactive	Etat normal si capteur non passant
		Entrée active	Etat normal si capteur passant
Voyants O0...O9		Sortie inactive	Etat normal si sortie non activée.
		Sortie active	Etat normal si sortie activée

 voyant éteint

 voyant allumé fixe

Si les voyants I5 (TSX Nano 10 E/S), I8 (TSX Nano 16 E/S) ou I13 (TSX Nano 24 E/S) sont clignotants (suite de 5 clignotements brefs toutes les secondes) (1) :
 Les voyants I0 à I7 et O0 à O7 indiquent l'état 0 (éteint) ou 1 (allumé) des bits internes %M112 à %M127.

Voyants	Signification	Type	Voyants	Signification	Type
I0	%M112	TSX Nano	O0	%M120	TSX Nano
I1	%M113	10/16/24 E/S	O1	%M121	10/16/24 E/S
I2	%M114		O2	%M122	
I3	%M115		O3	%M123	
I4	%M116	TSX Nano	O4	%M124	TSX Nano
I5	%M117	24 E/S	O5	%M125	24 E/S
I6	%M118		O6	%M126	
I7	%M119		O7	%M127	

(1) Dans ce cas le bit système %S69 est à l'état 1.

1.2 Analyse des défauts à partir des bits et mots système

Sur détection d'un défaut, le système automate positionne un bit ou mot système correspondant à l'erreur détectée. Cette information pourra être utilisée ou non par le programme application.

Le terminal FTX 117 ou le logiciel PL7-07 (sur FTX 417/507 ou compatible PC) en mode réglage permet la visualisation des bits système (touche %S) et des mots système (touche %SW). Voir mode réglage intercalaire C - chapitre 9 dans manuel TSX Nano/ FTX 117 ou intercalaire C - chapitre 14 dans manuel TSX Nano PL7-07.

1.2-1 Bits système

Bits système	Fonction	Désignation
%S10	Défaut d'E/S	Normalement à l'état 1. Est mis à l'état 0 quand un défaut d'entrées/sorties de l'automate de base ou de l'extension d'entrées/sorties (configuration non conforme, défaut d'échange, défaut matériel, disjonction des sorties statiques protégées) est détecté. Les bits %S118 et %S119 indiquent l'automate en défaut et les mots %SW118 et %SW119 précisent la nature du défaut. Le bit %S10 est remis à 1 dès la disparition du défaut.
%S11	Débordement du chien de garde	Normalement à l'état 0, est mis à l'état 1 par le système dès que le temps d'exécution du programme devient supérieur au temps de cycle maximum (chien de garde de 150 ms). Provoque le passage en STOP de l'automate.
%S19	Débordement de période de scrutation (scrutation périodique)	Normalement à l'état 0, ce bit est mis à l'état 1 par le système en cas de dépassement de la période d'exécution (temps d'exécution du programme supérieur à la période définie par l'utilisateur en configuration ou programmé dans SW0). Ce bit est remis à l'état 0 par l'utilisateur.
%S71	Echange sur liaison extension	Normalement à l'état 0. Est mis à l'état 1 dès qu'une extension d'entrées/sorties ou une extension automate échange avec l'automate de base sur la liaison extension. Le bit %S71 est remis à 0 quand aucun échange s'effectue sur la liaison extension. Le mot %SW71 de l'automate de base donne la liste et l'état des extensions présentes.
%S118	Défaut d'E/S sur automate de base	Normalement à l'état 0. Est mis à l'état 1 quand un défaut d'entrées/sorties est détecté sur l'automate de base. Le mot %SW118 permet de déterminer la nature du défaut. Le bit %S118 est remis à 0 dès la disparition du défaut.
%S119	Défaut d'E/S sur extension automate	Normalement à l'état 0. Est mis à l'état 1 quand un défaut d'entrées/sorties est détecté sur l'automate d'extension d'entrées/sorties. Le mot %SW119 permet de déterminer la nature du défaut. Le bit %S119 est remis à 0 dès la disparition du défaut.

1.2-2 Mots système

Mots système	Fonction	Désignation
%SW71	Défaut sur liaison extension	Indique l'état de la communication de chaque extension présente avec l'automate de base : bit 1 : 1= extension d'entrées/sorties bit 2 : 1= extension automate n°2 bit 3 : 1= extension automate n°3 bit 4 : 1= extension automate n°4 bit à l'état 0: extension absente, non alimentée, non câblée ou en défaut. bit à l'état 1: extension présente et échangeant avec l'automate de base
%SW118	Etat automate de base	Indique les défauts détectés sur l'automate de base. bit 0 : 0= disjonction des sorties statiques (1) bit 3 : 0= défaut alimentation capteur bit 8 : 0= défaut interne ou défaut matériel TSX Nano bit 9 : 0= défaut externe ou défaut dialogue bit 11 : 0= automate en auto-tests bit 13 : 0= défaut de configuration
%SW119	Etat extension automate	Indique les défauts détectés sur l'extension automate bit 0 : 0= disjonction des sorties statiques (1) bit 3 : 0= défaut alimentation bit 8 : 0= défaut interne ou défaut matériel bit 9 : 0= défaut externe ou défaut dialogue bit 11 : 0= automate en auto-tests bit 14 : 0= absence de l'extension alors que cette dernière était présente à l'initialisation

(1) suite à la surcharge ou court-circuit sur l'une des sorties.

Chapitre	Page
1 Cahier des charges : portique de lavage automatique de véhicules	1/1
1.1 Description de l'application	1/1
1.2 Fonctionnement de l'application	1/2
1.2-1 Cycle automatique de lavage	1/2
1.2-2 Arrêt manuel de cycle sur incident	1/2
1.3 Représentation graphique du cycle de lavage	1/3
2 Solution câblée	2/1
2.1 Schéma de puissance de l'installation	2/1
2.2 Schéma de commande	2/1
3 Solution avec automate TSX Nano 16 entrées/sorties	3/1
3.1 Schéma de puissance de l'installation	3/1
3.2 Schéma de raccordement de l'automate	3/2
3.3 Nomenclature	3/3
3.3-1 Entrées automate	3/3
3.3-2 Sorties automate	3/3
3.3-3 Variables internes automate	3/3
3.4 Schéma à contacts équivalent	3/4
3.5 Programme langage à contacts	3/5
3.6 Configurations du bloc fonction	3/7
3.7 Programmation du bloc horodateur RTC	3/7

Important

L'exemple présenté dans ce chapitre est décrit à titre didactique. Son utilisation dans une application industrielle nécessite des adaptations selon les règles de sécurité en vigueur dans le secteur d'activité concerné.

1 Cahier des charges: portique de lavage automatique de véhicules

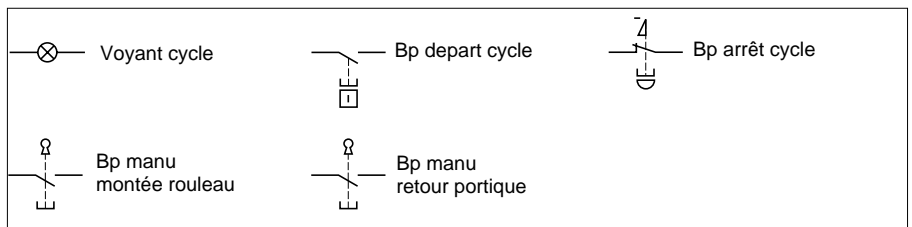
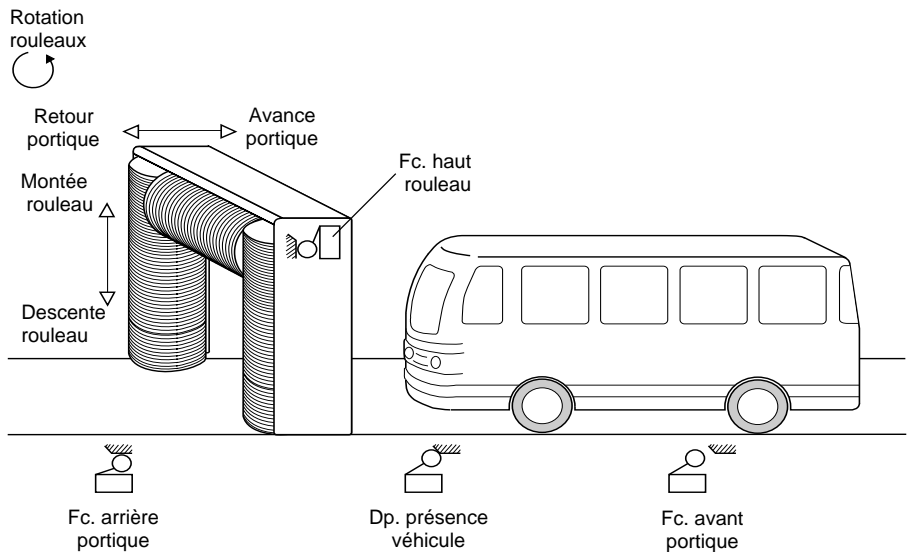
1.1 Description de l'application

Un portique de lavage est composé de :

- Un portique supportant des rouleaux horizontal et verticaux, entraîné par un moteur deux sens de marche (avance et retour),
- Un moteur de rotation des rouleaux horizontal et verticaux,
- Un moteur pour la montée et descente du rouleau horizontal.

Des interrupteurs de position contrôlent les positions :

- Haute du rouleau horizontal,
- Avant et arrière du portique.



1.2 Fonctionnement de l'application

1.2-1 Cycle automatique de lavage

Conditions initiales : Le portique est en position arrière (fc. arrière) et le rouleau horizontal est en position haute (fc. haut rouleau). Un véhicule est présent dans l'aire de lavage (dp. présence véhicule).

Les conditions initiales étant réunies, une action sur le bouton poussoir Bp. départ cycle lance le cycle suivant :

- Voyant cycle allumé et attente de 10 secondes (KA0),
- Descente du rouleau horizontal (KM1) pendant 5 secondes (KA1),
- Mise en rotation des rouleaux (KM3) et avance du portique (KM4) . On suppose dans cet exemple que les pompes de projection d'eau sont activées en même temps que le moteur de rotation des rouleaux.
- Arrêt de l'avance portique par le détecteur fc. avant portique et commande du retour portique (KM5),
- Arrêt par le détecteur fc. arrière du retour portique et de la rotation des rouleaux. Commande de la remontée du rouleau horizontal (KM2) jusqu'au détecteur fc haut qui provoque la fin de cycle.

Un horodateur gère les jours et les heures d'ouverture (du lundi au samedi de 8 heures à 19 heures 30 minutes. En dehors de ces plages horaires, aucune demande de départ cycle n'est prise en compte.

Un compteur hebdomadaire comptabilise le nombre de lavages effectués. Il est remis à zéro automatiquement tous les lundis à 8 heures. Un autre compteur cumule le nombre de lavages effectués au fil des semaines.

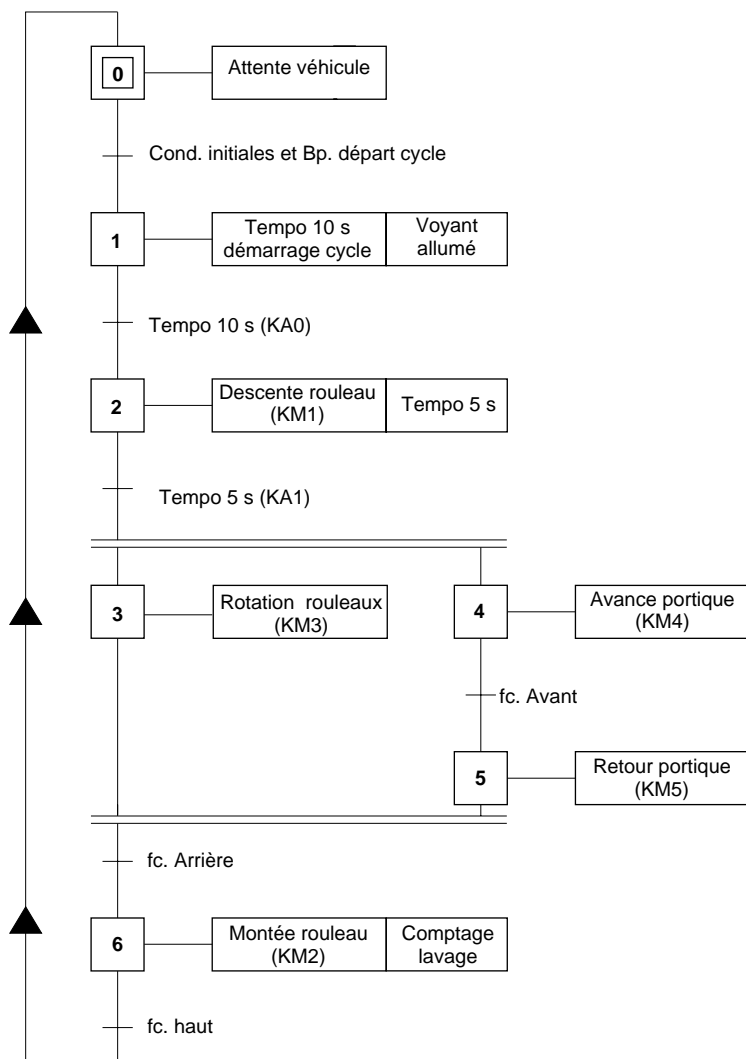
1.2-2 Arrêt manuel de cycle sur incident

Un bouton poussoir à accrochage Bp. arrêt cycle provoque à tout moment du cycle l'arrêt de celui-ci (arrêt immédiat de tous les moteurs). Pour la relance d'un nouveau cycle, il est nécessaire de provoquer :

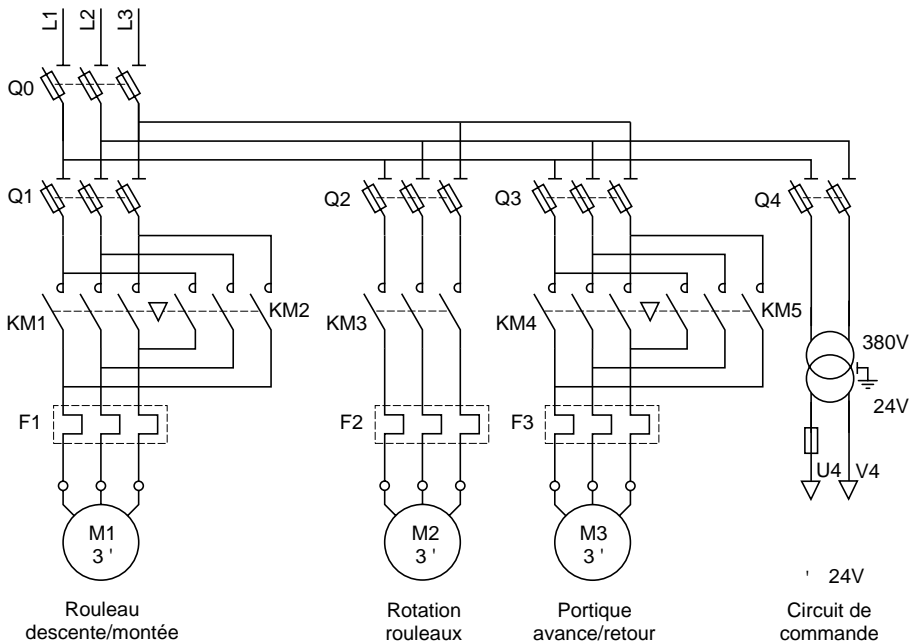
- La remontée du rouleau horizontal (jusqu'au détecteur fc.haut) par appui maintenu du bouton poussoir Bp. manu montée rouleau,
 - Le retour du portique en position arrière (jusqu'au détecteur fc. arrière) par appui maintenu du bouton poussoir Bp. manu retour portique,
 - Le déverrouillage du bouton poussoir arrêt cycle.
-

1.3 Représentation graphique du cycle de lavage

Le Grafcet ci-dessous représente graphiquement le fonctionnement automatique du portique de lavage.



2.1 Schéma de puissance de l'installation

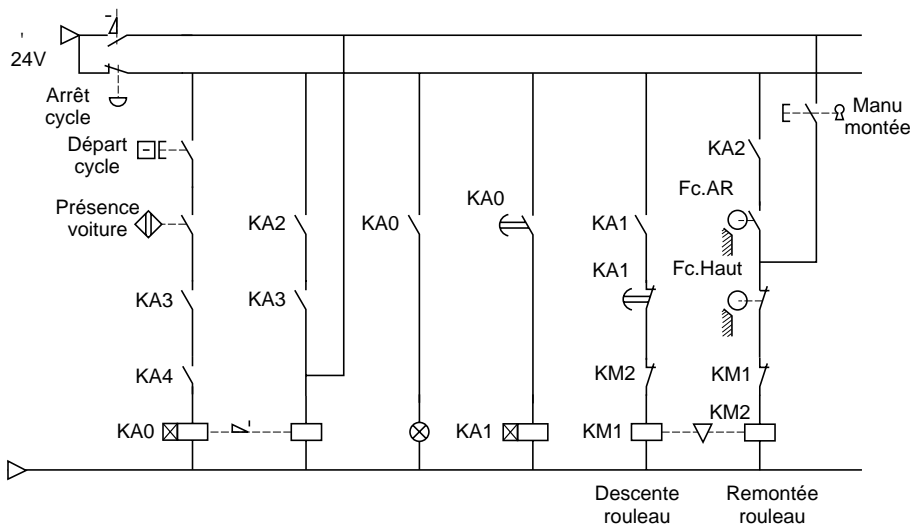


2.2 Schéma de commande

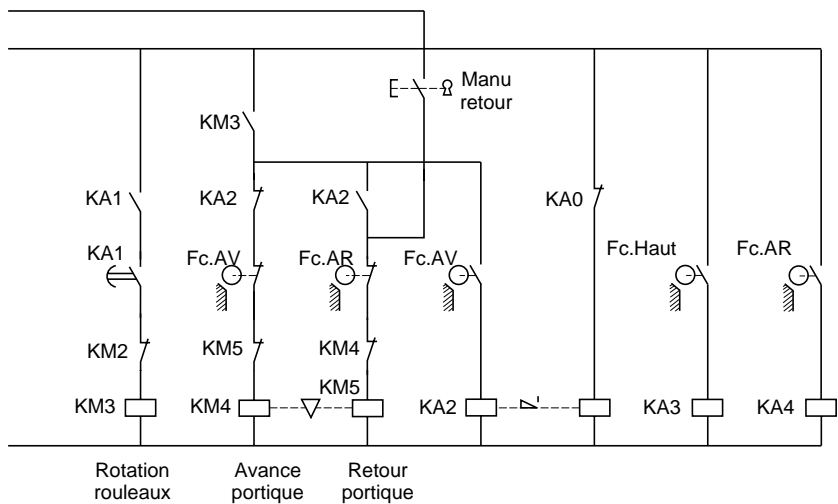
Le schéma ci-dessous correspond aux fonctionnements en cycle automatique ainsi que sur arrêt manuel du cycle. Les fonctions horodateur et compteur totalisateur n'apparaissent pas dans ce schéma.

Deux contacteurs auxiliaires à mémoire à accrochage mécanique (KA0 et KA2) assurent la mémorisation de l'avancement du cycle. De ce fait après coupure secteur, le cycle se poursuit à partir de l'état où il avait été interrompu. Les deux temporisations sont réalisées par deux additifs temporisés sur KA0 et KA1.

Les deux interrupteurs de position arrière et haut, du fait du nombre de contacts utilisés dans le schéma nécessitent deux contacteurs auxiliaires supplémentaires (KA3 et KA4).

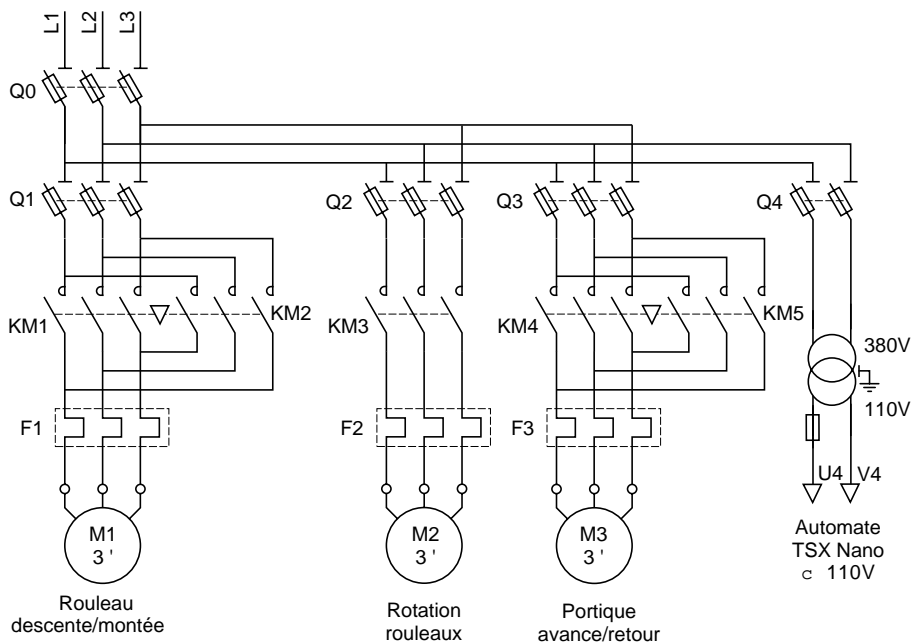


E



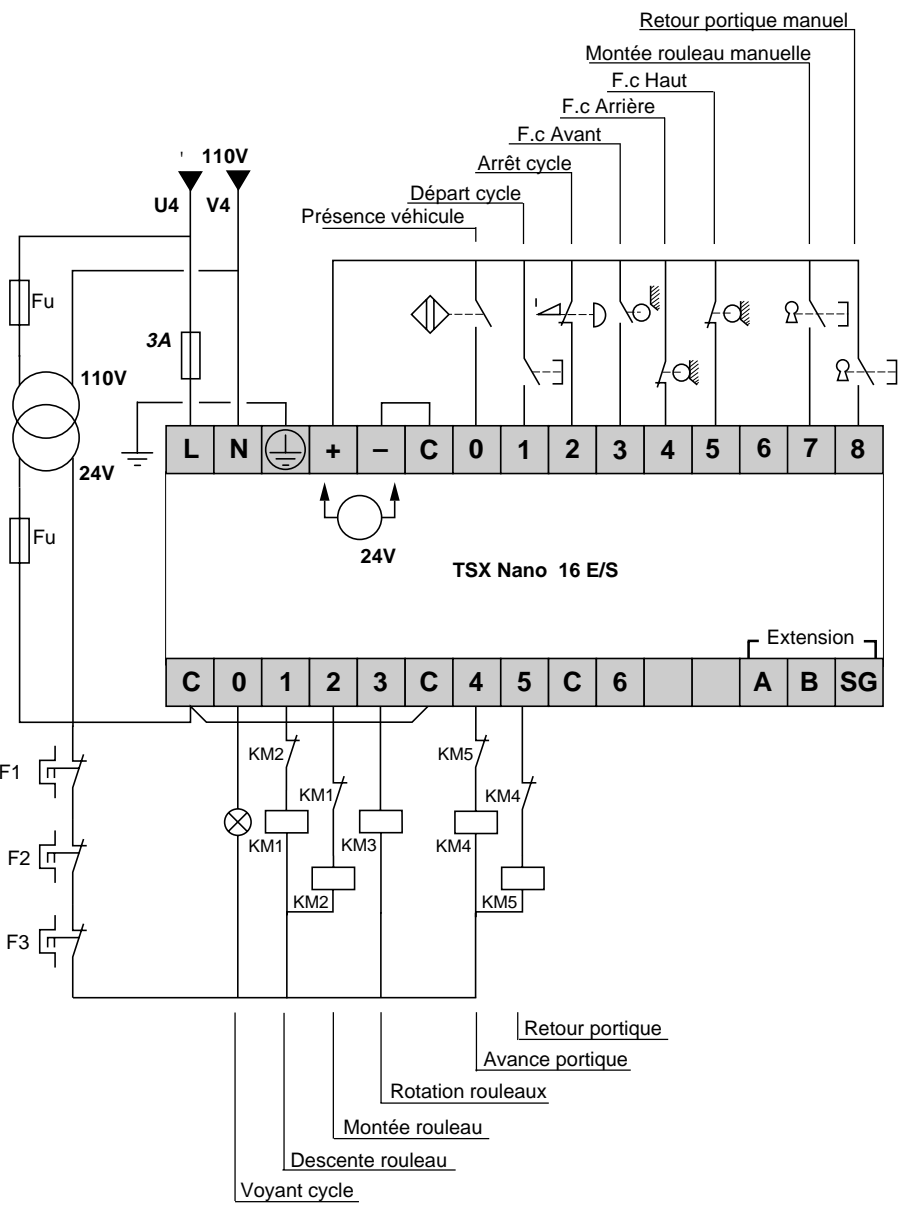
3.1 Schéma de puissance de l'installation

Ce schéma est identique à celui de la solution logique cablée. L'alimentation 110VCA de l'automate TSX Nano s'effectue par un transformateur 380/110VCA. La tension d'alimentation des préactionneurs est fixée à 24VCA fournie par un second transformateur.



3.2 Schéma de raccordement de l'automate

E



3.3 Nomenclature

3.3-1 Entrées automate

Repère	Adresse	Désignation
Présence véhicule	%I0.0	Dp. présence véhicule
Départ cycle	%I0.1	Bp. départ cycle
Arrêt	%I0.2	Bp. arrêt cycle
Fc. Avant	%I0.3	Fc. avant portique
Fc. Arrière	%I0.4	Fc. arrière portique
Fc. Haut	%I0.5	Fc. haut rouleau
Manu remontée	%I0.7	Bp.manu.montée rouleau
Manu.retour	%I0.8	Bp.manu.retour portique

3.3-2 Sorties automate

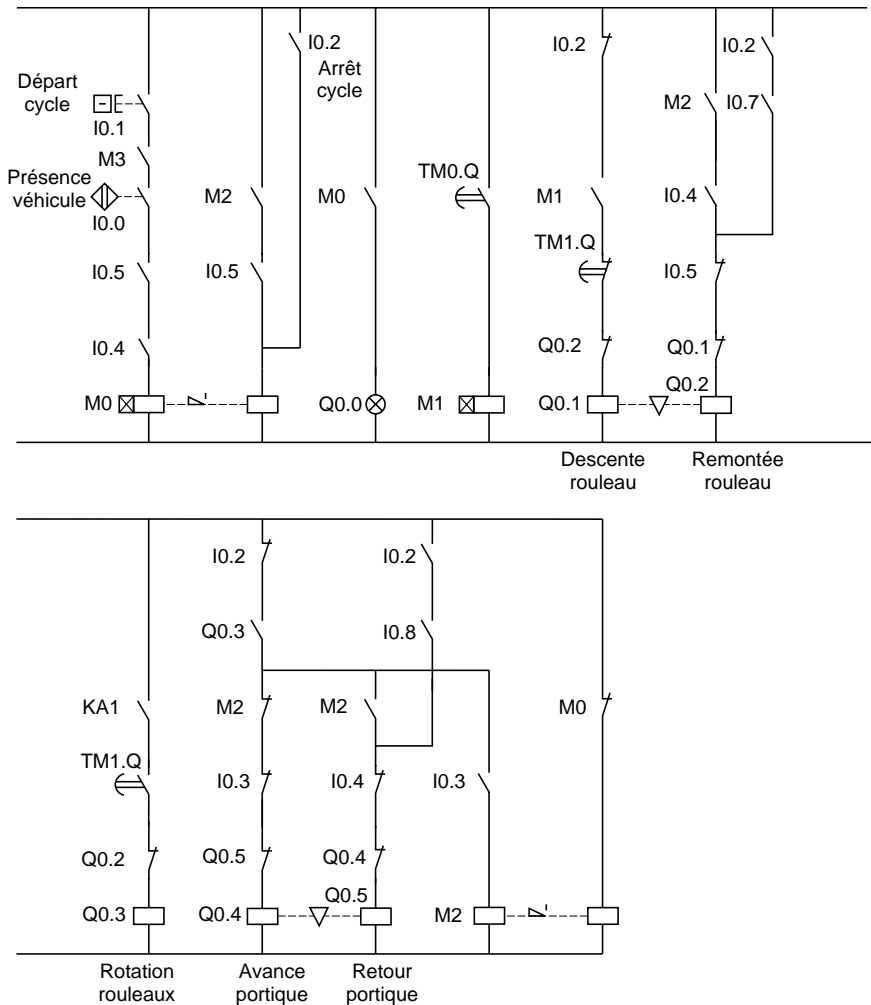
Repère	Adresse	Désignation
Voyant	%Q0.0	Voyant cycle
KM1	%Q0.1	Contacteur descente rouleau
KM2	%Q0.2	Contacteur montée rouleau
KM3	%Q0.3	Contacteur rotation rouleaux
KM4	%Q0.4	Contacteur avance portique
KM5	%Q0.5	Contacteur retour portique

3.3-3 Variables internes automate

Type	Adresse	Désignation
Bit interne	%M0	Variable mémorisation départ cycle (KA0)
Bit interne	%M1	Variable descente rouleau (KA1)
Bit interne	%M2	Variable mémo. avance portique (KA1)
Bit interne	%M3	Variable sortie horodateur
Bit interne	%M4	Variable test du lundi
Bit interne	%M5	Variable pour création impulsion sur %M4
Bit interne	%M6	Variable pour création impulsion sur %M2
Bit interne	%M7	Variable test sortie %TM1
Mot interne	%MW0	Totalisateur nombre de lavage
Mot système	%SW50	Secondes/jour courant horodateur
Fonction temporisateur	%TM0	Temporisateur départ cycle
Fonction temporisateur	%TM1	Temporisateur descente rouleau
Fonction compteur	%C0	Compteur hebdomadaire de lavage
Fonction horodateur	RTC0	Horodateur

3.4 Schéma à contacts équivalent

Directement déduit du schéma de commande de la solution câblée présenté au chapitre 2.2, le schéma adapté à la solution automate est le suivant.



Les contacteurs auxiliaires KA0, KA1 et KA2 sont remplacés par les bits internes %M0, %M1 et %M2. Les contacteurs auxiliaires KA3 et KA4 sont sans objets, le nombre de tests d'une même variable automate n'étant pas limité dans un programme.

Le front montant du bit interne %M4, mis à l'état 1 tous les lundis (par le test des quatre premiers bits du mot système %SW50) provoque la mise à zéro du compteur hebdomadaire de lavage (compteur %C0).

Le totalisateur du nombre de lavage (mot interne %MW0) est reinitialisé à la valeur 1 automatiquement lorsque sa valeur atteint 30 001.

3.5 Programme List

Le programme réversible en langage à contacts est le suivant :

000	LD	%I0.1	<i>Bp. départ cycle</i>
	AND	%I0.0	<i>Dp. présence véhicule</i>
	AND	%M3	Bit autorisation horodateur
	AND	%I0.5	<i>Fc. haut rouleau</i>
	AND	%I0.4	<i>Fc. arrière portique</i>
005	S	%M0	Mémo départ cycle
	LD	%M2	
	AND	%I0.5	
	OR	%I0.2	<i>Bp. arrêt cycle</i>
	R	%M0	
010	LD	%M0	
	ST	%Q0.0	Voyant cycle
	BLK	%TM0	
	LD	%Q0.0	
	IN		Tempo départ cycle (10s)
015	END_BLK		
	LD	%TM0.Q	
	ST	%M1	
	BLK	%TM1	
	LD	%M1	
020	IN		Tempo descente rouleau (5s)
	END_BLK		
	LD	%TM1.Q	
	ST	%M7	
	LDN	%I0.2	
025	AND	%M1	
	ANDN	%M7	
	ANDN	%Q0.2	
	ST	%Q0.1	Descente rouleau (KM1)
	LD	%M2	
030	AND	%I0.4	
	OR(%I0.2	
	AND	%I0.7	<i>Bp.manu.montée rouleau</i>
)		
	ANDN	%I0.5	
035	ANDN	%Q0.1	
	ST	%Q0.2	Remontée rouleau (KM2)
	LD	%M1	
	AND	%M7	
	ANDN	%Q0.2	
040	ST	%Q0.3	Rotation rouleaux (KM3)
	LDN	%I0.2	

	AND	%Q0.3	
	MPS		
	ANDN	%M2	
045	ANDN	%I0.3	<i>Fc. avant portique</i>
	ANDN	%Q0.5	
	ST	%Q0.4	Avance portique (KM4)
	MPP		
	AND	%I0.3	
050	S	%M2	Mémo. avance portique
	LDN	%I0.2	
	AND	%Q0.3	
	AND	%M2	
	OR(%I0.2	
055	AND	%I0.8	<i>Bp. manu. retour portique</i>
)		
	ANDN	%I0.4	
	ANDN	%Q0.4	
	ST	%Q0.5	Retour portique (KM5)
060	LDN	%M0	
	R	%M2	
	LDN	%SW50:X0	{ test du jour de la
	ANDN	%SW50:X1	{ semaine: Lundi
	ANDN	%SW50:X2	{ 1° octet= 0
065	ANDN	%SW50:X3	{
	ST	%M4	
	BLK	%C0	Compteur hebdomadaire de lavage
	LD	%M4	(0 à 9999)
	ANDN	%M5	
070	R		Remise à 0 tous les lundis
	LD	%M2	
	CU		Comptage
	END_BLK		
	LD	%M4	
075	ST	%M5	
	LD	%M2	
	ANDN	%M6	
	[INC %MW0]		Totalisateur de lavage (0 à 30 000)
	LD	%M2	
080	ST	%M6	
	LD	[%MW0=30001]	{Remise du totalisateur à la valeur 1
	[%MW0:=1]		{quand passage à la valeur 30 001
	END		Fin de programme

3.6 Configuration des blocs fonction

L'accès à la configuration des blocs fonctions s'effectue par la sélection :
e 4 m 2

- Temporisateur départ cycle %TM0

%TM0	
BT:	100 ms
%TM0 . P:	100
Type:	TON

- Temporisateur descente rouleau %TM1

%TM1	
BT:	100 ms
%TM1 . P:	50
Type:	TON

- Compteur hebdomadaire de lavage %C0

%C0	
%C0 . P:	9999
Regl age:	N

3.7 Programmation de l'horodateur

L'accès à la programmation de l'horodateur RTC0 s'effectue par la sélection :
e 1 3

Cette programmation correspond à une ouverture de la station de lavage :

- Du 2 janvier au 31 décembre,
- Les lundi, mardi, mercredi, jeudi, vendredi et samedi,
- De 8 heures à 19 heures 30 minutes.

RTC: 0	Q %M3
2	Jan - μ31 Dec
LMMJVS.	
08: 00- μ19: 30	

Chapitre	Page
Produits connectables sur prise terminal	1/1
1.1 Caractéristiques de la prise terminal	1/1
1.2 Généralités sur la prise terminal	1/2
1.3 TSX Nano en mode ASCII	1/3
1.4 TSX Nano Maître sur UNI-TELWAY	1/4
1.5 TSX Nano Esclave sur UNI-TELWAY	1/7
1.6 Le Time-out UNI-TELWAY (TSX 07 3●)	1/8
1.7 Terminaux d'exploitation XBT ou pupitres d'exploitation CCX17	1/9
1.8 Requêtes UNI-TE supportées par le TSX Nano (serveur)	1/10
Produits connectables sur port d'extension	2/1
2.1 Caractéristiques du port d'extension MODBUS / JBUS	2/1
2.2 MODBUS / JBUS sur TSX Nano	2/2
2.2-1 Généralités	2/2
2.2-2 Configuration de la liaison MODBUS	2/4
2.2-3 Requêtes supportées par le TSX Nano en MODBUS	2/5
2.2-4 Gestion du voyant COM	2/6
2.2-5 Bits et mots systèmes associés	2/6
2.3 Requêtes standard MODBUS	2/7
2.3-1 Lecture de n bits internes %Mi	2/7
2.3-2 Lecture de n mots internes %MWi	2/8
2.3-3 Ecriture d'un bit interne %Mi	2/9
2.3-4 Ecriture d'un mot interne %MWi	2/10

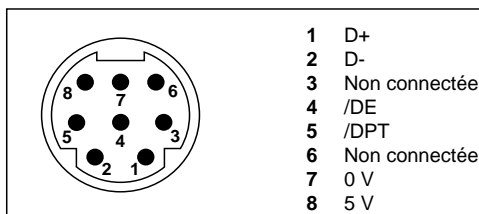
Chapitre	Page
2.3-5	Ecriture de n bits internes %Mi
2.3-6	Ecriture de n mots internes %MWi
2.3-7	Calcul du LRC
2.3-8	Algorithme de calcul du CRC 16
2.4	Requêtes d'accès au serveur UNITE du TSX Nano
2.4-1	Identification
2.4-2	Read-CPU
2.4-3	Requête RUN
2.4-4	Requête STOP
2.4-5	Requête INIT
2.5	Limitations

1.1 Caractéristiques de la prise terminal

• Caractéristiques

Type de liaison	: RS485
Protocole	: UNI- TE.
Débit binaire	: 9600 bits/s
Type de connecteur	: Mini DIN, 8 points à verrouillage rapide
Distance maximum de la liaison	
FTX 117	: 10 m
UNI-TELWAY	: 10 m
ASCII	: 10 m

• Brochage du connecteur



Le signal /DPT permet la sélection du mode de fonctionnement de la prise terminal :

/DPT = 1	Mode UNI-TELWAY Maître
/DPT = 0	Mode UNI-TELWAY Esclave ou mode ASCII.

Pour être en mode ASCII, les broches 5 et 7 doivent être reliées.

Note :

Il est conseillé de raccorder les équipements hors tension (excepté les terminaux de programmation).

L'utilisation d'équipements ASCII et UNI-TELWAY est exclusive. L'utilisation d'un terminal de programmation (FTX 117, ...) nécessite la déconnexion de l'équipement ASCII.

1.2 Généralités sur la prise terminal

La prise terminal du TSX Nano peut fonctionner suivant trois modes différents:

- UNI-TELWAY Maître (TSX 07 2., TSX 07 3.),
- UNI-TELWAY Esclave (TSX 07 3. uniquement),
- ASCII (TSX 07 2., TSX 07 3.).

La sélection du mode de fonctionnement de la prise terminal s'effectue par configuration logicielle et l'exploitation du signal /DPT (broche n°5) de la fiche Mini DIN :

- Lorsque le signal /DPT est à 1 (broche n°5 non raccordée), la prise terminal est en mode UNI-TELWAY Maître.
- Lorsque le signal /DPT est à 0 (broche n°5 reliée à la broche n°7=0V), la prise terminal est en mode ASCII ou UNI-TELWAY Esclave. Le choix est réalisé par configuration logicielle à l'aide des outils de programmation PL707 et FTX 117 (mode ASCII par défaut).

L'état du /DPT est renseigné par le bit système %S100.

Ecran de configuration:

PRISE CONSOLE

Type
☒ ASCII ☐ UNI-TELWAY esclave

Bits/sec
☐ 1200 ☐ 2400 ☐ 4800 ☒ 9600

Adresse esclave UNI-TELWAY: 4

Parité
☒ Impaire
☐ Paire
☐ Aucun

Bits Données
☒ 8 Bits ☐ 7 Bits

Bits Stop
☒ 1 Bit ☐ 2 Bits

Time out UNI-TELWAY
Modes Maître et Esclave (car.): 30

OK Annuler

1.3 TSX Nano en mode ASCII

Ce mode caractères, simplifié sur le TSX Nano permet l'émission (TSX 07 2. et TSX 07 3.) et/ou la réception (TSX 07 3. uniquement) d'une chaîne de caractères de/ou vers un équipement simple (imprimante ou terminal) sans contrôle de flux.

Ce mode est destiné à un fonctionnement en liaison de type point à point.

La configuration de la prise terminal est modifiable sur le TSX 07 3., elle s'effectue dans l'écran de configuration de l'automate:

- Type: Half-Duplex
- Vitesse: 1200, 2400, 4800, **9600 bds**
- Format: 1 bit de start, 7 ou **8 bits** de données, **1** ou **2 bits de stop**
- Parité: paire, **impaire**, sans.

Les valeurs en **gras** sont les valeurs par défaut, elles ne sont pas modifiables sur les TSX 07 2..

Ecran de configuration ASCII:

La programmation en mode ASCII est réalisée par l'intermédiaire de l'instruction EXCH ainsi que par le bloc fonction de contrôle des échanges %MSG. Voir chapitre 3.3-6, intercalaire B.

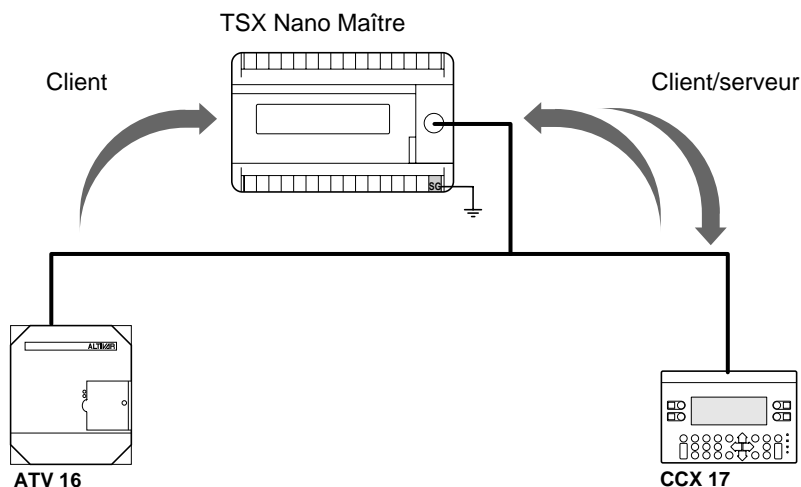
1.4 TSX Nano Maître sur UNI-TELWAY

Le TSX Nano est compatible avec d'autres équipements UNI-TELWAY tels que les équipements d'interfaces homme-machine (MMI), d'identification inductive, les variateurs de vitesse,

Ce sous-chapitre ne présente que les principes généraux de connexion de ces équipements sur le bus UNI-TELWAY. Pour plus de détails, se reporter à leur manuel de mise en œuvre.

Lorsque le TSX Nano est maître de la liaison UNI-TELWAY. Il contrôle le réseau et interroge les esclaves à intervalles réguliers. Ils sont connectés à la prise terminal du TSX Nano.

Exemple d'architecture



Bien que le TSX Nano soit normalement serveur UNI-TE, il peut cependant être client UNI-TE mais avec des fonctionnalités limitées pour les TSX 07 2• :

• Configuration

La plupart des équipements UNI-TELWAY nécessitent l'utilisation de deux adresses. La première adresse est configurée physiquement par l'utilisateur (à l'aide de micro-contacts, ...), la seconde correspond généralement à l'adresse physique + 1. Le TSX Nano peut communiquer avec un terminal de programmation et avec au maximum deux autres équipements UNI-TELWAY. Affectation des adresses :

- 0 : TSX Nano (maître de la liaison)
- 1 : terminal de programmation (FTX117 ou logiciel PL7-07)
- 2-3-5 : équipements clients seulement (TSX 07 2•)
- 4 : équipement client et/ou serveur.(TSX 07 2•)
- 2-3-4-5: équipements clients et/ou serveurs (TSX 07 3•)

Récapitulatif des équipements connectables et leurs adresses possibles

Type	Référence	Adresses possibles		
		1	2,3	4,5
Terminal de programmation	FTX117 PL7-07	Oui	Non	Non
MMI	XBT-8 CCX17	Non	Oui	Oui
Équipement d'identification	XGS	Non	Oui	Oui
Variateur de vitesse	ATV	Non	Oui (1)	Oui

(1) Uniquement TSX Nano V3

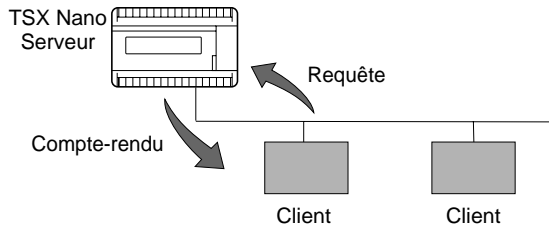
La configuration de la prise terminal en mode UNI-TELWAY Maître n'est pas modifiable. Elle gère 5 adresses liaisons esclaves selon le format de transmission suivant:

- Type: Half-Duplex
- Vitesse: 9600 bds
- Format: 1 bit de start, 8 bits de données, 1 bit de stop
- Parité: impaire.

Le Time-out est paramétrable dans l'écran de configuration.

Fonction serveur UNI-TE :

Le TSX Nano répond à une commande émise par le client. Un client est un équipement intelligent qui prend l'initiative de la communication avec le TSX Nano. Il peut écrire ou lire des données du TSX Nano.



Adressage

- Le TSX Nano est maître de la liaison UNI-TELWAY (son adresse est toujours 0),
- il scrute les adresses liaisons de 1 à 5,
 - le terminal de programmation (FTX117 ou logiciel PL7-07) doit toujours avoir l'adresse 1,
 - d'une manière générale, l'adresse d'un équipement UNI-TELWAY est définie soit par des micro-contacts dans les boîtiers de dérivation soit par les câbles de raccordement.

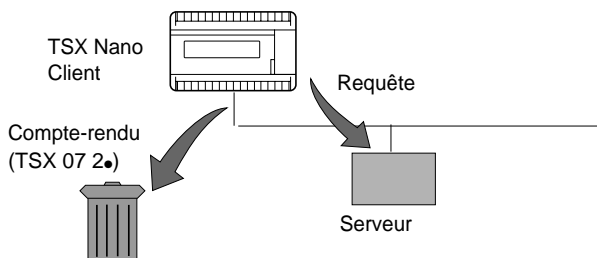
Notes:

- La communication entre esclaves n'est pas possible, lorsque le maître de bus est un TSX Nano,
- Le TSX 07 2• ne peut pas être esclave UNI-TELWAY.

Tous les équipements du bus peuvent interroger le serveur système du TSX Nano en utilisant l'adresse destination 0.254.0. Le serveur UNITE du TSX Nano ne répond qu'aux trames destinées à cette adresse. Les messages contenant une autre adresse destination seront ignorés.

• Fonction client UNI-TE :

Le TSX Nano prend l'initiative de la communication avec un serveur. Le serveur est un équipement intelligent qui exécute les commandes envoyées par le TSX Nano.



Le TSX 07 3• Maître peut envoyer une requête vers n'importe quel esclave d'adresse 1 à 5 à l'aide de l'instruction EXCH. Il utilise l'adresse source 0.254.16 .

Le TSX 07 2• ne peut envoyer une requête qu'à l'esclave d'adresse liaison 4 (utilisation de l'instruction EXCH), par conséquent, seules les requêtes Ecriture et Données Non Sollicitées peuvent être utilisées. Il utilise l'adresse source 0.254.10.

ATTENTION:

La fonction client UNI-TE du TSX 07 3• inverse les données émises (poids fort / poids faible) par rapport au TSX 07 2•. Les applications fonctionnant avec des TSX 07 2• devront être modifiées pour tenir compte de cette inversion si elles sont chargées sur un TSX 07 3•.

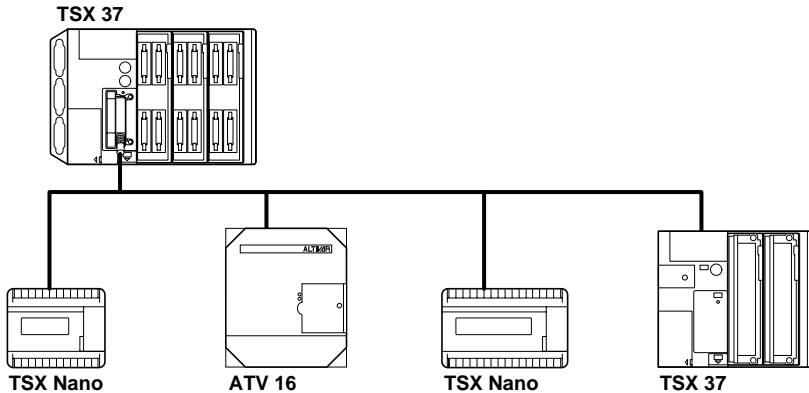
La programmation en mode UNI-TELWAY Maître est réalisée par l'intermédiaire de l'instruction EXCH ainsi que par le bloc fonction de contrôle des échanges %MSG. Voir chapitre 3.3-6, intercalaire B.

1.5 TSX Nano Esclave sur UNI-TELWAY

Ce protocole n'est disponible qu'avec des TSX Nano de version supérieure ou égale à 3 (TSX 07 3●).

Il permet le raccordement simultané, en multi-point de plusieurs équipements (automate, console de programmation, équipement de dialogue opérateur, variateur de vitesse, etc...).

Exemple d'architecture



Contrairement au mode Maître, la configuration de la prise terminal en mode UNI-TELWAY esclave est modifiable par l'utilisateur à l'aide des outils de programmation PL7 07 ou FTX 417 dans l'écran de configuration de l'automate :

- Type: Half-duplex
- Vitesse: 1200, 2400, 4800 ou **9600 bds**
- Format: **1 bit** de start, 7 ou **8 bits** de données, 1 ou 2 bits de stop
- Time-out: **30** à 255 (voir chapitre 1.6)
- Adresse: **4** (1 à 97)
- Parité: paire, **impaire** ou sans.

Les valeurs en **gras** sont les valeurs par défaut.

Le TSX Nano utilise 2 adresses logiques consécutives :

- AD0: adresse de base (celle de la configuration), appelée adresse serveur. N'importe quel équipement (local ou distant) peut s'adresser au serveur système du TSX Nano en utilisant AD0 comme adresse destination.
- AD1 = AD0 + 1 appelée adresse client. C'est avec cette adresse que le TSX Nano peut émettre une requête vers n'importe quel équipement du bus UNI-TELWAY (Maître ou Esclave) à l'aide de l'instruction EXCH.

Voir chapitre 3.3-6 intercalaire B.

La modification des caractéristiques de la prise terminal effectuée par l'écran de configuration n'est prise en compte que sur reprise secteur ou modification du signal /DPT de la fiche Mini DIN (déconnexion / reconnexion de la prise terminal).

1.6 Le Time-out UNI-TELWAY (TSX 07 3.)

La couche liaison du protocole UNI-TELWAY (Maître ou Esclave) utilise un time-out. Ce Time-out correspond à la durée d'émission d'un nombre de caractères émis sur la ligne:

- TSX 07 3.: 30 caractères par défaut. Cette durée est portée à 125 caractères si le TSX Nano ne possède pas d'application.

A la fin de l'émission d'une trame, un équipement (Maître ou Esclave) déclare un Time-out.

Si aucun acquittement n'arrive avant l'expiration de ce délai, l'échange est non-acquitté. L'émetteur réitérera sa trame dès que le protocole le lui permettra.

L'écran de configuration de la prise terminal permet de configurer un Time-out entre 30 et 8000 caractères.

Les valeurs comprises entre 30 et 250 correspondent à un nombre équivalent de caractères. Les valeurs comprises entre 251 et 255 correspondent aux valeurs suivantes:

- 251 = 500 caractères
- 252 = 1000 caractères
- 253 = 2000 caractères
- 254 = 4000 caractères
- 255 = 8000 caractères.

A la mise sous tension ou sur modification du signal /DPT, la valeur configurée est chargée dans le poids faible du mot système %SW14 (le poids fort est ignoré).

Ainsi, la valeur du Time-out peut être affinée en écrivant la nouvelle valeur dans le mot système %SW14. Elle sera prise en compte à la fin du cycle automate.

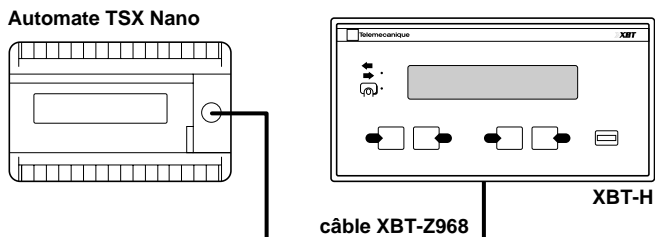
L'augmentation du Time-out permet de connecter le TSX Nano à des équipements lents de type modem par exemple.

Il est possible de descendre la valeur du Time-out jusqu'à 10 caractères par l'utilisation du mot système %SW14.

1.7 Terminals d'exploitation XBT ou pupitres d'exploitation CCX17

• Raccordement automatique

Le terminal d'exploitation XBT ou pupitre d'exploitation CCX 17 se connecte sur la prise terminal de l'automate TSX Nano par l'intermédiaire d'un câble XBT-Z968.



• Caractéristiques de la liaison

- liaison RS 485
- Protocole UNI-TE

• Terminals XBT connectables sur la prise terminal de l'automate TSX Nano

- XBT-A8 •
- XBT-B8 •
- XBT-BB8 •
- XBT-C8 •
- XBT-K8 •
- XBT-M8 •
- XBT-A8 •
- XBT - H
- XBT - P
- XBT - E

• Pupitres CCX 17 connectables sur la prise terminal de l'automate TSX Nano

- TCCX 17 20 F
- TCCX 17 20 FW
- TCCX 17 20 FPS
- TCCX 17 20 L
- TCCX 17 20 LW
- TCCX 17 20 LPS
- TCCX 17 30 L
- TCCX 17 30 LW
- TCCX 17 30 LPS

• Mise en oeuvre terminal d'exploitation XBT :

- terminal d'exploitation XBT

(voir documentation technique relative à l'XBT utilisé)

- Echanges TSX Nano / XBT

Le langage PL7 permet d'afficher les messages sur le terminal XBT à partir:

- de l'instruction EXCH: permettant l'envoi de message,
- du bloc fonction %MSG: permettant le contrôle des échanges.

(voir chapitre 3.3-6 - intercalaire B: envoi de messages et contrôle des échanges).

• Remarques

Syntaxe des données au niveau XBT et au niveau TSX Nano (1)

Données	Syntaxe XBT (1)	Syntaxe TSX Nano
Bit interne	Bi	%Mi
Mot interne	Wi	%MWi

(1) sauf XBT - H/P/E.

1.8 Requête

UNI-TE supportée

par le TSX Nano (serveur)

Requêtes standards

Service	Requête	Question		Réponse		Signification
		Hexa	Déci	Hexa	Déci	
Données (lecture)	Lecture d'un bit	00	00	30	48	Lecture d'un bit %M.
	Lecture d'un mot	04	04	34	52	Lecture d'un mot %MW.
	Lecture d'objets	36	54	66	102	Lecture d'objets (%Mi, %Mi:L, %Mwi, %Mwi:L).
Données (écriture)	Ecriture d'un bit	10	16	FE	254	Ecriture d'un bit %M
	Ecriture d'un mot	14	20	FE	254	Ecriture d'un mot %MW.
	Ecriture d'objets	37	55	FE	254	Ecriture d'objets (%Mi, %Mi:L, %Mwi, %Mwi:L).
Modes de marche	RUN	24	36	FE	254	Mise en RUN d'un équipement.
	STOP	25	37	FE	254	Mise en STOP d'un équipement.

Requêtes spécifiques

Service	Requête	Question		Réponse		Signification
		Hexa	Déci	Hexa	Déci	
Données (lecture)	Lecture d'un bit système	01	01	31	49	Lecture d'un bit %S
	Lecture d'un mot constant	05	05	35	53	Lecture d'un mot %KW
	Lecture d'un mot système.	06	06	36	54	Lecture d'un mot système %SW
	Lecture d'étapes Grafcet	2A	42	5A	90	Lecture d'étapes Grafcet %X
Données (écriture)	Ecriture d'un bit système	11	17	FE	254	Ecriture d'un bit %S
	Ecriture d'un mot système	15	21	FE	254	Ecriture d'un mot %SW

2.1 Caractéristiques du port d'extension MODBUS / JBUS

Le port d'extension du TSX 07 3₊ peut être utilisé pour connecter d'autres automates TSX Nano en extension d'entrées / sorties (voir intercalaire A, chapitre 1.4) ou bien en extension automates (intercalaire A, chapitre 1.10).

A partir du TSX 07 3₊, ce port d'extension permet une liaison de type MODBUS Esclave.

Caractéristiques de la liaison MODBUS:

- Couche physique: RS485 non isolée, longueur limitée à 200 mètres
- Couche liaison: Transmission asynchrone
Trame ASCII (7 bits) ou **RTU (8 bits)**
Débit binaire: 1200, 2400, 4800, **9600** ou 19200 bds
Parité: **Paire**, impaire ou sans
Nombre de bits de stop: **1** ou 2
Temps inter-caractère: **3** (1 à 127) caractères
- Configuration physique: 28 équipements max.
98 adresses logiques (**1** à 98)
- Services: Bits: 128 bits par requête
Mots: 120 mots de 16 bits par requête
Sécurité: un paramètre de contrôle (CRC16) sur chaque trame (en RTU).

Les valeurs en **gras** sont les valeurs par défaut..

Nota: Il est conseillé de raccorder les équipements hors tension.

2.2 MODBUS / JBUS sur TSX Nano

2.2-1 Généralités

Les services offerts d'échanges de données (%Mi et %MWi) sont communs à MODBUS et à JBUS, cela permet de faire communiquer des équipements MODBUS avec des équipements JBUS.

Le protocole MODBUS / JBUS permet l'échange de données entre un Maître et des Esclaves , il ne permet pas la communication directe entre équipements Esclaves, ni la transparence réseau.

Communication MODBUS en mode ASCII:

- Le fonctionnement en mode ASCII est prévu pour connecter des équipements simples (écrans ...etc..) . La trame est complète mais les trames sont deux fois plus longues qu'en mode RTU.

Détail d'une trame ASCII:

Entête(3A) 1 octet	N°esclave 2octets	Code fonction 2 octet s	Données 2n octets	LRC 2 octets	CR 1 octet	LF* 1 octet
-----------------------	----------------------	----------------------------	----------------------	-----------------	---------------	----------------

*Le mot système %SW67 permet de paramétrer le caractère de fin de trame (LF). Il est écrit à 16#000A par le système sur initialisation à froid.

L'utilisateur peut modifier par programmation ou par réglage ce mot système quand le Maître Modbus utilise un caractère de fin de trame différent de 16#000A.

Communication MODBUS en mode RTU:

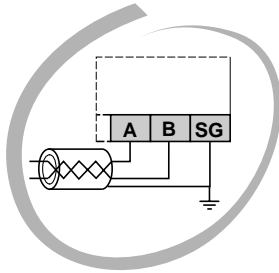
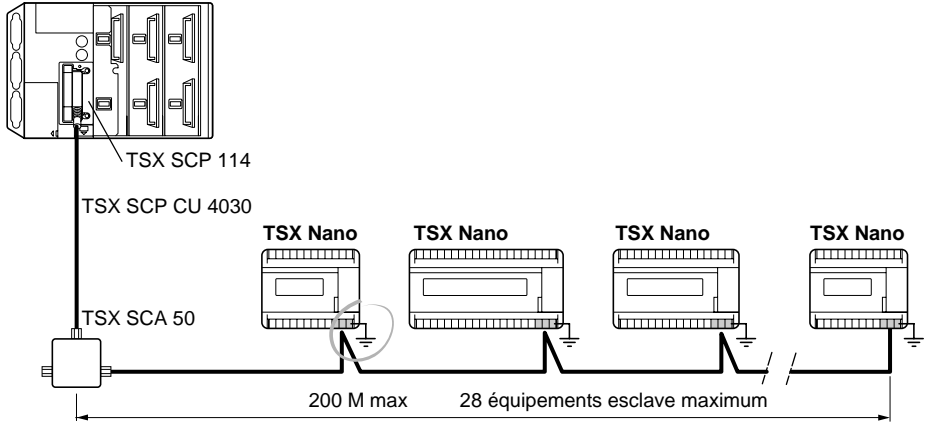
- Une trame en mode RTU ne comporte pas d'octet d'entête ni d'octet de fin de message:

N°esclave 1 octet	Code fonction 1 octet	Données n octets	CRC16 2 octets
----------------------	--------------------------	---------------------	-------------------

Exemple de raccordement

Par chainage

TSX Micro



2.2-2 Configuration de la liaison MODBUS

La configuration de la liaison extension en MODBUS / JBUS s'effectue depuis le menu **Configuration** du PL7 07 en choisissant **Port d'extension**.

La boîte de dialogue suivante apparaît:

PORT D'EXTENSION

Type
☐ Extension AP ☒ Modbus Esclave

Bits/sec
☐ 1200 ☐ 2400 ☐ 4800 ☒ 9600 ☐ 19200

Extension
☐ Oui ☒ Non

Adresse Esclave :

Time Out (Car.) :

Extension E/S
☒ Oui ☐ Non

Bits Données
☒ 8 (RTU) ☐ 7 (ASCII)

Parité
☒ Paire
☐ Impaire
☐ Aucun

AP2
☒ Oui ☐ Non

AP3
☒ Oui ☐ Non

AP4
☒ Oui ☐ Non

Bits Stop
☒ 1 Bit ☐ 2 Bits

OK Annuler

En fonctionnement dynamique, les paramètres (vitesse et format) ne sont pas modifiables.

Il n'y a pas de mécanisme d'adaptation de la vitesse du TSX Nano Esclave par rapport à la vitesse du Maître.

Toute modification de la vitesse et du format du protocole est prise en compte immédiatement par le TSX Nano dès la validation de l'écran de configuration.

Les champs **Extension**, **IO Extension**, **AP2**, **AP3** et **AP4** ne sont pas significatifs si le protocole MODBUS est sélectionné.

Voir intercalaire C chapitre 5.19 pour plus de détails.

2.2-3 Requêtes supportées par le TSX Nano en MODBUS

Le TSX Nano supporte les requêtes listée ci-dessous, les autres seront rejetées avec le code d'exception **Fonction inconnue: 01**.

La fonction MODBUS ne traite qu'une seule requête à la fois, car dans le protocole MODBUS le Maître doit attendre la réponse de l'esclave ou le déclenchement d'un timeout avant de lui adresser une nouvelle requête.

Les requêtes standards MODBUS :

- 01 / 02 : Lecture de n bits internes consécutifs: Accès aux bits %M0 à %M127
- 05 : Ecriture d'un bit interne: Accès aux bits %M0 à %M127
- 15 : Ecriture de n bits internes consécutifs: Accès aux bits %M0 à %M127
- 03 / 04 : Lecture de n mots consécutifs: Accès aux mots %MW0 à %MW255
- 06 : Ecriture d'un mot interne: Accès aux mots %MW0 à %MW255
- 16 : Ecriture de n mots consécutifs: Accès aux mots %MW0 à %MW255.

Ces requêtes sont détaillées au chapitre 2.3.

Les requêtes d'accès au serveur UNITE du TSX Nano:

- 0F : Identification
- 4F : Read CPU
- 24 : RUN
- 25 : STOP
- 33 : INIT

Ces requêtes sont détaillées au chapitre 2.4.

Les codes d'exception:

Un code d'exception est retourné par l'esclave lorsqu'il ne sait pas traiter la requête. La trame réponse comprend alors:

- Le code fonction reçu, incrémenté de la valeur 16#0080
- Le code d'exception qui dépend de la nature de l'erreur.

Les deux codes d'exception traités par le TSX Nano sont:

- 01: Fonction inconnue (requête non supportée par le TSX Nano)
- 03: Données invalides (numéro de bit ou mot non géré par le TSX Nano, écriture d'un bit avec une valeur différente de 16#FF00 ou 16#0000 etc...).

2.2-4 Gestion du voyant COM

Sur fin d'envoi de la réponse à une requête, le TSX Nano allume le voyant de communication pendant 50ms.

On obtient ainsi un clignotement du voyant dont la fréquence dépend des échanges entre le Maître et le TSX Nano .

2.2-5 Bits et mots systèmes associés

Sur traitement d'une requête MODBUS, le TSX Nano positionne le bit système %S70 à 1.

Ce bit peut être exploité par l'utilisateur. La remise à 0 est à la charge de l'utilisateur.

Le mot système %SW67 permet de paramétrer le caractère de fin de trame (LF) en mode ASCII.

Il est écrit à 16#000A par le système sur initialisation à froid.

L'utilisateur peut le modifier par programmation ou par réglage quand le Maître utilise un caractère de fin de trame différent de 16#000A .

2.3 Requêtes standard MODBUS

Ces requêtes permettent d'échanger des objets MODBUS par l'accès aux objets %Mi et %MWi du TSX Nano.

Les requêtes sont détaillées ci-dessous en mode RTU. En mode ASCII, les données sont les mêmes , Le CRC 16 étant remplacé par le LRC.

2.3-1 Lecture de n bits internes %Mi

Fonction 01 ou 02

Question :

Numéro Esclave	1 ou 2	N° du 1er bit		Nbre de bit		CRC 16
		PF	Pf	PF	Pf	
1 octet	1 octet	2 octets		2 octets		2 octets

Réponse :

Numéro Esclave	1 ou 2	Nombre d'octets lus	Valeur	Valeur	CRC 16
1 octet	1 octet	2 octets				2 octets

Exemple : lecture du bit %M3 de l'Esclave 2

Question	02	01	0003	0001	CRC 16
----------	----	----	------	------	--------

Réponse	02	01	01	xx	CRC 16
---------	----	----	----	----	--------

— 00 si %M3 = 0

— 01 si %M3 = 1

F

2.3-2 Lecture de n mots internes %MWi

Fonction 03 ou 04

Question :

Numéro Esclave	3 ou 4	N° du 1er mot		Nbre de mot		CRC 16
		PF	Pf	PF	Pf	
1 octet	1 octet	2 octets		2 octets		2 octets

Réponse :

Numéro Esclave	3 ou 4	Nombre d'octets lus	Valeur du 1er mot		Valeur dernier mot		CRC 16
			PF	Pf		PF	Pf	
1 octet	1 octet	1 octet	2 octets			2 octets		2 octets

Exemple : lecture des mots %MW20 à %MW24 de l'Esclave 6

Question	06	04	14	05	CRC 16
----------	----	----	----	----	--------

Réponse	06	04	0A	xxxx	xxxx	CRC 16
			Valeur de %MW20			Valeur de %MW24	

2.3-3 Ecriture d'un bit interne %Mi

Fonction 05

Question :

Numéro Esclave	5	N° du bit PF Pf		Valeur du bit	CRC 16
1 octet	1 octet	2 octets		2 octets	2 octets

Le champ "Valeur du bit" a deux valeurs possibles à l'exclusion de tout autre :

- bit à 0 = 0000,
- bit à 1 = FF00.

Réponse :

Numéro Esclave	5	N° du bit PF Pf		Valeur du bit	CRC 16
1 octet	1 octet	2 octets		2 octets	2 octets

Exemple : écriture de la valeur 1 dans le bit %M3 de l'Esclave 2

Question	02	05	03	FF00	CRC 16
-----------------	----	----	----	------	--------

Réponse	02	05	03	FF00	CRC 16
----------------	----	----	----	------	--------

2.3-4 Ecriture d'un mot interne %MWi

Fonction 06

Question :

Numéro Esclave	6	N° du mot		Valeur du mot		CRC 16
		PF	Pf	PF	Pf	
1 octet	1 octet	2 octets		2 octets		2 octets

Réponse :

Numéro Esclave	6	N° du mot		Valeur du mot		CRC 16
		PF	Pf	PF	Pf	
1 octet	1 octet	2 octets		2 octets		2 octets

Exemple : écriture de la valeur 16#3A15 dans le mot %MW12 de l'Esclave 5

Question	05	06	0C	3A15	CRC 16
----------	----	----	----	------	--------

Réponse	05	06	0C	3A15	CRC 16
---------	----	----	----	------	--------

2.3-5 Ecriture de n bits internes %Mi

Fonction 15

Question :

Numéro Esclave	0F	Adresse 1er bit à écrire	Nombre de bits à écrire	Nombre d'octets	Valeur des bits à écrire	CRC 16
1 octet	1 octet	2 octets	2 octets	1 octet	n octets	2 octets

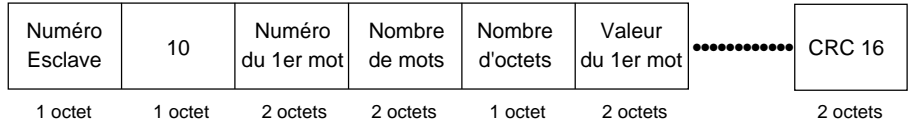
Réponse :

Numéro Esclave	0F	Adresse 1er bit écrit	Nombre de bits écrits	CRC 16
1 octet	1 octet	2 octets	2 octets	2 octets

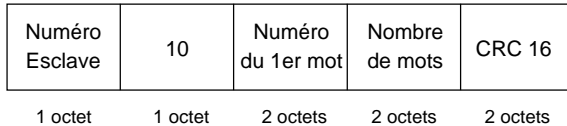
2.3-6 Ecriture de n mots internes %MWi

Fonction 16 (H'10')

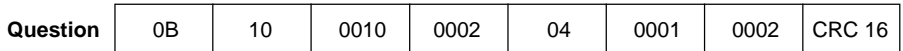
Question :



Réponse :



Exemple : Ecriture des valeurs 1 et 2 dans les mots %MW16 et %MW17 de l'Esclave 11



2.3-7 Calcul du LRC

LRC: somme en hexadécimal, modulo FF, du contenu de la trame, hors en-têtes, complémentée à 2 et codée en ASCII.

Exemple de trame

Trame ASCII

3A	30	31	30	38	30	30	30	30	36	31	36	32	33	34	0D	0A
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Trame binaire équivalente

01	08	00	00	61	62	4872
----	----	----	----	----	----	------

Calcul du LRC

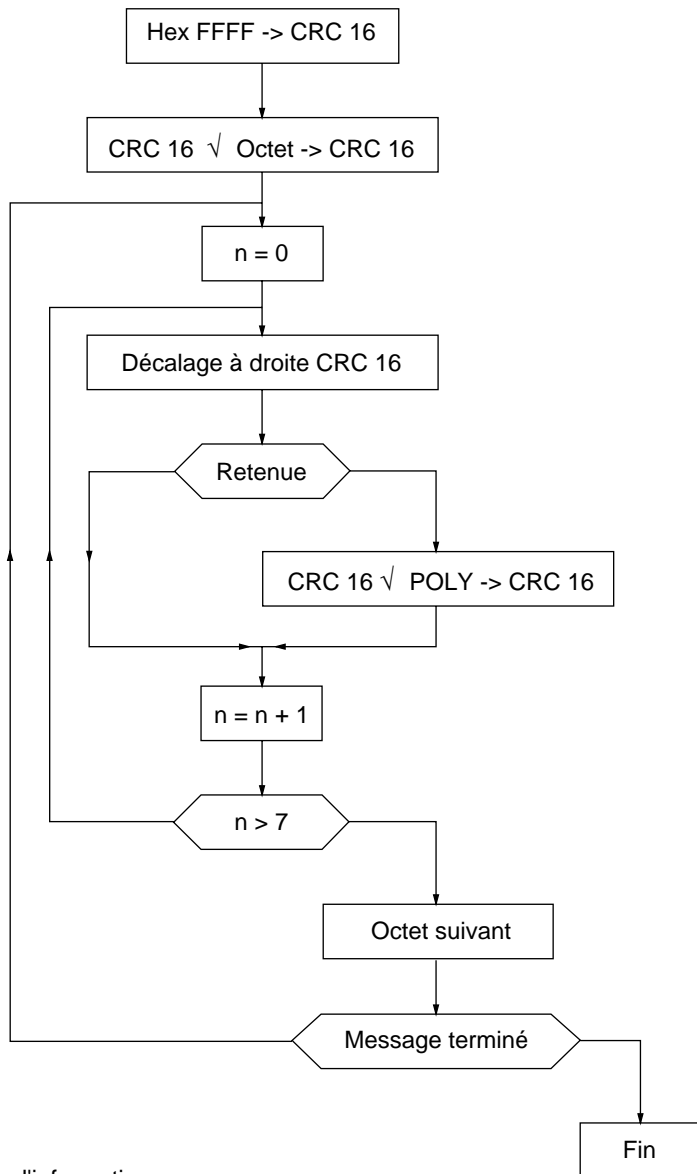
Somme en hexa, modulo FF, du contenu de la trame:

$$01 + 08 + 00 + 00 + 61 + 62 = CCH = 1100\ 1100$$

complément à 1:	0011	0011	
ajout 1:	0011	+ 1	
	0011	0100	
conversion en Hexadécimal	3	4	
Codage en ASCII	33	34	

LRC = 3334

2.3-7 Algorithme de calcul du CRC 16



√ = ou exclusif

n = nombre de bits d'information

POLY = polynôme de calcul du CRC 16 = 1010 0000 0000 0001

(polynôme générateur = $2 = x^2 = x^{15} = x^{16}$)

Dans le CRC 16, le 1^{er} octet émis est celui des poids faibles.

2.4 Requêtes d'accès au serveur UNITE du TSX Nano

Le principe consiste à coder une requête UNITE dans une requête MODBUS afin de solliciter le serveur UNITE habituellement au service de la prise terminal. Ce mécanisme utilise le code fonction spécifique: 65.

Principe:

Requête MODBUS Maître vers Esclave

Code fonction: 65	Code requête	Code catégorie	Données
-------------------	--------------	----------------	---------

Réponse MODBUS Esclave vers Maître

Code fonction: 65	Code réponse	Données réponse
-------------------	--------------	-----------------

Le code catégorie est toujours égal à 7.

Le code réponse UNITE permet de préciser le résultat de l'opération réalisée par le serveur.

On distingue trois cas :

- Code réponse de la requête = Code requête + 30 H: L'opération a été exécutée par le serveur, des informations complémentaires précisent le résultat de l'opération.
- Code réponse positive = 16#00FE: L'opération s'est correctement déroulée et aucune information complémentaire n'est transmise dans la réponse.
- Code réponse négative = 16#00FD: L'opération n'a pu être exécutée par le serveur (requête inconnue, valeur hors bornes etc...).

2.4-1 Identification

Ce service permet de fournir des informations d'identification et de structuration sur l'équipement dont le serveur UNITE est destinataire de la requête.

La requête Identification fournit également un minimum d'information de diagnostic, en donnant l'état des voyants et les Status équipement et module.

Code requête: 0F

Code réponse: 3F

Format du compte rendu: 27 octets

- **Type identification** (octet): Octet toujours égal à H'FF'.
- **Gamme produit** (octet): Ce paramètre identifie la gamme produit à laquelle appartient l'équipement: H'0B' = Gamme TSX Nano.
- **Version** (octet): Numéro de version de l'équipement, il est codé en deux quartets BCD: H'30' pour TSX07 30/31.
- **Chaine ASCII** (chaîne de caractères): Précise la référence commerciale du produit. Sur le TSX Nano elle se compose d'un octet longueur (H'10') suivi du texte ASCII (16 octets) suivants:
'TSX 07 3'
'1-' si présence horodateur, '0-' si absence horodateur
'10 ' si 10 I/O, '16 ' si 16 I/O, '16AC' si 16 I/O AC, '24 ' si 24 I/O
H'20'
H'20'
- **Etat équipement** (octet): Octet toujours égal à 0 sur le TSX Nano
- **Etat des voyants** (table de 8 bits): Cet octet indique l'état des 4 voyants de signalisation du TSX Nano:
RUN (bits 0 et 1), DEF (bits 2 et 3), I/O (bits 4 et 5), COM (bits 6 et 7).
Chaque voyant est codée sur 2 bits:
00: Eteint
01: clignotant
10: Allumé
11: Non significatif.
- **Type métier module de base** (octet): Octet égal à H'30' (type = CPU)
- **Référence du fabricant** (octet): Octet égal à H'09'
- **Référence catalogue** (octet): Désigne le type d'équipement:
H'01': Module 10 E/S
H'11': Module 16 E/S
H'21': Module 24 E/S
H'12': Module 16 E/S AC

- **Status module de base** (table de 8 bits): Les valeurs de ce champs sont conformes au standard des équipements configurables. Un bit positionné à 1 indique un défaut.

Num bit	Classe	Commentaires
0	DEF-INT	1: Défaut matériel TSX Nano
1	DEF-EXT	1: Défaut alimentation capteurss
2	Réservé	
3	MOD-At	1: Module en autotest
4	Réservé	
5	DEF-CNF	1: Extension configurée mais absente ou NOK
6	Réservé	
7	Réservé	

- **Nombre de composants** (octet): Octet égal à 0 car le module ne possède pas de sous modules.

2.4-2 Read-CPU

Ce service est utilisé pour diagnostiquer l'état du processeur d'un automate TSX Nano.

Code requête: 4F

Code réponse: 7F

Format du compte rendu (14 octets):

- **Extension** (byte): Ce champ est utilisé pour émettre un numéro de transaction: Non significatif.
- **Etat des leds** (table de 8 bits): voir requête Identification.
- **Status PLC** (table de 8 bits): Décrit l'état physique de l'automate:
 - Bit 0: RUN (1), STOP (0)
 - Bit 1: Application exécutable (1)
 - Bit 2: Cartouche présente (toujours 1)
 - Bit 3: Forçage en cours (1)
 - Bit 4: Réservé: 0
 - Bit 5: Défaut software (1)
 - Bit 6: Réservé: 0
 - Bit 7: Réservé: 0
- **Adresse du réservant** (table de 6 octets): Décrit l'adresse réseau de l'entité application qui a reservé le processeur. Sur le TSX Nano, cette valeur est fixée à 16#00FF pour chaque octet, pour signaler qu'aucune reservation n'est en cours.

-
- **Type d'erreur application (octet):** Octet toujours à 0.
 - **Information Debug** (table de 8 bits):
 Bit 0: Etat du forçage (1: Actif, 0: Pas de forçage)
 Bits 1 à 7: Toujours à 0.
 - **Gamme produit** (octet): Ce paramètre identifie la gamme produit à laquelle appartient l'équipement: H'0B' = Gamme TSX Nano.
 - **Information application / Automate** (table de 8 bits):
 Bit 0: Présence application en RAM (1)
 Bit1: Programme en RAM avec checksum OK (1)
 Bit 2: Programme en RAM exécutable (1)
 Bit 3: Programme en RAM protégé (1)
 Bit 4: Programme en RAM différent du programme EEPROM (1)
 Bit 5: Application compatible avec l'OS (1)
 Bit 6: Présence horodateur (1)
 Bit 7: 0
 - **Information Backup** (Table de 8 bits):
 Bit 0: Application présente (1)
 Bit1: Checksum OK (1)
 Bit 2: Application EEPROM exécutable (1)
 Bit 3: Application EEPROM protégée (1)
 Bit 4: Application Auto load (1)
 Bit 5: Application compatible avec l'OS (1)
 Bit 6: 0
 Bit 7: 0
-

2.4-3 Requête RUN

Code requête: 24

Code réponse: FE

Résultat négatif si:

- Entrée RUN / STOP configurée en STOP
- Application non exécutable
- Défaut Software (débordement chien de garde, lancement d'une étape G7 non existante).

L'envoi de cette requête vers un TSX Nano déjà en RUN n'est pas considéré comme une erreur.

2.4-4 Requête STOP

Code requête: 25

Code réponse: FE

Résultat négatif si application en RAM non exécutable.

Le passage en STOP d'un TSX Nano déjà en STOP n'est pas considéré comme une erreur.

2.4-5 Requête INIT

Code requête: 33

Code réponse 63

- **Type d'initialisation** (octet à emettre): Doit être toujours à 1.
- **Compte rendu**: Indique le résultat de l'opération d'initialisation:
 - 00: Initialisation OK
 - 01: Type d'initialisation différent de 1

2.5 Limitations

Le TSX Nano Esclave sur MODBUS a les limitations suivantes:

- **Couche physique:** RS485 uniquement. Pas d'adaptation automatique à la vitesse du Maître.
- **Position du sélecteur:** Le protocole MODBUS n'est valide que si la position du sélecteur rotatif de l'automate (lue uniquement à la mise sous tension) indique une utilisation du TSX Nano **en automate** (positions 0, 5, 6 et 7).
- **Protocole:** Le protocole est uniquement Esclave.
- **Communication d'Esclave à Esclave:** La communication directe d'Esclave à Esclave est impossible. Elle ne peut être réalisée que par le programme application du Maître.
- **Le passage en dynamique du mode ASCII au mode RTU** n'est pas possible.



Chapitre	Page
A.1 Accès aux instructions par le terminal FTX 117	A/1
A.2 Accès aux variables par le terminal FTX 117	A/3
A.3 Accès aux fonctions du terminal FTX 117	A/4
A.4 Codes erreur de la fonction diagnostic	A/6
A.5 Temps d'exécution et occupation mémoire des instructions	A/7
A.6 Complément sur le fonctionnement de la sortie SECURITE	A/11
A.7 Exemple de prépositionnement d'étapes Grafcet	A/12
A.8 Programmation des actions associées aux étapes Grafcet	A/13
A.9 Correspondance instructions List et instructions langage à contacts	A/14
A.10 Index	A/15



A.1 Accès aux instructions par le terminal FTX117

Instructions booléennes

Instruction	Touche	Instruction	Touche
LD	l	AND	a
LDN	l n	ANDN	a n
LDR	l R	ANDR	a R
LDF	l R R	ANDF	a R R
ST	s	OR	o
STN	s n	ORN	o n
S	Σ	ORR	o R
R	r	ORF	o R R
((1)	(XOR	x
))	XORN	x n
MPS	&	XORR	x R
MRD	é	XORF	x R R
MPP	"	N	n

Instructions sur blocs fonction

Instruction	Touche	Instruction	Touche
BLK	b	I	f 3 z
OUT_BLK	B	O	f 4 z
END_BLK	B B	U	f 5 z
IN	i	S	Σ
R	r		
CU	©		
CD	ð		

Instructions numériques

Instruction	Touche	Instruction	Touche
:=	[1 z	NOT	[1 2 z
+	[2 z	SHL	[1 3 z
-	[3 z	SHR	[1 4 z
*	[4 z	ROL	[1 5 z
/	[5 z	ROR	[1 6 z
REM	[6 z	BTI	[1 7 z
INC	[7 z	ITB	[1 8 z
DEC	[8 z	SQRT	[1 9 z
AND	[9 z	EXCH	[2 0 z
OR	[1 0 z	READ	[2 1 z
XOR	[1 1 z		

(1) doit être précédé d'une instruction booléenne LD, AND ou OR.

Instructions numériques de comparaison

Saisir au préalable l'instruction booléenne : LD, AND ou OR

Instruction	Touche	Instruction	Touche
=	[1 z	>	[4 z
<>	[2 z	<=	[5 z
<	[3 z	>=	[6 z

Instructions Grafcet

Instruction	Touche	Instruction	Touche
=*=	f 8 z	.*-	f 9 z
#	#	#D	f 1 0 z
=*=POST	f 1 1 z		

Instructions sur programme

Instruction	Touche	Instruction	Touche
END	ê	JMP	j
ENDC	ê c	JMPC	j c
ENDCN	êc n	JMPCN	j c n
SR	f1 z	%L	L
RET	f2 z	MCS	f 6 z
NOP	N	MCR	f 7 z

A.2 Accès aux variables par le terminal FTX117

Opérandes bits

Objet	Touche	Objet	Touche
%I	I	%S	S
%Q	Q	%X	v 9 z
%M	M		

Opérandes mots

Objet	Touche	Objet	Touche
%MW	μ	%IW	v 1 z
%KW	È	%QW	v 2 z
%SW	Ò		

Opérandes bits extraits de mots <Objet mot> <N°mot> h : <N°bit>

Exemple : %KW12:X5 : È 1 2 h : 5

Opérandes de blocs fonction

Objet	Touche	Objet	Touche
%C	C	%TM	†
%Ci.P	C<i>.	%TMi.P	†<i>.
%Ci.V	C<i>."	%TMi.V	†<i>."
%Ci.E	C<i>z	%TMi.Q	†<i>z
%Ci.D	C<i>z"	Q	v15z
%Ci.F	C<i>z" "	%DR	v11z
E	v12z	%DRi.S	v11z<i>.
D	v13z	%DRi.F	v11z<i>z
F	v14z		
%R	v10z	%Ri.E	v10zz"
%Ri.I	v10z<i>.	F	v14z
%Ri.O	v10z<i>"	E	v12z
%Ri.F	v10zz		
%PLS	v6z	%FC	v5z
%PLS.P	v6z .	%FC.S0	v5z .
%PLS.N	v6z ."	%FC.S1	v5z ."
%PLS.D	v6zz	%FC.P	v5z .""
%PLS.Q	v6zz"	%FC.V	v5z . 3 fois "
Q	v15z	%FC.F	v5zz
D	v13z	%FC.TH0	v5zz"
%PWM	v7z	%FC.TH1	v5zz""
%PWM.R	v7z .	F	v14z
%SBR	v3z	TH0	v17z
%SBRi.j	v3z<i> . <j>	TH1	v18z
%MSG	v8z	%SC	v4z
%MSG.D	v8zz	%SCi.j	v4z<i> . <j>
%MSG.E	v8zz"		

A.3 Accès aux fonctions par le terminal FTX117

Fonction	Sélection	Mode	Voir ch. Inter C
Affichage (choix d'une base d')	e2m4m2	TSX	8.2
	e3m2	TSX	9.1
Archivage d'une application	e2m1	FTX/TSX	6
Blocs fonction (Configuration)	e4m2	FTX/TSX	3.3
Communication (prise console)	e4m7	FTX/TSX	3.9
Comparaison	e2m1>	FTX/TSX	6
Compteur rapide (Configuration)	e4m3 5 fois "	FTX/TSX	3.4-3
Constantes (Saisie)	e4m5	FTX/TSX	3.6
Copie d'une sélection	e2m7	FTX/TSX	4.3-5
Diagnostic programme	e2m3	FTX/TSX	4.5
Effacement d'une application	e2m2	FTX/TSX	4.6
Effacement d'une ligne	e2hd	FTX/TSX	4.5
Effacement configuration	e4m6	FTX/TSX	3.7
Effacement programme	e2m2>	FTX/TSX	4.6
Entrées (Configuration)	e4m3	FTX/TSX	3.4
Entrées RUN/STOP(Config.)	e4m1	FTX/TSX	3.2
Etat automate (Visualisation)	e4m0	TSX	7.5
Forçage d'E/S	e3I0z	TSX	8.3/9.4
Horodateur (Programmation)	e13	FTX/TSX	5.1
Initialisation d'une application	e12	TSX	7.3
Insertion d'une ligne	e2i	FTX/TSX	4.3-3
Langue (Choix de la)	e52	FTX/TSX	2.3
Mise à l'heure	e14	TSX	7.2
Mise au point	e2m4	TSX	8
	e1h>	TSX	8
Mode scrutation (Choix)	e4m1 " "	FTX/TSX	3.2
Modification de programme	e2	FTX/TSX	4.3
Modification en RUN	e2	TSX	8.5
Mot de passe (Création/saisie)	e2m0	FTX/TSX	4.7
Nom de l'application (Saisie)	e2m0	FTX/TSX	4.7
Poursuivre une recherche	e2m51	FTX/TSX	4.4
Programmer l'application	e2	FTX/TSX	4.1

Fonction	Sélection	Mode	Voir ch. Inter C
Protéger en lecture	e2m1	FTX/TSX	6.5
Recherche	d'une ligne	FTX/TSX	4.4-1
	d'une instruction	FTX/TSX	4.4-3
	d'une variable	FTX/TSX	4.4-2
	poursuite d'une	FTX/TSX	4.4
Remplacement	e2m55	FTX/TSX	4.4-4
Réglage de variables	e3	TSX	9
Retour son (action/inhibition)	e53	FTX/TSX	2.4
Retro-éclairage (durée)	e53 "	FTX/TSX	2.4
RUN automate	e11z	TSX	7.4
Sauvegarde d'application	e2m1	TSX	6
Sélection d'une partie de prog.	e2m6	FTX/TSX	4.3-5
Sorties (Configuration)	e4m4	FTX/TSX	3.5
Sortie sécurité (Configuration)	e4m1	FTX/TSX	3.2
Sortie RUN/STOP (Config.)	e4m1 "	FTX/TSX	3.2
STOP automate	e11z	TSX	7.4
Table de données (Créer)	e3m5 à 8	FTX/TSX	9.8
Table de données (Effacer)	e3m9	FTX/TSX	9.9
Table de variables (Sauvegarder)	e3m1	FTX/TSX	9.10
Taille du programme	e2m0	FTX/TSX	4.7
Transfert de programme	e2m1	FTX/TSX	6
Versions logicielles	e0	FTX/TSX	2.1

A.4 Codes erreur de la fonction diagnostic

Code	Instruction	Explication
0	JMP %Li	Saut à une étiquette (Label) qui n'a pas été définie dans le programme.
1	SRi	Saut à un sous-programme qui n'a pas été défini dans le programme.
2	#i #Di	Désactivation d'une étape qui n'a pas été définie dans le programme.
3	JMP %Li	Saut à une étiquette (Label) qui a été définie dans une suite d'instructions situées entre parenthèses.
4	%Li:	Étiquette (Label) définie dans une suite d'instructions situées entre parenthèses.
5	SRi:	Définition d'un sous programme entre parenthèses
6	(Nombre d'imbrications de parenthèses supérieur à 8
7)	Parenthèse fermante non précédée d'une parenthèse ouvrante.
8	RET	Instruction de fin de sous programme entre parenthèses
9)	Toutes les parenthèses ne sont pas fermées
10	BLK	Un bloc fonction n'est pas fermé, instruction BLK non suivie d'une instruction END_BLK.
11	END_BLK OUT_BLK	Un bloc est fermé, sans avoir été ouvert précédemment instruction END_BLK ou OUT_BLK non précédée d'une instruction BLK.
12	%Li:	Définition d'une étiquette (Label) non suivie d'une instruction LD ou BLK.
13	SRi:	Définition d'un sous-programme non suivi d'une instruction LD ou BLK.
14	RET	Fin de sous programme sans définition de sous programme préalable.
15	MPS	Nombre d'imbrication d'instruction MPS supérieur à 8.
16	MRD	Instruction MRD non précédée d'une instruction MPS
17	MPP	Instruction MPP non précédée d'une instruction MPS
18	POST	Utilisation d'une instruction Grafcet dans le traitement préliminaire ou dans le traitement postérieur
19	-*i =*i	Définition d'étape Grafcet non suivie d'une instruction LD.
20	MPS	Nombre d'instructions MPS supérieur à celui des instructions MPP.
21	SRi:	Un sous programme ne se termine pas par l'instruction RET.

A.5 Temps d'exécution et occupation mémoire des instructions

Les temps sont exprimés en μ s.

La taille est exprimée en octets.

Instructions booléennes

Instructions	Avec opérandes bits			
	%I,%Q		%M, %S, %X, 0/1, %MWi:Xi... (1)	
	Temps	Taille	Temps	Taille
LD, LDN	0,2	2	0,4	4
LDR, LDF	0,5	4	-	-
AND, ANDN	0,2	2	0,6	4
ANDR,ANDF	0,8	4	-	-
AND(, AND(N	6,3	8	7	10
AND(R,AND(F	7	10	-	-
OR, ORN	0,4	2	0,7	4
ORR, ORF	0,8	4	-	-
OR(, OR(N	6,3	8	6,6	10
OR(R, OR(F	6,7	10	-	-
XOR,XORN	0,2	2	0,6	4
XORR,XORF	0,8	4	-	-
ST, STN (2)	0,2 (0,8)	2	0,9 (1,5)	4
S,R	0,8 (1,2)	4	1,5 (1,8)	6
N	0,6	4		
)	15	8		
MPS	0,6	4		
MRD	0,2	2		
MPP	0,2	2		

Ces temps sont à multiplier par 3 lorsque les instructions sont écrites au delà de la ligne programme 100.

- (1) Pour les bits extraits de mots %MW16 à %MW255 ou pour tous les autres types (%KW_i:X_j, %SW_i:X_j) ces temps sont multipliés par 1,5 et les tailles sont majorées de 2 octets.
- (2) Les temps donnés entre parenthèses correspondent aux temps d'exécution des instructions lorsque l'application a été initialisée avec utilisation des instructions MCS/MCR

Rappel : le choix de l'utilisation ou pas des instructions relais maître s'effectue lors de l'effacement de la mémoire application (voir ch4.6, intercalaire C).

Instruction sur blocs fonction (en programmation réversible)

Instruction	Temps d'exécution (en µs)	Occupation mémoire (en octets)
BLK %Tmi	46	4
BLK %Ci	45	4
BLK %Ri	48	4
BLK %SBRi	38	4
BLK %SCi	36	4
BLK %DRi		
BLK %FC	42	4
BLK %MSG	22	4
BLK %PMW	42	4
BLK %PLS	53	4
OUT_BLK	5,6	2
END_BLK		2
IN	1,2	4
R	0,6	4
CU	0,7	4
CD	0,7	4
I	1	4
O	1	4
U		
S	0,7	4

Instruction sur blocs fonction (en programmation non réversible)

Instruction	Temps d'exécution (en µs)	Occupation mémoire (en octets)
IN %Tmi	48	4
CD %Ci (CU %Ci)	46	4
S %Ci	49	4
R %Ci	47	4
U %DRi		
R %DRi		
LD %SCi.j	9	6
CD %SCi (CU %SCi)	38	4
ST %SCi.j	10	6
R %SCi	36	4
BLK %PMW	42	4
BLK %PLS	53	4
CD %SBRi (CU %SBRi)	39	4
R %SBRi	37	4
I %Ri (O %Ri)	49	4
R %Ri	48	4
IN %PWM	36	4
IN %PLS	46	4
S %PLS	42	4
R %PLS	58	4
IN %FC	43	4
S %FC	69	4
READ	9,8	6
SEND	10	8
R %MSG	25	4

Instructions numériques

Instruction	Temps d'exécution (en μ s)	Occupation mémoire (en octets)
:=	29,5	10
+	34	12
-	38	12
*	49	12
/	48	12
REM	49	12
INC	28	6
DEC	28	6
AND	37	12
OR	37	12
XOR	37	12
NOT	29	8
SHL	34	10
SHR	34	10
ROL	35	10
ROR	35	10
BTI	40	8
ITB	40	8
SQRT	80	8

Instructions de comparaison

Instruction	Temps d'exécution (en μ s)	Occupation mémoire (en octets)
LD[mot1 comp mot2]	18	8
AND[mot1 comp mot2]	19	10
AND([mot1 comp mot2]	24	14
OR[mot1 comp mot2]	21	10
OR([mot1 comp mot2]	25	14

comp : opérations de comparaison =>,<=<>,<=>,<=>

Instructions Grafcet

Instruction	Temps d'exécution (en μ s)	Occupation mémoire (en octets)
=*= i		4
#		4
#i		4
#Di		6
=*=POST		4
-*- i		6

Instruction sur programme

Instruction	Temps d'exécution (µs)	Occupation mémoire (en octets)
END	0,4	2
ENDC, ENDCN	0,6	4
SR	14	4
RET	2	6
NOP	0,4	2
JMP	7,8	4
JMPC, JMPCN	8	6
%Ln :	0,6	4
%SRn :	2	4
MCR	0,5	2
MCS	2,5	12

A.6 Complément sur le fonctionnement de la sortie SECURITE

Comme indiqué au chapitre A4.2, la sortie SECURITE est à l'état 1 quand l'automate est en RUN et si aucun défaut bloquant n'est détecté.

Aucun défaut bloquant n'est détecté lorsque toutes les conditions ci-dessous sont réunies :

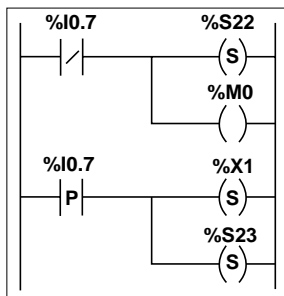
- Application exécutable. Le voyant ERR est éteint.
 - Aucun défaut d'entrées/sorties soit :
 - sorties statiques protégées non disjonctées,
 - aucun défaut alimentation capteurs,
 - pas de défaut matériel,
 - pas de défaut de communication,
 - configuration conforme.
- Le voyant I/O est éteint et le bit système %S10 est à 1.
- Pas de débordement de chien de garde logiciel. Le voyant RUN ne clignote pas et le bit système %S11 est à 0.
 - Pas de défaut d'exécution logicielle (pas d'appel d'une fonction non implémentée ou interdite, pas de lancement d'une étape Grafcet non programmée). Le voyant RUN ne clignote pas.
 - Pas de mise à zéro des sorties par le bit système %S9. Le bit système %S9 est à 0.

A.7 Exemple de prépositionnement d'étapes Grafcet

Dans cet exemple, aucune étape initiale (=*=i) n'est programmée, l'initialisation du graphe s'effectuant sur front montant de l'entrée %I0.7 dans le traitement préliminaire. L'exemple du chapitre B 2.3-2 est repris ci-dessous.

Traitement préliminaire

Dans le traitement préliminaire, l'état 0 de l'entrée %I0.7 provoque la mise à zéro du Grafcet (mise à 1 du bit système %S22) ce qui entraîne la désactivation des étapes actives sur l'ensemble du Grafcet.

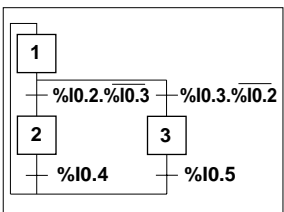


000	LDN	%I0.7
001	S	%S22
002	ST	%M0
003	LDR	%I0.7
004	S	%X1
005	S	%S23

Le front montant de l'entrée %I0.7 provoque le prépositionnement du graphe (activation de l'étape X1 associée à la mise à 1 du bit système %S23).

Traitement séquentiel

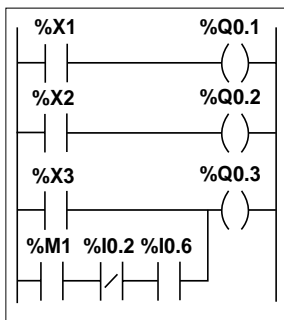
Aucune étape initiale est déclarée, l'étape 1 étant initialisée par l'entrée I0.7 à partir du traitement préliminaire.



006	-*-	1
007	LD	%I0.2
008	ANDN	%I0.3
009	#	2
010	LD	%I0.3
011	ANDN	%I0.2
012	#	3
013	-*-	2
014	LD	%I0.4
015	#	1
016	-*-	3
017	LD	%I0.5
018	#	1

Traitement postérieur

Le traitement postérieur, placé en fin de programme, commence à partir de l'instruction =*=POST.



019	=*=	POST
020	LD	%X1
021	ST	%Q0.1
022	LD	%X2
023	ST	%Q0.2
024	LD	%X3
025	OR(%M1
026	ANDN	%I0.2
027	AND	%I0.6
028)	
029	ST	%Q0.3

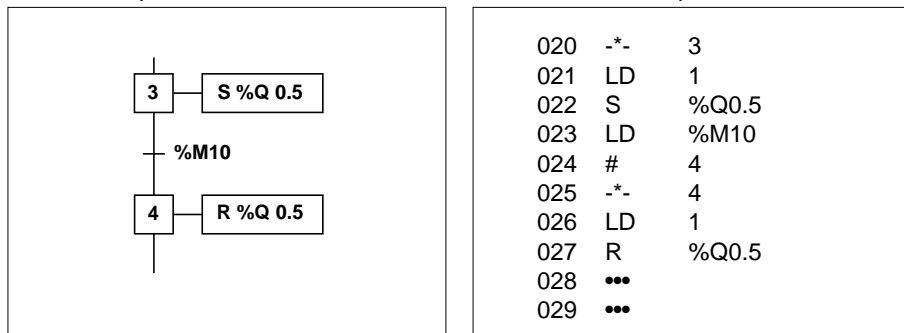
A.8 Programmation des actions associées aux étapes Grafcet

Les actions associées aux étapes peuvent se former de deux façons :

- soit par phrases List ou réseaux de contacts associés à l'étape. Dans ce cas les phrases List ou les réseaux de contacts ne sont scrutés que si l'étape à laquelle ils sont associés est active.
- soit dans le traitement postérieur par l'utilisation du bit étape %Xi (voir exemple chapitre B 2.3-2). Dans ce cas la scrutation a lieu à chaque cycle automate.

Pour des raisons de sécurité (scrutation du programme), il est préférable que les actions soient écrites dans le traitement postérieur.

Dans l'exemple ci-dessous, des actions sont associées aux étapes 3 et 4.



Les actions peuvent être asservies ou non à des conditions logiques et sont saisies sous forme de bobines ou instructions à l'enclenchement (S). Toute bobine ou instruction (S) doit obligatoirement être remise à zéro par une bobine ou instruction (R).

A.9 Correspondances instructions List et instructions langage à contacts

Instuction LIST	Equivalent Ladder	Désignation
LD, LDN, LDR, LDF		Chargement
ST, STN, R, S		Affectation
AND, ANDN, ANDR,ANDF		ET logique
OR, ORN, ORR, ORF		OU logique
AND (, OR ((8 niveaux possibles)		Parenthèses
XOR, XORN, XORR, XORF		OU exclusif
N	Non réversible	Négation
END, ENDC, ENDCN (ENDCN non réversible)		Fin de programme
%Li		Définition Etiquette
JMP, JMPC, JMPCN (JMPCN non réversible)		Saut vers étiquette 0< i < 16
SRn		Sous-programme
RET		Retour Sous-programme
MCR, MCS		Relais maître
NOP	Non réversible	Instruction nulle

A.10 Index

A

Addition	B 3/9
Adressage	
entrées/sorties	A 1/12
indexé	B 3/4
Affectation	B 3/5
Affichage	
bits internes	A 1/20
E/S	A 1/19
étapes Grafcet	B 2/26
état	A 1/19
Alimentation	
FTX 117	C 1/2
TSX 07	A 3/4, A 5/1
AND (sur mot)	B 3/11
AND, ANDN, ANDF, ANDR	B 2/5
Analogique (E/S)	A 1/23, B 4/1
Animation programme	C 8/1
Animation données	C 8/2, C 9/3, C 9/5
Application	C 7/1
Arithmétique	B 3/9, C 4/2
Arrêt/mise en route de l'automate	
par entrée RUN/STOP	A 4/1
ASCII (base de visualisation)	C 8/2, C 9/3
ASCII communication	
voir Communication	

B

Batterie TSX	A 1/5
Bits (liste)	B 2/1
d'entrées	A 1/12
de sorties	A 1/12
extraits de mots	B 3/2
internes	B 2/1
système	B 6/1
BLK, END_BLK, OUT_BLK	B 2/12
Blocs fonctions	
accès	C 1/10, G A/4
visualisation	C 9/4

programmation (principe)	B 2/12
configuration	C 3/2
Bloc horodateur (RTC)	B 5/1, C 5/1, C 7/1
BT (base de temps)	
%PLS	B 3/18
%PWM	B 3/16
%TM	B 2/13
BTI	B 3/13, C 4/2
Buzzer	C 2/2

C

Câblage TSX 07	A 3/1
Câbles et accessoires	A 3/13, F 1/4
Carte mémoire	C 6/1
transfert d'application	C 6/2
transfert de tables de données	C 9/3
CD (compteur/décompteur)	
%C	B 2/17
%SBR	B 3/44, C 4/2
%SC	B 3/46, C 4/2
Chaînes de bits	B 3/3
Chien de garde	A 1/8...A 1/10, D 1/1
Clavier FTX 117	C 1/1, C 1/4
Code BCD (rappel)	B 3/13
Codes d'erreur	
carte mémoire	C 9/6
diagnostic	G A/6
stop automate	C 7/1
Coller	C 4/6
COM (voyant)	A 1/19, D 1/1
Combinatoire et séquentielle	
(instruction)	B Chap.2
Communication ASCII	
câblage	F 1/1
description	B 3/29, B 3/32
Communication inter-automate	
câblage	A 3/14
description	A 1/13
utilisation/programme	B 3/48
mots d'échange	C 3/2
Communication UNI-TELWAY	
câblage	F 1/1
description	B 3/29, B 3/36
requêtes	F 1/10

utilisation/programme B 3/32
 Commutateur de sélection d'adresse A 1/3, A 1/11
 Comparaison
 instruction B 3/8
 applications C 6/2
 Compatibilité Préambule
 Compteur rapide
 configuration C 3/6
 fonction A 4/6, B 3/24, C 4/2
 Compteur/décompteur
 configuration C 3/6
 fonction A 4/6, B 3/24, C 4/2
 Conditions de service A 5/7
 Configuration
 éditeur (Cnf) C 1/9, C 3/1
 application C 3/2
 blocs fonctions C 3/3, C 3/8
 entrées C 3/5
 sorties C 3/7
 Constantes B 3/2
 Conversion
 affichage (décimal,
 hexadécimal, ASCII, binaire) C 8/2
 BCD B 3/13, C 4/2
 Création programme C 4/1
 Copier C 4/6
 CU (%C, %SBR, %SC) B 2/16, C 4/2
 Cycle automate A 1/8...A 1/10
 normal, périodique A 1/8, C 3/2

D

D %C B 2/16
 %MSG B 3/29
 %PLS B 3/18
 Date et heure
 courante C 7/1
 dernier arrêt B 5/3, C 7/1
 Débit binaire
 liaison extension C 3/2
 prise terminal F 1/1
 Débordement

index B 3/4
 tâche A 1/8, C 7/1
 Décalage
 (SHR, SHL, ROL, ROR) B 3/12, C 4/2
 DEC B 3/9, C 4/2
 Défaut
 analyse D 1/4
 visualisation D 1/1
 Dépannage D 1/1
 Déplacer C 4/6
 Description
 matérielle FTX C 1/1
 matérielle TSX A 1/3
 Diagnostic
 automate D 1/1
 programme C 4/10, G A/6
 Dimensions (encombrements) A 2/1, C 1/1
 Disjonction (sorties statiques) B 6/4
 Division B 3/9

E

E %C B 2/17, C 4/2
 %MSG B 3/30, C 4/2
 %R B 2/20, C 4/2
 E/S (voyant) A 1/19, D 1/2
 Ecran FTX 117 C 1/4, C 2/2
 Editeur
 automate C 1/7, C 1/8, C 1/10
 de configuration C 1/9, C 3/1
 de données C 1/9, C 9/1
 de programme C 1/8, C 4/1
 terminal C 2/1
 Effacement mémoire C 4/11
 Effacer une (des) instruction(s) C 4/5, C 4/6
 Encombrements
 automate A 2/2
 terminal C 1/1
 END, ENDC, ENDCN B 2/29
 Entrée Analogique
 Présentation A 1/23, B 4/1
 Caractéristiques B 4/4
 Mise en oeuvre B 4/2
 Raccordement A 3/15

Entrée RUN/STOP A 4/1, C 3/2

Entrées

adressage A 1/12

caractéristiques A 5/2

spécifiques A 1/13

Entrées 115 VCA

caractéristiques A 5/2

raccordement A 3/7

Entrées 24 VCC

caractéristiques A 5/2

raccordement A 3/5

ERR A 1/19, D 1/1

Etape Grafcet B 2/26, C 4/2, C 9/4

Etat automate A 1/19, D 1/1

Etat de l'automate (Sécurité) A 4/1

Etiquette (label) B 2/30

EXCH B 2/29, B 3/43

Extension automate

généralités A 1/21

configuration C 3/2

raccordement A 3/14

utilisation programme B 3/48

Extension E/S

généralités A 1/11

configuration C 3/2

raccordement A 3/13

F

F %C B 2/17

%DR B 2/23, C 4/2

%FCà B 3/20, C 4/2

%R B 2/20, C 4/2

FC B 3/20, C 4/2

FIFO B 2/20, C 4/2

Filtrage programmable A 1/15, C 3/5

Fonctions (accès) C 1/10, G A/4

Forçage C 8/2

Frequencemètre

configuration C 3/6

fonction A 4/5, B 3/23, C 4/2

Front montant/descendant B 2/2

Fun (touche accès instruction)

non mentionnée sur clavier C 4/2, G A/1

G

Générateur d'impulsions PWM

A 4/8, B 3/16

Grafcet B 2/25, C 4/2, C 6/2, C9/4

H

Hexadécimal B 3/1, C 8/2, C 9/3

Horodateur

description B 5/3

mise à l'heure B 5/4, C 7/1

bloc programmeur B 5/1, C 5/1

datation B 5/3

HSC A 4/4, B 3/21

I

I (%R) B 2/19, C 4/2

I/O A 1/19, D 1/1

IN

%FC B 3/20

%TM B 2/13

%PLS B 3/18

%PWM B 3/16

INC B 3/9, C 4/2

Indexation B 3/4

Informations C 4/12, C 7/2

Initialisation A 7/2, C 7/3

Insérer une (des) instruction(s) C 4/5

ITB B 3/13, C 4/2

J

JMP, JMPC, JMPCN B 2/30

L

Langage

liste d'instructions B 1/1

schéma à contacts B 1/4

Langue terminal (choix) C 2/2

LD, LD[, LDN, LDN[, LDF, LDR B 2/4

LIFO B 2/20, C 4/2

M

Maintenance	D 1/1
MCR, MCS	B 2/32, C 4/2, C 4/11
Mémoire	
automate	A 7/3, C 6/1
EEPROM automate	A 7/3, C 6/1
Flash EPROM FTX 117	C 6/1, C 6/2
RAM automate	A 1/5, C 6/1
RAM FTX 117	C 6/1, C 6/2
Mémorisation d'état	A 4/2, C 3/5
Menu	C 1/7
Mise au point	Chap. 8
Mise en marche/arrêt automate	
définition	C 7/2
Mode autonome FTX 117	C 1/3, C 1/7
Modbus	F Chap. 2
Mode connecté FTX 117	C 1/3, C 1/7
Modification	
de données	C 9/3, C 9/4
de programme	C 4/4
de programme en RUN	C 8/4
Montage	A 2/2
Mot de passe	C 4/12
Mots (définition, liste)	B 3/1
Mots constants	B 3/2, C 3/8
Mots d'échange	B 3/38
Mots indexés	B 3/4
Mots internes	B 3/1
Mots système	B 3/2, B 6/7...B 6/13
MPS, MRD, MPP	B 2/9
MSG (%MSG)	B 3/30
Multiplication	B 3/9

N

N	B 2/7
Négation	B 2/7
Nom application (saisie)	C 4/12
NOP	B 2/29
NOT (sur mot)	B 3/11, C 4/2

O

O (%R)	B 2/19, C 4/2
--------	---------------

OPEN	B 2/10
Opération arithmétique	B 3/9, C 4/2
Opération logique sur mots	B 3/11, C 4/2
OR (sur mot)	B 3/11, C 4/2
OR, ORN, ORF, ORR	B 2/6

P

Parenthèses	B 2/7
Périodique (cyclique)	A 1/8, C 3/2,
Port du terminal	B 3/29, F 1/1
Postérieur (Grafcet)	B 2/28, C 4/2
Potentiomètres	
description	A 1/18
opération	B 3/14
Préférences	C 2/2
Préliminaire (Grafcet)	B 2/27, C 4/2
Prépositionnement (Grafcet)	
	B 2/27, B 6/1
Prise terminal	B 3/29, F 1/1
Programmateur temporel	
	B 4/1, B 5/1, C 5/1
Programmateurs cycliques	B 2/23, C 4/2
Programme	
créer	C 4/1
effacer	C 4/5, C 4/11
généralités	C 4/1
insérer	C 4/4
modifier	C 4/4
sauvegarder	C 6/1
Protection application	C 1/13, C 6/4
Protection par mot de passe	C 4/12
PULSE (PLS)	A 4/7, B 3/18, C 3/7
PWM	A 4/8, B 3/16, C 3/7

Q

Q	
%PLS	B 3/18, C 4/2
%TM	B 2/13, C 3/3

R

R (reset)	
instructions booléennes	B 2/2
%C	B 2/17
%R	B 2/19
%DR	B 2/23
%MSG	B 3/30
%SBR	B 3/34
%SC	B 3/46
Raccordement	
alimentations	A 3/4
entrées analogiques	A 3/15
E/S	A 3/1, A 3/5-3/12
extension automate	A 3/14
extension E/S	A 3/13
sorties analogiques	A 3/16
READ (%FC.V)	B 3/22, B 3/24, C 4/2
Réarmement (sorties statiques)	B 6/5
Rechercher	C 4/7
Registre (blocs fonction)	B 2/20
Registre bits	B 3/44, C 4/2
Registre mots	B 2/20
Réglage analogique	
description	A 1/18
exploitation	B 3/14
Relais maître	
(MCS, MCR)	B 2/31, C 4/2, C 4/11
REM	B 3/9, C 4/2
Remplacer	C 4/9
Reprise secteur	A 7/1, C 5/2
RET	B 2/31, C 4/2
Retour alimentation	A 7/1
Reste (division)	B 3/9
Rétro-éclairage	C 2/2
Réversibilité	B 6/3
ROL	B 3/12, C 4/2
ROR	B 3/12, C 4/2
RTC	B 5/1, C 5/1
RUN	C 7/2
RUN (LED)	A 1/19, D 1/1
RUN/STOP	A 4/1

S

S instructions booléennes	B 2/2
%C	C 2/17
%FC	B 3/20
%SBR	B 3/44
%SC	B 3/46
Saut de programme	B 2/30
Sauvegarde	C 6/1
données dans automate	C 9/6
données (table de données)	C 9/6
programme dans automate	C 6/4
programme par console, (RAM sauvegardée, flash ou carte)	C 6/2
Scrutation	
normale, périodique	A 1/8, C 3/2
SECURITE	A 4/1, C 3/2
Sélecteur d'adresse	
	A 1/3, A 1/11, A 1/21
Sélectionner	C 4/6
SHL	B 3/12, C 4/2
SHORT	B 2/10
SHR	B 3/12, C 4/2
Sortie Analogique	A 1/23
caractéristiques	B 4/4, B 4/7
mise en oeuvre	B 4/5
raccordement	A 3/16
encombrement	A 2/1
Sortie transistor	A 1/13
caractéristiques	A 5/3
raccordement	A 3/9-3/12
Sortie protégée	A 1/17, B 7/4
Sortie sécurité	A 4/1, C 5/17
Sorties	
adressage	A 1/12
caractéristiques	A 5/3
spécifiques	A 1/13
Sorties (24 VCC, récept.)	
caractéristiques	A 5/3
raccordement	A 3/9-3/10
Sorties (24 VCC, source)	
caractéristiques	A 5/3
raccordement	A 3/11-3/12

Sorties relais		d'état	A 1/19
caractéristiques	A 5/4	E/S	A 1/19, C 9/3
raccordement	A 3/7-3/9	étapes Grafcet	C 9/4
Sorties protégées	A 5/3	Vitesse d'exécution	A 1/5, G A/7
Sorties réflexes	A 4/3, B 3/21, C 3/6	Voyant	A 1/19
Sous-programme	B 2/31, C 4/2	Var (touche accès variable)	C 4/2, G A/3
Soustraction	B 3/9, C 4/2		
SQRT	B 3/9, C 4/2	W	
SRi	B 2/30, C 4/2		
ST, STN	B 2/4	Watchdog	A 1/10
STOP	C 7/2	X	

T		XOR (sur mot)	B 3/11, C 4/2
		XOR, XORN, XORF, XORR	B 2/6

Table de données			
création/modification	C 9/5		
transfert	C 9/6	#Di	B 2/26, C 4/2
Tableau de mots	B 3/3	#i	B 2/26
Temporisateurs	B 2/13, B 2/15, C 3/3	%Ci	B 2/17
Temps de cycle		%DRi	B 2/23, C 4/2
accès	C 6/7	%EXCH	B 3/29...B 3/33
description	A 1/5, A 1/8...A 1/10	%FC	B 3/20, C 4/2
exécution	G A/7	%IWxx	B 3/38, C 4/2
lecture terminal	C 3/2	%Ix.y	A 1/12
Traitement séquentiel (Grafcet)	B 2/26	%KWxx	B 3/2
Transfert	C 6/1	%Li:	B 2/25, B 2/30
Transition (Grafcet)	B 2/25, C 4/2	%MSG	B 3/29, C 4/2

U		%MWxx	B 3/1
		%Mxx	B 2/1
		%PLS	B 3/18, C 4/2
U (%DR)	B 2/22, C 4/2	%PWM	B 3/16, C 4/2
UNI-TELWAY	F 1/1	%QWxx	B 3/48, C 4/2
		%Qx.y	A 1/12

V		%Ri	B 2/20, C 4/2
		%SBRi	B 3/44, C 4/2
Valeur courante (bloc.V)	B 2/11	%SCi	B 3/46, C 4/2
Valeur de présélection (bloc.P)	B 2/11	%SWxx	B 6/7
Valeur immédiate	B 3/1	%Sxx	B 6/1
Variable		%Tmi	B 2/13
accès	C 9/2, G A/3	%Xi	B 2/26, C 4/2
non mentionnée sur clavier	C 4/2	-*-i	B 2/26, C 4/2
Version logicielle	C 2/1	=*-i	B 2/26, C 4/2
Visualisation		=*=POST	B 2/26, C 4/2
bits internes	A 1/20		
blocs fonctions	C 3/2, C 9/4		