



Généralités PL7-2

A

Le langage à contacts PL7-2

B

Le langage grafcet PL7-2

C

Aide à l'exploitation

D

Communication par prise terminal

E

Exemples d'application et annexes

F

G

H



Chapitre	Page
1 <i>Présentation du logiciel PL7-2</i>	1/1
1.1 Présentation des langages PL7-2	1/2
1.2 Structure du logiciel PL7-2	1/4
2 <i>Les objets adressables du logiciel PL7-2</i>	2/1
2.1 Adressage	2/2
2.2 Objets bits	2/7
2.3 Objets mots	2/9
2.4 Objets blocs fonction	2/12
2.5 Indexation des objets	2/14
3 <i>Structure mémoire utilisateur</i>	3/1
3.1 Généralités	3/2
3.2 Mémoire TSX 17-20	3/3
3.3 Mémoire TSX 27/47	3/5
4 <i>Structure logicielle monotâche</i>	4/1
4.1 Description tâche maître	4/2
4.2 Traitement du programme	4/4
4.3 Fonctionnement structure monotâche	4/5
4.4 Gestion des bases de temps	4/6
5 <i>Structure logicielle bitâche</i>	5/1
5.1 Description de la tâche rapide	5/2
5.2 Tâche rapide du TSX 17-20	5/3
5.3 Tâche rapide des TSX 27/47	5/5
5.4 Gestion des bases de temps	5/6



Chapitre	Page
6 <i>Rôle des bits et mots système</i>	6/1
6.1 Bits système	6/2
6.2 Mots système	6/8
7 <i>Coupures et reprises secteur</i>	7/1
7.1 Principe	7/2
7.2 Comportement sur reprise secteur structure monotâche	7/4
7.3 Comportement sur reprise secteur en structure bitâche	7/8
7.4 Programmation de la reprise secteur	7/10



Sous chapitre	Page
1.1 Présentation des langages PL7-2	1/2
1.1-1 Le langage à contacts	1/2
1.1-2 Le langage GRAFCET	1/2
1.1-3 Avantages des langages graphiques	1/3
1.1-4 Aide à l'exploitation	
1.2 Structure du logiciel PL7-2	1/4
1.2-1 Structure Monotâche	1/4
1.2-2 Structure Bitâche	1/4
Ce chapitre se termine à la page	1/6

1.1 Présentation des langages PL7-2

Les langages PL7-2 sont des langages graphiques destinés à programmer les automates TSX 17-20, TSX 27, TSX 47-J et TSX 47-10/20.

Les langages graphiques PL7-2 sont:

1.1-1 Le langage à contacts

Ce langage est adapté à la programmation des traitements logiques. Les programmes écrits en langage à contacts se composent d'une succession de réseaux de contacts, véhiculant des informations logiques.

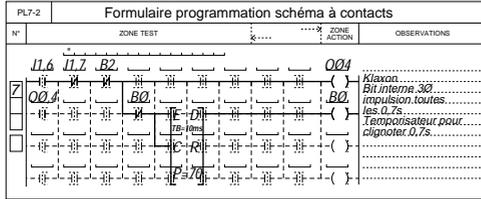
Les principaux éléments de ce langage sont:

 les contacts : qui permettent de LIRE (de tester) des niveaux logiques.

 les bobines : qui permettent d'ECRIRE (de définir) des niveaux logiques.

 les blocs fonctions: qui permettent de réaliser des fonctions d'automatismes pré-programmées.

 les blocs opérations : qui permettent d'effectuer des opérations arithmétiques, logiques, etc.

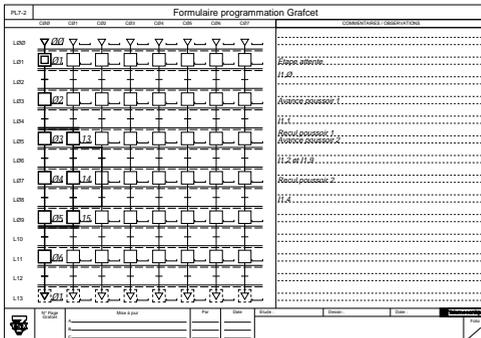


1.1-2 Le langage GRAFCET

Issu d'une méthode d'analyse (conforme à la norme GRAFCET), ce langage est conçu pour décrire des fonctionnements séquentiels. Il comprend:

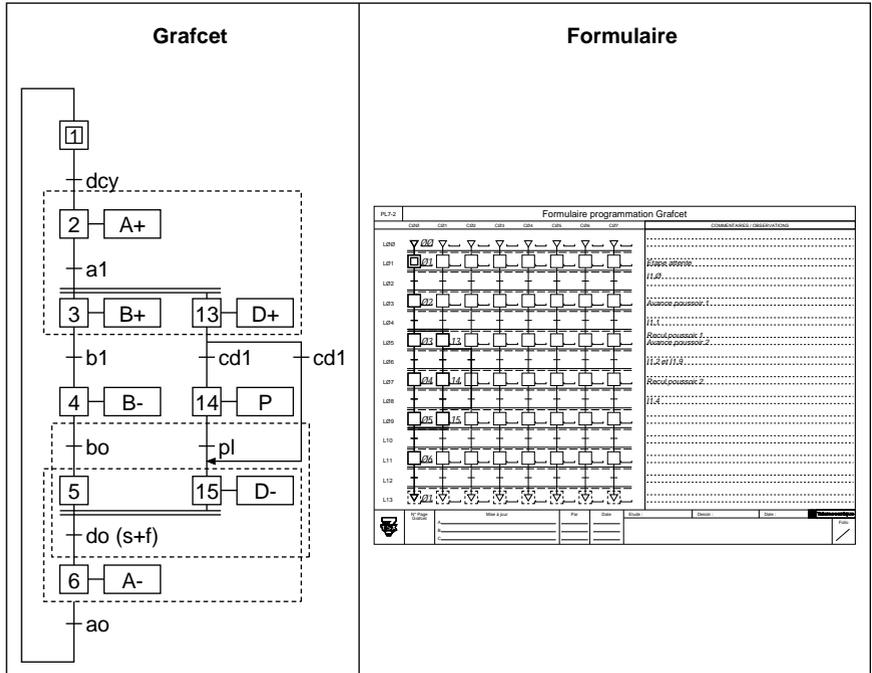
 le GRAFCET graphique qui se compose d'étapes, de transitions et de liaisons orientées définissant l'ossature du programme.

Des réseaux de contacts définissent les réceptivités associées aux transitions et les actions associées à chaque étape.



1.1-3 Avantages des langages graphiques

- Programmation aisée car ces langages sont proches de l'analyse : transcription directe de l'étude.
- Mise au point et maintenance facilitées grâce à la visualisation dynamique des réseaux de contacts et des étapes du GRAFCET.
- Documentation application reflétant l'étude et la saisie.



1.1-4 Aide à l'exploitation

Les langages PL7-2 proposent également une structure d'accueil (bits système, diagnostic...) permettant d'intégrer facilement et méthodiquement les modes de marches et d'arrêts de l'application.

1.2 Structure du logiciel PL7-2

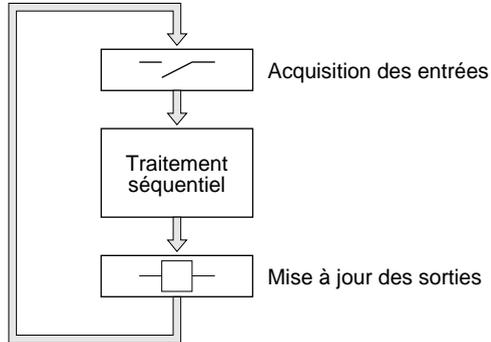
1.2-1 Structure Monotâche

La tâche exécutée de manière cyclique est appelée tâche maître.

Elle est surveillée par un chien de garde logiciel, fixé à 150 ms.

Le cycle automate dans le cas d'une structure monotâche est :

- Acquisition des entrées
- Scrutation du programme: scrutation des réseaux de contacts et des pages GRAFCET et exécution des instructions programmées
- Mise à jour des sorties



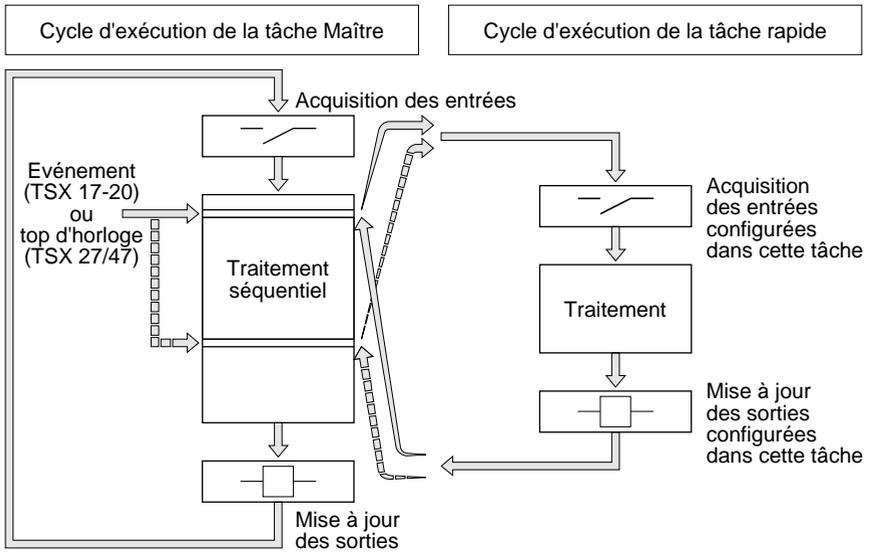
1.2-2 Structure Bitâche

La structure bitâche permet de gérer deux fonctions différentes auxquelles sont associées des programmes spécifiques: chaque programme est implanté dans une tâche. Ces tâches sont indépendantes et exécutées par le même processeur.

Ce type de structure a pour but de répartir la conception et de simplifier la mise au point. Chaque programme est écrit et mis au point dans la tâche correspondante, indépendamment de l'autre.

La structure bitâche des automates TSX 17-20, TSX 27/47 se présente comme suit :

- Tâche Maître : voir structure monotâche
- Tâche rapide : destinée aux traitements de courte durée.
 - Tâche rapide TSX 27/47 : tâche périodique, gérée par une base de temps (5 ou 10 ms).
 - Tâche rapide TSX 17-20 : tâche événementielle non périodique, activée soit par des événements externes, soit par des événements internes à l'automate. Cette tâche assure des temps de réponses courts et garantis (< 2 ms).

Structure Bitâche (suite)



Sous chapitre	Page
2.1 Adressage	2/2
2.1-1 Adressage des entrées/sorties du TSX 17-20	2/2
2.1-2 Adressage des entrées/sorties du TSX 27	2/4
2.1-3 Adressage des entrées/sorties des TSX 47-J, TSX 47-10/20	2/5
2.2 Objets bits	2/7
2.2-1 Généralités	2/7
2.2-2 Liste des objets bits	2/7
2.2-3 Définition des principaux objets bits	2/8
2.3 Objets mots	2/9
2.3-1 Généralités	2/9
2.3-2 Liste des objets mots et bits associés	2/10
2.3-3 Définition des différents objets mots	2/11
2.4 Objets blocs fonctions	2/12
2.4-1 Généralités	2/12
2.4-2 Liste des mots et bits des blocs fonctions	2/13
2.5 Indexation des objets	2/14
2.5-1 Adressage direct	2/14
2.5-2 Adressage indexé	2/14
Ce chapitre se termine à la page	2/14

2.1 Adressage

2.1-1 Adressage des entrées/sorties du TSX 17-20

Le TSX 17-20 est un automate compact et modulaire. Aux entrées/sorties de l'automate de base peuvent s'ajouter les entrées/sorties des blocs ou modules d'extensions.

L'adressage des entrées/sorties est géographique.

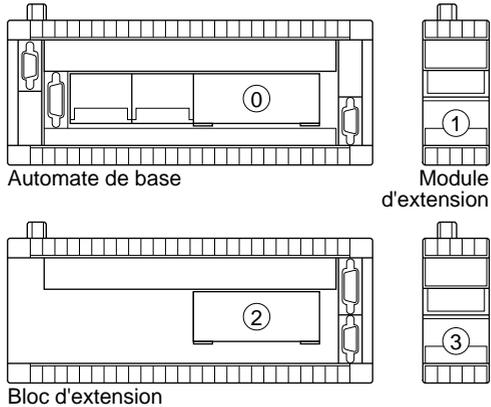
Ces entrées/sorties sont adressées dans le programme par les instructions :

I/Ox,i I: Entrée (Input) O: Sortie (Output)

x : numéro du module (0 à 3)

- ① automate de base
- ① première extension connectée
- ② deuxième extension connectée
- ③ troisième extension connectée

i : numéro de la voie dans le module.

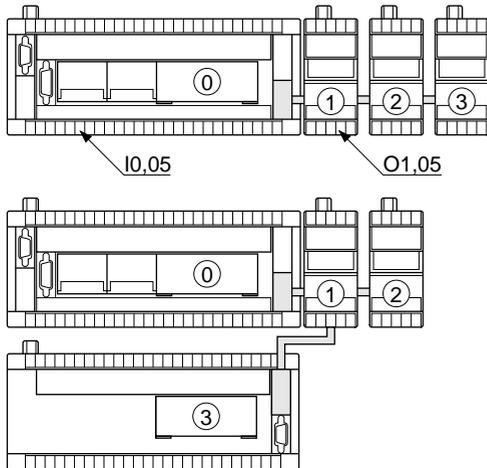


Nota : l'adressage des blocs ou modules d'extensions et des modules intelligents dépend de l'ordre de leur connexion.

Configuration maximum

Automate de base	Blocs d'extension E/S TOR	Modules d'extension E/S TOR ou/et modules intelligents
1	3	0
1	2	1
1	1	2
1	0	3(*)

Exemples d'associations



(*) modules intelligents TSX SCG 113 ou 116 :
2 modules maximum.
(voir manuel de mise en œuvre TSX D 11000F pour associations possibles.)

Adressage des entrées/sorties du TSX 17-20 (suite)

Types d'automates TSX 17-20

Type	Nombre d'E/S	Entrées	Sorties	Entrées événementielles
automates de base TSX 17-20	20 (12E+8S)	de I0,00 à I0,11	de O0,00 à O0,07	10,24 10,25
	34 (22E+12S)	de I0,00 à I0,21	de O0,00 à O0,11	10,24 10,25
	40 (24E+16S)	de I0,00 à I0,23	de O0,00 à O0,15	10,24 10,25
blocs d'extension pour automate de base TSX 17-20	34 (22E+12S)	de Ix,00 à Ix,21	de Ox,00 à Ox,11	–
	40 (24E+16S)	de Ix,00 à Ix,23	de Ox,00 à Ox,15	–
modules d'extension pour automate de base TSX 17-20	8 (8E)	de Ix,00 à Ix,07	–	–
	6 (6S)	–	de Ox,00 à Ox,05	–

Entrée I0,00 et sortie O0,00

Ces deux voies peuvent avoir chacune deux utilisations différentes:

Voie	Utilisation par défaut	Utilisation après configuration
Entrée I0,00	Entrée normale	Entrée de commande "RUN/STOP" automate
Sortie O0,00	Sortie normale	Sortie "SECU" automate

Adressage des mots registres d'entrées/sorties des modules intelligents

IWx,i : mot d'entrée

OWx,i : mot de sortie

x : numéro de module (1 à 3)

i : numéro du mot (0 à 7)

Nota : Les modules intelligents possèdent également des bits d'entrées/sorties dont l'adressage est identique à celui des extensions d'entrées/sorties T.O.R.

2.1-2 Adressage des entrées/sorties du TSX 27

Le TSX 27 est un automate compact qui possède 1, 2, 3 ou 4 cartes d'entrées/sorties. Chaque carte possède 20 entrées/sorties (12 entrées et 8 sorties).

L'adressage d'une entrée ou d'une sortie est défini dans le programme par 4 caractères:

I/Oy,i I: Entrée (Input) O: Sortie (Output)

y : emplacement (0 à 7)

, : virgule

i : numéro de voie dans la carte

0 à 7 pour les sorties

0 à B pour les entrées

Chaque carte d'entrées/sorties possède:

- en partie supérieure un ensemble de 8 sorties repérées par les emplacements 0, 2, 4 ou 6 dans l'automate, et par leurs numéros de voies de 0 à 7.

exemple: O4,5 = sixième voie de la carte de l'emplacement 4.

- en partie inférieure un ensemble de 12 entrées repérées par les emplacements 1, 3, 5 ou 7 dans l'automate, et par leurs numéros de voies de 0 à B.

exemple: I3,A = onzième voie de la carte de l'emplacement 3.

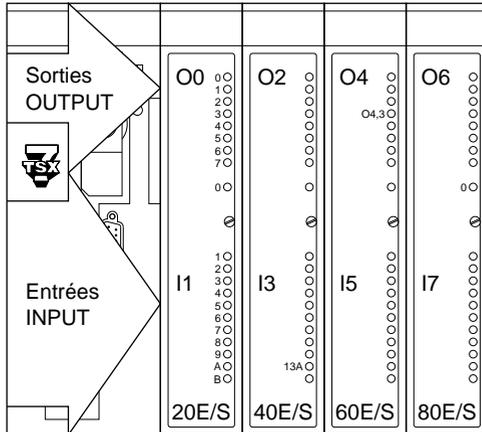


Tableau récapitulatif

Sorties (Output)	Oy,i	y = 0, 2, 4, 6 i = 0, 1, ..., 7
Entrées (Input)	Iy,i	y = 1, 3, 5, 7 i = 0, 1, ..., B

2.1-3 Adressage des entrées/sorties des TSX 47-J, TSX 47-10/20

Le TSX 47 est un automate modulaire qui peut gérer jusqu'à 16 modules d'entrées/sorties. Ces modules sont de 4, 8, 16, 24 ou 32 voies.

L'adressage d'une entrée ou d'une sortie est défini dans le programme par les caractères suivants:

I/O y , i I: Entrée (Input) O: Sortie (Output)

x : numéro du bac

0 : bac de base

1 : bac d'extension

y : emplacement du module dans le bac (0 à 7)

, : virgule

i : numéro de la voie dans le module (0 à F modules 4 à 16 voies)

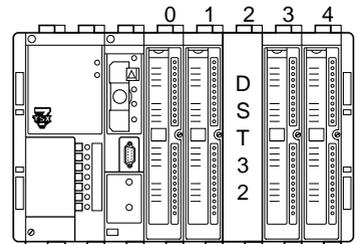
Cas des modules 24 ou 32 voies

Pour adresser des entrées/sorties d'un module 24 ou 32 voies, il faut procéder de la façon suivante:

- le module doit être obligatoirement dans le bac de base: $x = 0$
- les 16 premières voies sont adressées normalement: I/O y , i $i = 0$ à F
- les voies \geq à 16 sont adressées par : I/O1 y , i $i = 0$ à 7 modules 24 voies
 $i = 0$ à F modules 32 voies

Cas du TSX 47-J

Modules	4 voies	8 voies	16 voies	24 voies	32 voies
Adresses I/O	y,0 à y,3	y,0 à y,3	y,0 à y,F	0y,0 à 0y,F	0y,0 à 0y,F
				1y,0 à 1y,7	1y,0 à 1y,F



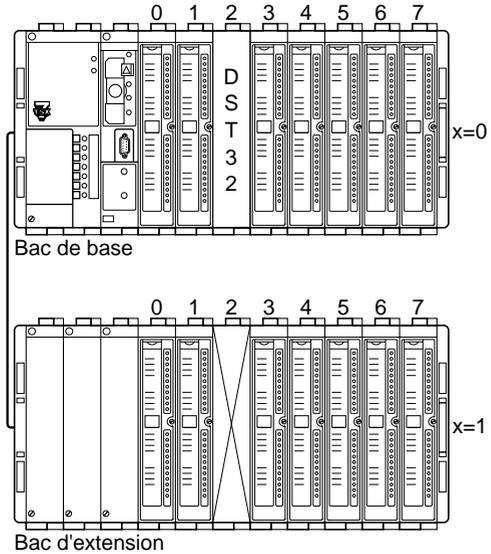
Bac de base

Le TSX 47-J peut gérer jusqu'à 5 modules dans le bac de base, il n'accepte pas de bac d'extension.

Cas du TSX 47-10/20

Les modules 24 ou 32 voies doivent être implantés dans le bac de base.

L'emplacement portant le même numéro dans le bac d'extension doit rester vide dans ce cas.



Modules	4 voies	8 voies	16 voies	24 voies	32 voies
Adresses I/O (x=0)	y,0 à y,3	y,0 à y,3	y,0 à y,F	0y,0 à 0y,F	0y,0 à 0y,F
Adresses I/O (x=1)	1y,0 à 1y,3	1y,0 à 1y,7	1y,0 à 1y,F		

Adressage des mots registres d'entrées/sorties des coupleurs intelligents pour automate TSX 47-20 :

IWy,i : mot d'entrée

OWy,i : mot de sortie

y : numéro d'emplacement du module (0 à 3*)

i : numéro du mot (0 à 7)

Nota : Les coupleurs possèdent également des bits d'entrées/sorties dont l'adressage est identique à celui des modules d'entrées/sorties T.O.R. d'entrées/sorties T.O.R.

* Les coupleurs intelligents ne peuvent être implantés que dans les 4 premiers emplacements du bac 0. Le nombre de coupleurs est limité à 2.

2.2 Objets bits

2.2-1 Généralités

Les langages accèdent au bit et au bit extrait de mot soit en lecture, soit en lecture/écriture.

2.2-2 Liste des objets bits

Les différentes notations utilisées sont : R pour lecture, W pour écriture, F pour forçage.

Bits/Nature	Adressage			Possibilités
	TSX 17-20	TSX 27	TSX 47	
Bits d'entrées/ sorties T.O.R.	Ix,i (96) Ox,i (64)	ly,i (48) Oy,i (32)	Ixy,i Oxy,i (256)*	R/F R/W/F
Bits entrées Événementielles	I0,24 I0,25			R R
Bits internes	Bi (256)	Bi (256)	Bi (256)	R/W/F
Bits système	SYi (24)	SYi (24)	SYi (24)	R/W selon i
Bits d'étapes GRAF CET	Xi (96)	Xi (96)	Xi (96)	R/W
Bits de blocs Fonctions	Ti,D; Di,F; ... (voir objets blocs fonctions)			
Bits extraits de mots	Wi,j; ... (voir page 2/10)			
Bits défauts	Sx,i (4x8)	Oy,S (4)	Ixy,S (16) Oxy,S (16)	R R

Remarque : Lecture/écriture : s'effectue par programme ou par terminal.

Forçage :

entrées/sorties : le forçage est prioritaire sur l'acquisition des entrées et sur le positionnement des sorties défini par programme, et ce jusqu'à sa suppression.

bits internes : le forçage est prioritaire sur l'état défini par le programme utilisateur.

* 256 est le nombre maximum d'entrées et de sorties.

2.2-3 Définition des principaux objets bits

- **Bits d'entrées/sorties**

Ces bits sont les "images logiques" des états électriques des entrées/sorties des modules. Ils sont définis dans la mémoire de données et sont mis à jour à chaque tour de cycle.

- **Bits défauts des modules (bits Status)**

Chaque module dispose d'un bit adressable, de rang S, qui est positionné par le système à 1 lors de l'apparition d'un défaut sur le module concerné :

TSX 17-20 : il possède 8 bits de Status pour chaque module: $S_{i,j}$ ($i = 0$ à 3 et $j = 0$ à 7).

TSX 27/47 : $O_{y,S}$; $I_{x,S}$; $O_{xy,S}$ signalent un défaut sur le module correspondant lorsqu'ils sont à 1.

Les définitions des bits status sont données dans "Aide à l'exploitation", chapitre 3 : Aide au diagnostic, intercalaire D.

- **Bits internes**

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

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

- **Bits système**

Les bits système SY0 à SY23 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 6 du présent intercalaire.

2.3-2 Liste des objets mots et bits associés

Les différentes notations utilisées sont : R pour lecture et W pour écriture

Mots/Nature	Adressage			Possibilités
	TSX 17-20	TSX 27	TSX 47	
Mots internes	Wi (1024)	Wi (128)	Wi (1024)	R/W ⁽¹⁾
Mots constants	CWi (1024)	CWi (1024)	CWi (1024)	R ⁽¹⁾
Mots système	SWi (64)		SWi (4)	R/W selon i
Mots communs réseau	COMi,j(16x4) FIPWAY		COMi,j (16x4) TELWAY	R/W ⁽²⁾
Mots de blocs Fonctions	Mi,P; Di,S; ... (voir objets blocs fonctions)			
Temps d'activité d'étapes GRAFCET	Xi,V (96)	Xi,V (96)	Xi,V (96)	R
Mots registres d'entrées/sorties	IWx,i (3x8) OWx,i (3x8)		IWxy,i (8x4) OWxy,i (8x4)	R R/W

Bits extraits de mots/nature	Adressage			Possibilités
	TSX 17-20	TSX 27	TSX 47	
Bit j du mot interne i	Wi,j(128x16)	Wi,j(128x16)	Wi,j(128x16)	R/W
Bit j du mot constant i	CWi,j(128x16)	CWi,j(128x16)	CWi,j(128x16)	R
Bit j du mot système i	SWi,j(64x16)		SWi,j(4x16)	R/W
Bits k du mot commun j de la station i	COMi,j,k(64x16)		COMi,j,k(64x16)	R/W ⁽²⁾
Bit j du mot registre d'E/S i	IWx,i,j (24x16) OWx,i,j (24x16)		IWxy,i,j (32x16) OWxy,i,j (32x16)	R/W ⁽³⁾ R/W ⁽³⁾

Remarque : A partir du rang 10 et jusqu'au rang 15, le bit j extrait de mot est adressé, en hexadécimal, par les lettres A (pour le rang 10) à F (pour le rang 15, voir page 2/9). Exemple : W100,B.

⁽¹⁾ accès direct aux 128 premiers mots. Par transfert de table pour les mots suivants,

⁽²⁾ écriture par le programme de la station i, et lecture de tous les bits,

⁽³⁾ écriture uniquement par le programme.

2.3-3 Définition des différents objets mots

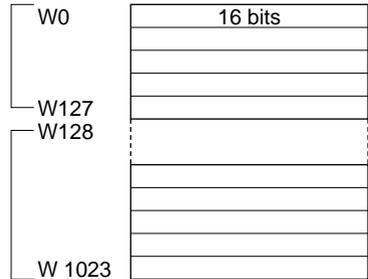
• Mots internes

Les mots internes sont destinés à stocker des valeurs en cours de programme

Ils sont rangés dans la mémoire de données.

Les mots W0 à W127 sont accessibles directement par programme (en lecture/écriture). Ils sont utilisés comme mots de travail.

Les mots W128 à W1023 sont dits mots de stockage. Ils ne sont accessibles que par transfert de tables de mots⁽¹⁾.

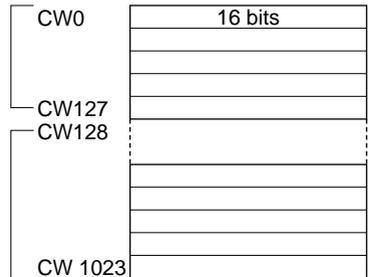


• Mots constants

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. Le nombre de CWi en exploitation, doit être défini par CONFIGURATION.

Les mots constants CW0 à CW127 sont accessibles directement par programme (en lecture uniquement).

Les mots CW128 à CW1023 ne sont pas accessibles directement, ils doivent être transférés par tables dans des mots internes⁽¹⁾.



• Mots registres d'entrées/sorties

Les mots registres sont des objets mots de 16 bits associés aux coupleurs intelligents.

Chacun de ces coupleurs dispose de 8 mots registres d'entrées IW et de 8 mots registres de sorties OW.

• Mots communs

Les mots communs sont des objets mots de 16 bits communs à toutes les stations connectées au réseau de communication FIPWAY ou TELWAY. Ils constituent dans chaque automate une table de 64 mots pour un réseau composé de 16 stations.

• Mots système

Ces mots de 16 bits assurent plusieurs fonctions :

- certains renseignent sur l'état du système par lecture des mots SWi (temps de fonctionnement système et application, etc...).
- d'autres permettent d'agir sur l'application (gestion réseau, etc...).

Les mots système sont décrits sous-chapitre 6.2 du présent intercalaire.

⁽¹⁾ voir blocs opération transferts chapitre 5, intercalaire B.

• Mots associés au Grafcet

Ces mots représentent le temps d'activité des étapes Grafcet. La valeur maximale du temps est de 9999 incréments, la base de temps étant de 1s.

• Objets bits extraits de mots

Le logiciel PL7-2 permet d'extraire l'un des 16 bits de certains mots.

Le repère du mot est alors complété par le rang du bit extrait séparé par une virgule.

2.4 Objets blocs fonctions

2.4.1 Généralités

Le langage à contacts utilise une ressource appelée : blocs fonctions, lesquels se répartissent en deux groupes :

- les blocs fonctions d'automatismes : ils mettent à disposition de l'utilisateur des fonctions pré-programmées telles que temporisateurs, monostables, compteurs, registres, etc...
- le bloc fonction texte : il permet d'échanger des données entre programme utilisateur et coupleur intelligent, ou prise terminal, ou système, ou autre programme utilisateur. Ces communications peuvent s'effectuer localement ou à travers le réseau FIPWAY, TELWAY ou le bus UNI-TELWAY.

Les autres types de blocs, comparaison et opérations portent sur les mots définis précédemment et ci-après.

2.4-2 Liste des mots et bits des blocs fonctions

Les blocs fonctions mettent en oeuvre des objets bits de sortie et mots spécifiques. Les notations suivantes sont utilisées : R pour lecture, W pour écriture.

Blocs fonction (Nb)	Mots et bits associés	adresses	possibilités	
Temporisateur Ti TSX 17-20 (32) TSX 27/47 (16)	mot	valeur de présélection	Ti,P	R/W ⁽¹⁾
		valeur courante	Ti,V	R
	bit	temporisateur en cours	Ti,R	R
		temporisateur écoulé	Ti,D	R
Monostable Mi TSX 17-20 (8) TSX 27/47 (8)	mot	valeur de présélection	Mi,P	R/W ⁽¹⁾
		valeur courante	Mi,V	R
	bit	monostable en cours	Mi,R	R

⁽¹⁾ Si "YES" en mode CONFIGURATION, modification possible en mode REGLAGE.

Blocs fonction (Nb)	Mots et bits associés	adresses	possibilités
Compteur/décompteur Ci TSX 17-20 (31) TSX 27/47 (16)	mot	valeur de présélection	Ci,P R/W ⁽¹⁾
		valeur courante	Ci,V R
	bit	débordement comptage	Ci,E R
		présélection atteinte	Ci,D R
Registre Ri TSX 17-20 (4) TSX 27/47 (4)	mot	mot d'entrée	Ri,I R/W
		mot de sortie	Ri,O R/W
	bit	registre plein	Ri,F R
		registre vide	Ri,E R
Texte TXTi TSX 17-20 (8) TSX 27/47 (8)	mot	longueur de table en octets	TXTi,L W
		status	TXTi,S R
		adresse coupleur et n° voie	TXTi,M W
		code requête	TXTi,C W
		adresse station FIPWAY/TELWAY	TXTi,A W
		n° bloc texte en communic.	TXTi,T W
		compte rendu d'échange	TXTi,R W
	bit	échange terminé	TXTi,D R
		échange erroné	TXTi,E R
Programmeur cyclique Di TSX 17-20 (8) TSX 27/47 (8)	mot	numéro du pas actif	Di,S R/W
		tps. activité pas courant	Di,V R
		16 bits d'ordres	Di,Wj R/W ⁽²⁾
	bit	dernier pas en cours	Di,F R
Compteur/ Temporisateur rapide FC TSX 17-20 (1)	mot	valeur de présélection	FC,P R/W ⁽¹⁾
		valeur courante	FC,V R/W
	bit	remise à 0 externe	FC,E R
		présélection atteinte	FC,D R
Horodateur H TSX 17-20 (illimité)	mot	Type : "WEEK" ou "YEAR" sélection jours "LMMJVSD"	VD W
		consigne de début	BGN W
		consigne de fin	END W
	bit	val. courante < consigne	< R
		val. courante = consigne	= R
		val. courante > consigne	> R

Ces mots et bits ne sont pas indexables.

⁽¹⁾ Si "YES" en mode CONFIGURATION, modification possible en mode REGLAGE,

⁽²⁾ Ecrit uniquement par configuration du bloc Di.

2.5 Indexation des objets

2.5-1 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 : B126 (bit interne d'adresse 126)

2.5-2 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 W_i . Le nombre de "mots index" n'est pas limité.

Exemple : $W108(W2)$: mot d'adresse direct 108 + contenu du mot $W2$.

Si le mot $W2$ a pour contenu la valeur 12, écrire $W108(W2)$ équivaut donc à écrire $W120$.

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.

Le contenu du mot utilisé comme index ne doit jamais être négatif. Le contenu du mot index ne doit jamais dépasser la valeur $127^{(1)}$, diminuée de l'adresse du mot indexé (mot interne ou mot constant).

Exemple : $W80(W100)$: le contenu de $W100$ ne doit pas dépasser : $127 - 80 = 47$. Les objets bits et mots indexables sont présentés dans les sous chapitres 2.2-2 et 2.3-2 du présent intercalaire.

Débordement d'index, bit système SY20

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 $127^{(1)}$.

En cas de débordement d'index, le système provoque la mise à l'état 1 du bit système SY20.

La surveillance du débordement est à la charge de l'utilisateur : le bit SY20 doit être lu par le programme utilisateur pour traitement éventuel. Sa remise à zéro est à la charge de l'utilisateur.

SY20 (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.

⁽¹⁾ 1023 pour TSX 17-20 VL > 2, si SY2=1 pour l'indexation des mots internes W_i .



Sous chapitre	Page
3.1 Généralités	3/2
3.1-1 Mémoire RAM on board	3/2
3.1-2 Mémoires cartouches utilisateur	3/2
3.2 Mémoires TSX 17-20	3/3
3.2-1 RAM on board TSX 17-20	3/3
3.2-2 Cartouches mémoire pour TSX 17-20	3/3
3.2-3 Fonctionnement de la cartouche EEPROM du TSX 17-20	3/4
3.2-4 Modes de Marches de la cartouche EEPROM à la mise sous tension	3/4
3.3 Mémoires TSX 27/47	3/5
3.3-1 RAM on board TSX 27/47	3/5
3.3-2 Cartouches mémoire pour TSX 27/47	3/5
Ce chapitre se termine à la page	3/6

3.1 Généralités

L'espace mémoire des automates TSX 17-20/27/47, accessible à l'utilisateur est une mémoire RAM CMOS (Random Access Memory ou mémoire vive). Cet espace, est découpé en deux parties distinctes :

- la mémoire programme, intégrée dans l'automate ou sur cartouche utilisateur (suivant automate), contient les instructions de programme et les données constantes définies par configuration (CWi, etc).
- la mémoire de données, intégrée au module processeur (RAM on Board), est la mémoire de travail dans laquelle sont stockés les résultats intermédiaires de calculs et les états divers (états logiques de bits internes : Bi, mots internes : Wi, etc).

3.1-1 Mémoire RAM on board

L'espace RAM on board est une mémoire intégrée au module processeur. Elle doit être maintenue sous tension pour conserver son contenu (tension secteur, pile , ...).

3.1-2 Mémoires cartouches utilisateur

La cartouche mémoire permet de stocker le programme application et les données constantes uniquement, en dehors de l'espace mémoire automate (unité centrale).

En programmation avec le terminal connecté, ou en exploitation, cette cartouche doit être insérée dans son emplacement réservé.

Ces cartouches mémoire peuvent être du type :

- RAM CMOS (Random Access Memory) : mémoire vive secourue par batterie Cadmium-nickel ou par pile (TSX 17-20),
- EPROM (Erasable Programmable Read Only Memory) : mémoire morte effaçable uniquement par exposition aux rayons ultra-violet,
- EEPROM (Erasable Electrical Programmable Read Only Memory) : mémoire morte effaçable électriquement par terminal.

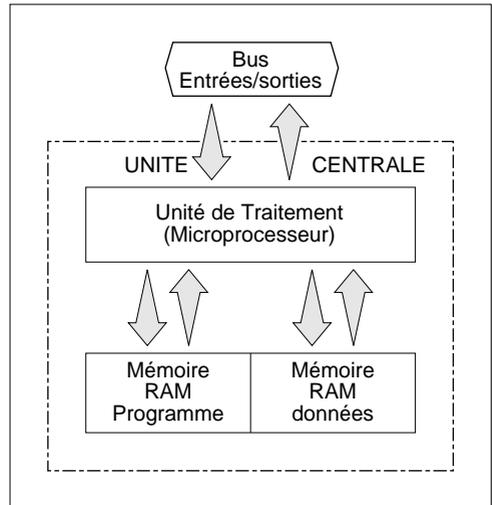
3.2 Mémoires TSX 17-20

3.2-1 RAM on board TSX 17-20

La RAM du TSX 17-20 est définie comme la mémoire de travail dans laquelle le processeur vient chercher ses instructions et ses données. C'est une mémoire de 32 K-octets, intégrée dans l'unité centrale. Elle se compose :

- d'une zone mémoire programme configurable : (8 ou 24 K-octets),
- d'une zone mémoire de données de 8 K-octets (dont une partie est réservée au système).

Cette mémoire peut être sauvegardée lorsque l'automate est hors tension, grâce à une pile lithium (un condensateur assure une sauvegarde minimum d'une heure).



3.2-2 Cartouches mémoire pour TSX 17-20

Le programme sauvegardé dans la mémoire cartouche doit être transféré dans la RAM programme pour pouvoir être exécuté. En effet, le processeur du TSX 17-20 ne peut travailler qu'avec cette dernière.

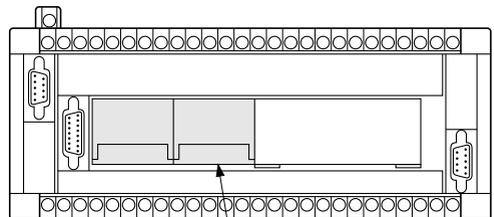
Deux types de cartouches sont destinés au TSX 17-20 :

- EPROM : 8/24 K-octets référencées respectivement TSX MC70 E28/E224.

A chaque mise sous tension et à condition que leurs contenus soient différents, un transfert EPROM vers RAM est automatiquement exécuté.

- EEPROM : 8/24 K-octets référencées respectivement TSX MC70 E38/E324.

Les cartouches EEPROM possèdent deux modes de fonctionnements définis par la position d'un commutateur placé à l'intérieur de celles-ci.



Cartouches EPROM et EEPROM

Attention : Il est impératif de retirer ou d'enficher les cartouches mémoire uniquement lorsque le TSX 17-20 est hors tension.

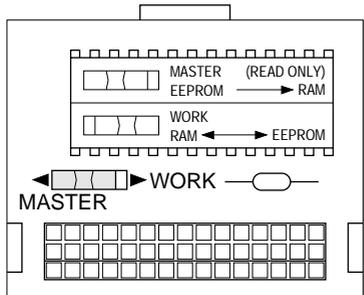
3.2-3 Fonctionnement de la cartouche EEPROM du TSX 17-20

La mémoire de sauvegarde EEPROM possède deux modes de fonctionnement, définis par la position du commutateur placé à l'intérieur de la cartouche :

Commutateur en position travail : WORK

Les transferts de programme entre la mémoire interne RAM et la cartouche EEPROM sont autorisés dans les deux sens, et exécutés à partir du terminal. Ce mode de fonctionnement permet entre autres :

- de sauvegarder un programme après l'avoir écrit,
- de rappeler un programme sauvegardé, le modifier, puis le sauvegarder à nouveau (cas d'un programme non protégé⁽¹⁾).



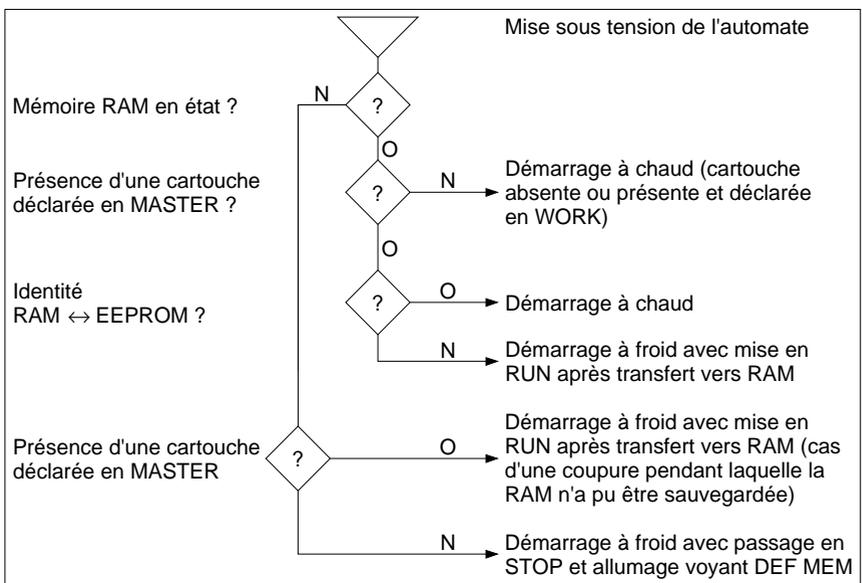
A la mise sous tension, le transfert EEPROM vers RAM n'est pas automatique.

Commutateur en position sauvegarde : MASTER

Les transferts de programme ne sont plus autorisés que dans le sens mémoire EEPROM vers mémoire RAM.

A chaque reprise secteur et si le contenu de la RAM automate est différent de celui de l'EEPROM, le contenu de celle-ci est transféré automatiquement en mémoire RAM automate.

3.2-4 Modes de Marches de la cartouche EEPROM à la mise sous tension



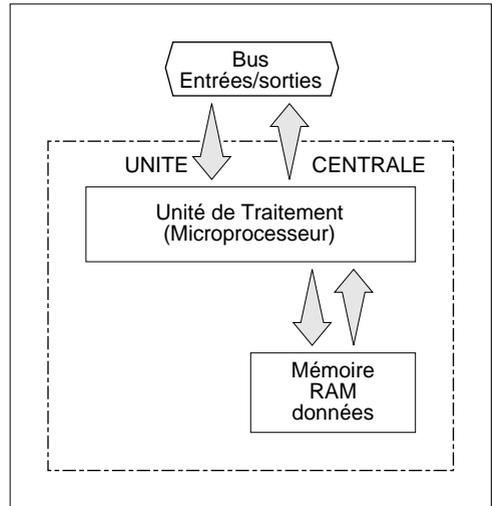
⁽¹⁾ La protection d'un programme consiste, lors de sa sauvegarde, à le rendre inaccessible à la lecture et à la modification (voir "Aide à l'exploitation" - chapitre 4 - intercalaire D : Fonctions spécifiques au TSX 17-20).

3.3 Mémoires TSX 27/47

3.3-1 RAM on board TSX 27/47

La RAM on board des automates TSX 27/47 contient uniquement une zone mémoire de données de 8 K-octets (dont une partie est réservée au système). La mémoire programme n'est pas intégrée dans l'unité centrale.

Cette mémoire RAM on board est secourue par une batterie Cadmium-nickel lorsque l'automate est hors tension.

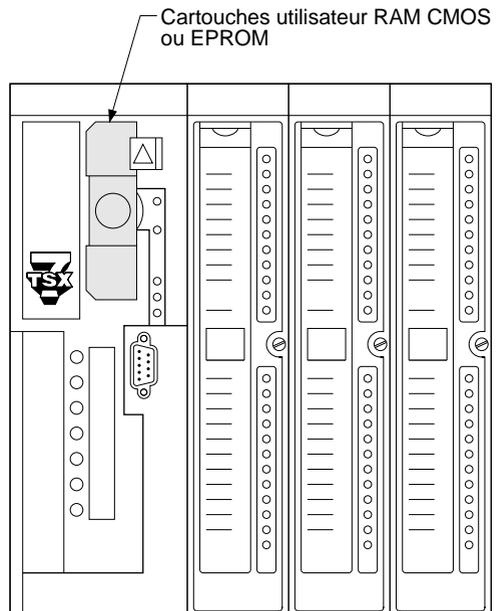


3.3-2 Cartouches mémoire pour TSX 27/47

Le processeur vient obligatoirement chercher ses instructions de programme dans la mémoire cartouche.

Deux types de cartouches sont disponibles pour les automates TSX 27/47 :

- RAM CMOS secourue par batterie : 8/16/32 K-octets, référencées respectivement TSX RAM 8 8/16 8/32 8. La durée de la sauvegarde est de 6 semaines en dehors de l'automate. **Ne pas omettre de charger cette batterie en insérant préalablement la cartouche dans un automate sous tension pendant 24 heures.**
- EPROM 8/16/32 K-octets référencées respectivement TSX RPM 8 8/16 8/32 8.



Attention: Il est déconseillé de débriquer sous tension la cartouche utilisateur insérée dans l'automate. Toutefois le débriquer accidentel de la cartouche utilisateur sous tension ne provoque aucune détérioration de l'automate mais a nécessairement des conséquences sur l'application.



Sous chapitre	Page
4.1 Description tâche maître	4/2
4.2 Traitement du programme	4/4
4.3 Fonctionnement structure monotâche	4/5
4.3-1 Cycle de fonctionnement	4/5
4.3-2 Temps de réponse automate	4/5
4.3-3 Débordement de tâche	4/6
4.4 Gestion des bases de temps	4/6
Ce chapitre se termine à la page	4/6

4.1 Description tâche maître

Le programme d'une application monotâche est contenu dans une seule tâche utilisateur: la tâche maître.

L'exécution de la tâche maître est effectuée de manière cyclique, selon un temps fonction de la taille du programme⁽¹⁾. Durant ce cycle, sont réalisées les fonctions suivantes :

Gestion système :

Le système assure implicitement :

- Prise en compte des bits et mots système et leur mise à jour.
- Surveillance de l'automate:
 - contrôle de l'exécutabilité de la mémoire programme,
 - gestion du temps (mise à jour des valeurs courantes de l'horodateur et actualisation des mots SW50 à SW53 (sur TSX 17-20),
 - surveillance des modules d'entrées/sorties (bits de défauts, configuration modules, bits système SY10 et SY16, etc),
 - mise à jour des voyants: RUN, I/O, MEM et PROG (sur TSX 17-20),
 - détection des passages RUN/STOP,
 - surveillance d'autres paramètres système.
- Traitement des requêtes du terminal.
- Routage de la messagerie (messagerie transparente, blocs textes, etc) :
 - analyse des "buffers" d'entrées, contenant les messages reçus,
 - identification des destinataires (coupleurs, réseau, etc),
 - transfert des messages dans les "buffers" de sorties, associés aux destinataires.

Acquisition des entrées :

- Prise en compte en mémoire de données des messages, en provenance des coupleurs intelligents⁽³⁾.
- Ecriture en mémoire de données des messages FIPWAY/TELWAY⁽²⁾ et mots communs (COMi,j).
- Ecriture en mémoire de données des mots registres d'entrées (IW) des coupleurs⁽³⁾.
- Ecriture en mémoire des bits d'entrées (I) des modules d'entrées.

Traitement du programme

Le programme écrit par l'utilisateur est alors exécuté⁽¹⁾.

Mise à jour des sorties

- Ecriture des bits de sorties (O) dans les modules de sorties.
- Ecriture des mots registres de sorties (OW) dans les coupleurs⁽³⁾.
- Emission vers le coupleur FIPWAY/TELWAY des messages⁽²⁾ et des mots communs (COMi,j).
- Emission des messages vers les coupleurs intelligents⁽³⁾.

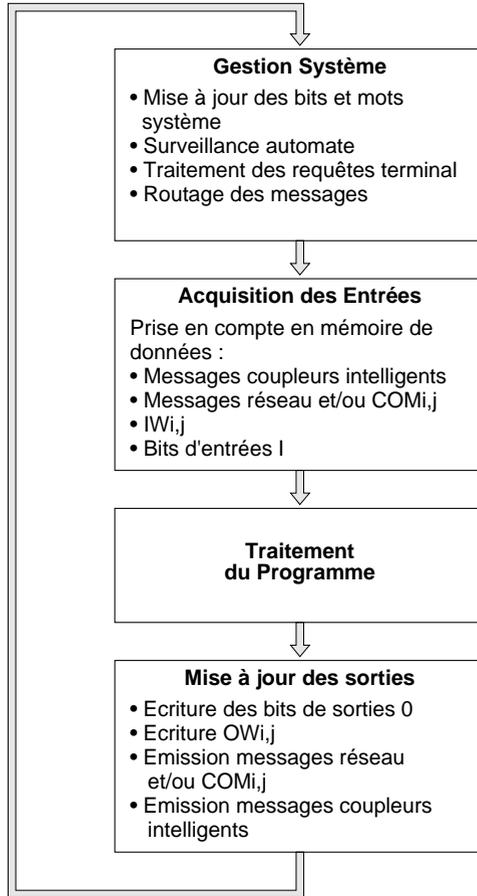
⁽¹⁾ un chien de garde logiciel déclenche un défaut d'exécution programme si le temps d'exécution du programme dépasse 150 ms (voir page 4/6).

⁽²⁾ Coupleur FIPWAY sur TSX 17-20 ou TELWAY sur TSX 47: à chaque cycle de la tâche maître, un message en réception et un en émission peuvent être traités à la fois.

⁽³⁾ A chaque cycle de la tâche maître, sont traités alternativement:

- l'acquisition et la mise à jour des mots de registres IW et OW,
- l'émission et la réception de messages par bloc texte pour coupleurs intelligents.

Synoptique du cycle de la tâche maître

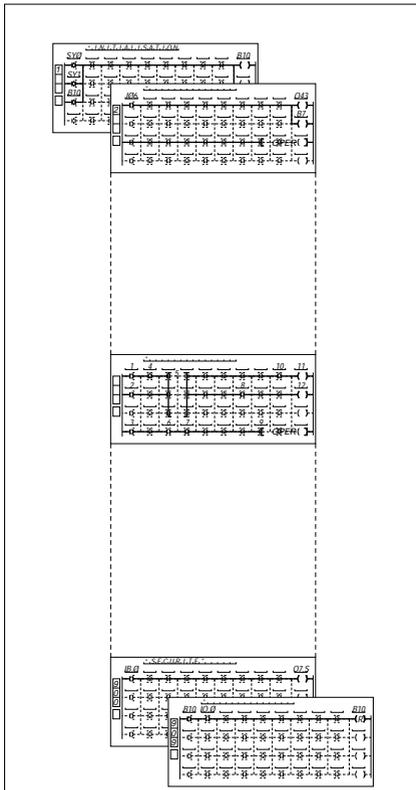


4.2 Traitement du programme

Suivant le langage utilisé, le programme de la tâche maître sera structuré en modules distincts :

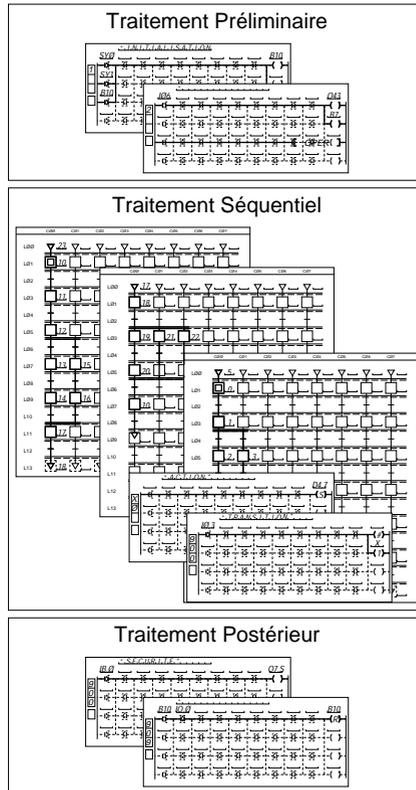
Langage à contacts

Module programme :
LADDER (LAD)



Langage Grafcet

Module programme :
Préliminaire (PRE)
Séquentiel (SEQ)
Postérieur (POS)



- Modules programme LADDER, Préliminaire, Postérieur :**
 Ces modules sont écrits à l'aide du langage à contacts. Ils sont composés de réseaux de contacts successifs, repérés par des étiquettes (1 à 999).
 Les modules programmes préliminaire et postérieur sont associés au module programme séquentiel dans le langage GRAFCET.
- Module programme Séquentiel :**
 Il est réservé aux pages GRAFCET pour la saisie du graphe. Il fait appel pour l'écriture de chaque réceptivité ou action au langage à contacts.

4.3 Fonctionnement structure monotâche

4.3-1 Cycle de fonctionnement

Lorsque l'automate est en RUN, le cycle de fonctionnement est celui décrit en 4.1. Lorsque l'automate est en STOP, le cycle de fonctionnement comprend dans tous les cas la gestion du système. Suivant la nature de l'automate, il se décompose comme suit :

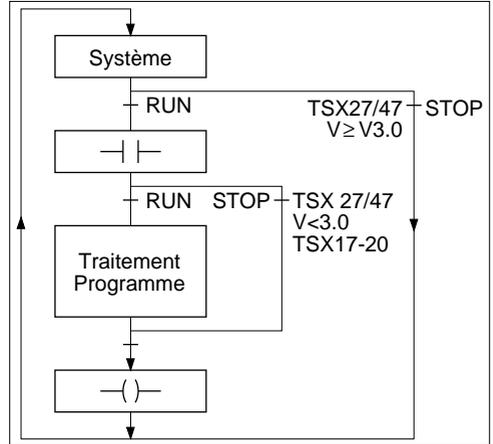
Cycle en STOP des TSX 27/47

$V \geq 3$:

Le processeur effectue uniquement la gestion du système. Il n'y donc pas de mise à jour de la mémoire de données.

Cycle en STOP des TSX 17-20, 27/47 $V < 3$:

Le processeur effectue la gestion du système, l'acquisition des entrées et la mise à jour des sorties décrites à la page 4/2.

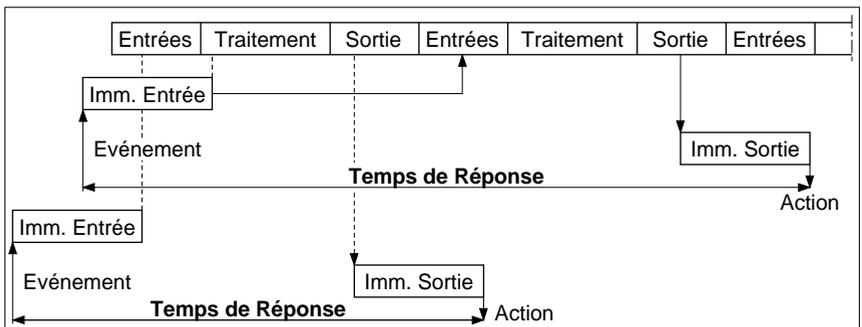


4.3-2 Temps de réponse automate

Le temps de réponse automate (TR) est le temps qui sépare l'arrivée d'une information sur le module d'entrée et la mise à jour de la sortie correspondante. Ce temps dépend du temps d'immunité des entrées/sorties (TIE/TIS, environ 10 ms pour une prise en compte), et du temps de cycle automate (TC).

Le temps de réponse de la tâche maître est aléatoire. Il est égal à :

$$TR = TIE + (1 \text{ à } 2)TC + TIS$$



4.3-3 Débordement de tâche

L'exécution du programme utilisateur est contrôlée par un chien de garde logiciel, fixé à 150 ms. Si son exécution dépasse 150ms (ex. : boucle dans le programme), le débordement tâche entraîne un défaut logiciel bloquant soit :

- arrêt immédiat de l'automate,
- voyants STOP et PROG clignotants sur TSX 17-20,
- voyant STOP clignotant et voyant MEM fixe sur TSX 27/47.

La relance de l'automate nécessite une reprise secteur du type reprise à froid.

Les phases : gestion système, acquisition des entrées et mise à jour des sorties ne sont pas contrôlées par le chien de garde.

4.4 Gestion des bases de temps

Les bases de temps des automates TSX 17-20/27/47 sont gérées par une horloge interne qui fonctionne de manière asynchrone par rapport au temps de cycle.

Les bits et mots système associés à la gestion du temps (SY5, SY6, SY7, les bits et valeurs courantes de temporisateurs et monostables, ainsi que les temps d'activité d'étapes GRAFCET et les temps enveloppes de programmeurs cycliques), sont rafraîchis sur interruption système, indépendamment de la position dans le cycle, et ceci à chaque unité de leur base de temps respective.



Sous chapitre	Pages
5.1 Description de la tâche rapide	5/2
5.2 Tâche rapide du TSX 17-20	5/3
5.2-1 Entrées événementielles	5/4
5.2-2 Entrées/sorties affectées à la tâche rapide	5/4
5.2-3 Temps de réponse de la tâche rapide	5/4
5.3 Tâche rapide des TSX 27/47	5/5
5.3-1 Bases de temps	5/6
5.3-2 Entrées/sorties affectées à la tâche rapide	5/6
5.4 Gestion des bases de temps	5/6
Ce chapitre se termine à la page	5/6

5.1 Description de la tâche rapide

Pour répondre au besoin de prise en compte d'événements rapides et temporaires, les automates TSX 17-20, TSX 27/47 V \geq 3 intègrent en plus de la tâche maître, une tâche rapide :

- événementielle pour le TSX 17-20,
- périodique pour les TSX 27/47.

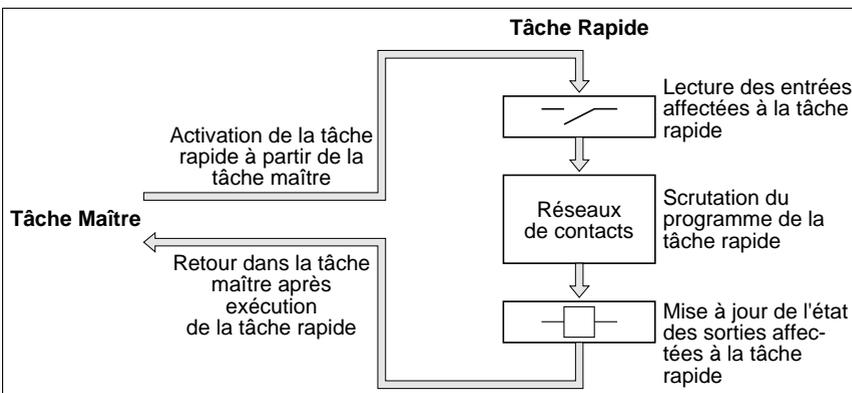
Cette tâche prioritaire est réservée aux traitements de courte durée, nécessitant une rapidité d'exécution, incompatible avec les temps d'exécution de la tâche maître (voir temps de réponse page 4/5 du présent intercalaire).

La tâche rapide ne peut être programmée qu'avec le langage à contacts (structure LADDER).

Le programme de la tâche rapide ne doit pas excéder 20 réseaux de contacts (étiquettes admises: 1 à 20). Son temps de cycle doit être inférieur ou égal à 2 ms.

Le bit système SY19 permet à l'utilisateur de valider ou d'inhiber cette tâche, en le positionnant respectivement à 0 ou 1. Le positionnement de ce bit doit être fait en tâche maître.

Une fois cette tâche activée, le processeur lit l'état des entrées qui lui sont affectées, exécute son programme, puis met à jour les sorties qui lui sont aussi affectées, avant même de revenir dans la tâche maître.



5.2 Tâche rapide du TSX 17-20

La tâche rapide du TSX 17-20 offre des temps de réponse courts et garantis, pour des traitements de courte durée.

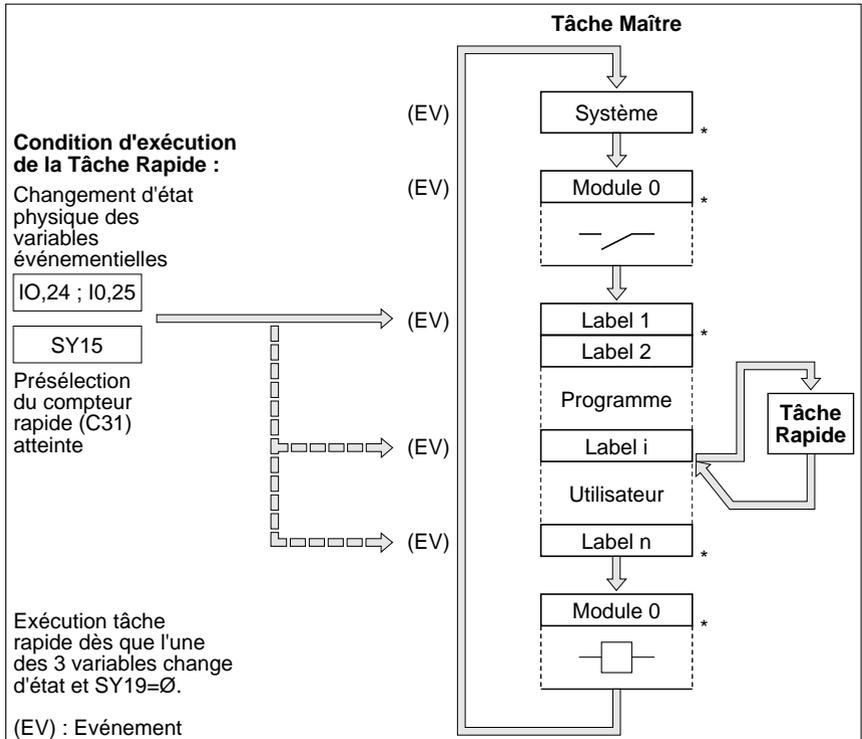
Principe

Si SY19 = 0, cette tâche est activée, après interruption de la tâche maître, dès l'apparition de l'un des événements suivants :

- changement d'état physique : front montant ou front descendant sur l'une des entrées dédiées I0,24 ou I0,25 (appelées aussi entrées événementielles).
- présélection atteinte : arrivée du compteur rapide FC à sa valeur de présélection (SY15 = 1). Une fois à 1, ce bit système doit être remis à 0 par programme, dans la tâche rapide.

Si SY19 = 0 et si un événement est présent, l'interruption de la tâche maître et l'activation de la tâche rapide ne peuvent être effectuées que durant les phases suivantes :

- gestion du système,
- en fin de scrutation d'un module d'entrées/sorties (après lecture ou écriture complète),
- exécution du programme utilisateur:
 - après scrutation d'un réseau complet,
 - ou exécution d'une action (Grafcet),
 - ou évaluation d'une réceptivité (Grafcet).



Important : Si un événement apparaît durant l'exécution de la tâche rapide, celle-ci termine son cycle et en relance un immédiatement, sans retour dans la tâche maître. Dans ce cas attention au déclenchement du chien de garde.

5.2-1 Entrées événementielles

Les entrées I0,24 et I0,25 sont des entrées 24VCC ayant un temps d'immunité réduit (<1 ms). Elles sont :

- **Soient dédiées à la tâche rapide**
 lorsqu'elles sont configurées en entrées événementielles, le changement d'état physique de l'une de ces deux entrées, lance le cycle de la tâche rapide (front montant ou front descendant).
 Le changement d'état commandé par programme ou par terminal n'entraîne pas l'exécution du cycle de la tâche rapide.
- **Soient des entrées T.O.R. normales**
 lorsqu'elles ne sont pas configurées en entrées événementielles, elles peuvent être utilisées comme des entrées normales.
 Quelque soit l'utilisation de ces entrées, la faible immunité de celles-ci nécessite l'emploi d'un câble blindé pour leur raccordement.

5.2-2 Entrées/sorties affectées à la tâche rapide

Seul un octet d'entrée (I0,00 à I0,07 ou I0,08 à I0,15 ou I0,16 à I0,23), et un octet de sortie (O0,00 à O0,07 ou O0,08 à O0,15) de l'automate de base peuvent être affectés à la tâche rapide.

Les deux octets ne sont gérés par la tâche rapide que s'ils sont configurés à l'aide d'un terminal en mode CONFIGURATION.

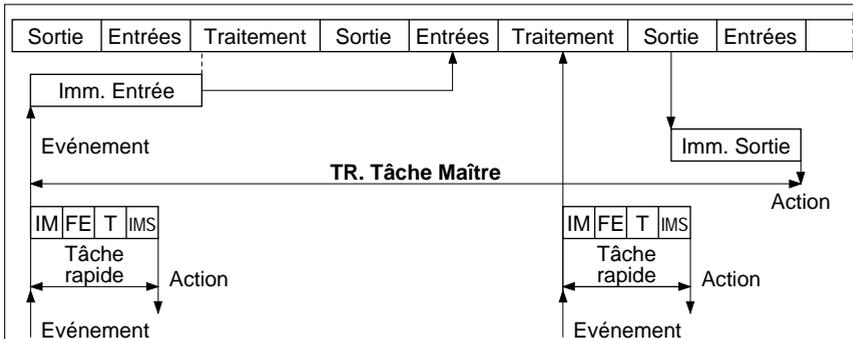
Lorsque la tâche rapide est activée, l'octet d'entrées et l'octet de sorties associés sont gérés par cette dernière. L'octet de sortie est géré par les 2 tâches.

Lorsque la tâche rapide n'est pas activée, l'octet d'entrées et l'octet de sorties associés sont gérés uniquement par la tâche maître.

5.2-3 Temps de réponse de la tâche rapide

Le temps de réponse de la tâche rapide (TRr) du TSX 17-20 est court et garanti. Il dépend de l'immunité de I0,24 et I0,25 "IM", du temps "FE" de désactivation de la tâche maître "Fin d'Exécution" d'un réseau,...), de son temps de cycle "T" et du temps de réponse des sorties "IMS" :

$$TRr = IM + FE + 1T + IMS$$



5.3 Tâche rapide des TSX 27/47

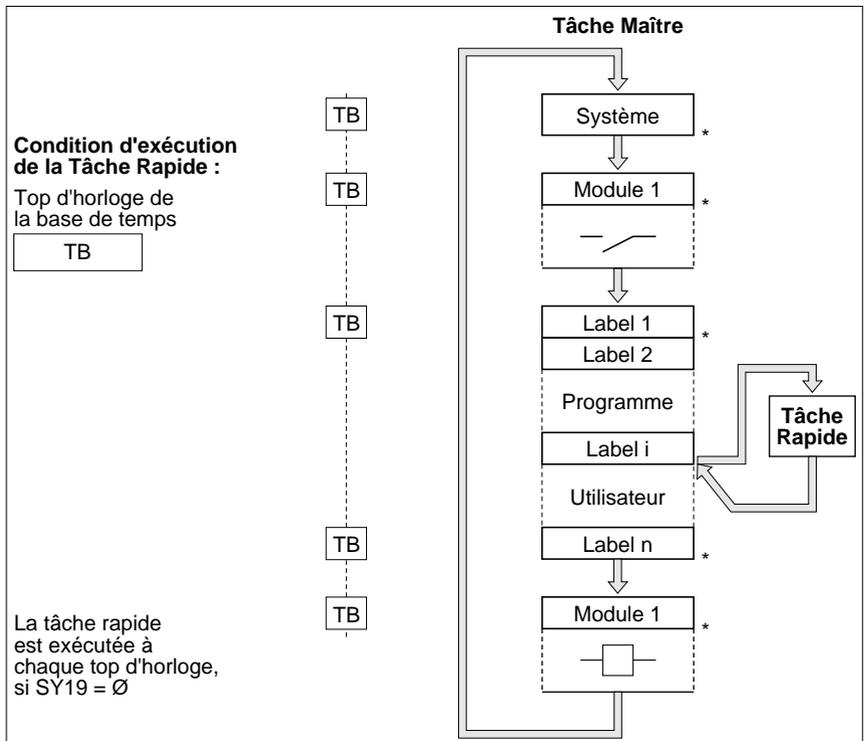
La tâche rapide des automates TSX 27/47 est destinée à des traitements de courte durée et à fréquences élevées.

Principe

Si SY19 = 0, cette tâche est activée, après interruption de la tâche maître, à chaque période d'une base de temps définie (5 ou 10 ms).

Si SY19 = 0 et si le top d'horloge est donné, l'interruption de la tâche maître et l'activation de la tâche rapide ne peuvent être effectuées que durant les phases suivantes :

- gestion du système,
- en fin de scrutation d'un module d'entrées/sorties (après lecture ou écriture complète),
- exécution du programme utilisateur :
 - après scrutation d'un réseau complet,
 - ou exécution d'une action (Grafcet),
 - ou évaluation d'une réceptivité (Grafcet).



5.3-1 Bases de temps

La période d'exécution de la tâche rapide est définie par le bit système SY3 :

- SY3 = 0 tâche rapide exécutée tous les 10 ms,
- SY3 = 1 tâche rapide exécutée tous les 5 ms. Cette période ne doit pas être utilisée si l'automate gère aussi un coupleur intelligent ou un coupleur TELWAY.

SY3	Base de temps	Temps d'exécution (1)	Nombre de réseaux
1	5ms	1ms	10 maximum
0	10ms	2ms	20 maximum

(1) Temps maximum assurant un fonctionnement correct.

5.3-2 Entrées/sorties affectées à la tâche rapide

L'affectation des entrées/sorties se définit par configuration :

- TSX 27 : 12 entrées (I1,0 à I1,B) et 8 sorties (O0,0 à O0,7) peuvent être affectées à la tâche rapide par configuration.
- TSX 47-J/47-10/20 : Seuls un module d'entrées et un module de sorties peuvent être affectés à la tâche rapide, par configuration.

Lorsque la tâche rapide est activée, les entrées et les sorties associées sont gérées par cette dernière. Le module de sortie est géré par les 2 tâches.

Lorsque la tâche rapide n'est pas activée, les entrées et les sorties associées sont gérées par la tâche maître.

5.4 Gestion des bases de temps

Les bases de temps des automates TSX 17-20/27/47 sont gérées par une horloge interne qui fonctionne de manière asynchrone par rapport au temps de cycle, quelle que soit la structure du système (monotâche ou bitâche).

Les bits et mots système associés à la gestion du temps (SY5, SY6, SY7, les bits et valeurs courantes de temporisateurs et monostables, ainsi que les temps d'activité d'étapes GRAFCET et les temps enveloppes de programmeurs cycliques), sont rafraîchies sur interruption système, indépendamment de la position dans le cycle, et ceci à chaque unité de leur base de temps respective.



Sous chapitre	page
6.1 Bits système	6/2
6.2 Mots système	6/8
6.2-1 Mots système TSX 47	6/8
6.2-2 Mots système TSX 17-20	6/9
 Ce chapitre se termine à la page	 6/12

6.1 Bits système

Les automates TSX 17-20/27/47-J/47-10/20 disposent de 24 bits système (SY0 à SY23) qui indiquent les états de l'automate ou permettent d'agir sur le fonctionnement de celui-ci.

Ces bits sont 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
SY0	Reprise à froid	<p>Normalement à l'état 0, est mis à l'état 1 par :</p> <ul style="list-style-type: none"> • changement d'une cartouche mémoire utilisateur, • reprise secteur avec perte des données (défaut batterie sur UC ou sur cartouche TSX 27/47. Défaut pile sur TSX 17-20 ou action sur verrou cartouche TSX 27/47 V<3) • programme utilisateur, • terminal (mode REGLAGE ou INIT), • changement. cartouche micro logiciel PL7-2 sur TSX 17-20. <p>Ce bit est mis à 1 durant le premier cycle complet. Il est remis à 0 avant le cycle suivant.</p> <p>Sur reprise à froid, l'automate repart toujours en exécution (RUN).</p> <p>Fonctionnement : voir sous chapitre 7.2-3 "comportement sur reprise à froid".</p>
SY1	Reprise à chaud	<p>Normalement à l'état 0, est mis à l'état 1 par :</p> <ul style="list-style-type: none"> • action sur un verrou cartouche TSX 47 V>3, • reprise secteur avec sauvegarde des données, • programme utilisateur, • terminal (mode REGLAGE). <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 sous chapitre 7.2-2 "comportement sur reprise à chaud"</p>
SY2	Extension adressage indexé 128 à 1024	<p>Uniquement pour TSX 17-20 VL > 2.0 :</p> <p>SY2 = 0 : adressage indexé W0...W127</p> <p>SY2 = 1 : adressage indexé W0...W1023</p>

Bits système	Fonctions	Désignation
SY3	Base de temps Tâche rapide TSX 27/47 V>3 Mise à l'heure Horodateur TSX 17-20	Normalement à l'état 0 : <ul style="list-style-type: none"> • SY3 = 0 tâche rapide exécutée toutes les 10 ms, • SY3 = 1 tâche rapide exécutée toutes les 5 ms, • SY3 = 0 rafraîchissement de l'heure validé • SY3 = 1 gel de l'horodateur pour mise à l'heure La gestion de ce bit est à la charge de l'utilisateur.
SY5 SY6 SY7	Bases de temps : 100ms 1s 1mm	Bits dont le changement d'état est cadencé par une horloge interne. Ils sont asynchrones par rapport au cycle de l'automate.
SY8	Sécurité des sorties (général)	Normalement 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 maintien les sorties dans l'état défini en cas de non exécution normale du programme ou sur STOP automate.
SY9	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, même lorsque celui-ci est en RUN, • état 0 : les sorties sont mises à jour normalement.
SY10	Défaut entrées/sorties (général) TSX 47 TSX 27 TSX 17-20	Normalement à l'état 1. Est mis à l'état 0 quand (le voyant I/O du processeur est également allumé) <ul style="list-style-type: none"> • défaut sur un module d'entrées/sorties d'un bac de base ou d'extension (configuration non conforme, défaut d'échange, absence bornier ou module, défaut module), • défaut sur un bac d'extension (coupure ou défaut alimentation, défaut liaison ou module d'extension). • Défaut d'un module d'entrées/sorties. • Défaut d'échange avec un module ou défaut de configuration. Les bits diagnostic des défauts d'entrées/sorties renseignent sur la nature du défaut (voir page 3.2 - intercalaire D). Le bit SY10 est remis à 1 dès la disparition du défaut.

Bits système	Fonction	Désignation
SY11	Rafraîchissement mots communs FIPWAY/TSX 17-20 TELWAY/TSX 47	Normalement à l'état 0, est mis à l'état 1 dès qu'une station locale a reçu les mots communs d'au moins une station distante. Ce bit doit être mis à 0 par programme. Il peut être mis à 0 par le terminal pour vérifier le cycle d'échange des mots communs
SY12	Réseau en fonctionnement FIPWAY/TSX 17-20 TELWAY/TSX 47	Normalement à l'état 0, est mis à l'état 1 dès que le coupleur de la station locale échange des mots communs avec une autre station du réseau. N'indique pas que toutes les stations fonctionnent. Est mis à l'état 0 sur défaut du coupleur.
SY13	Etat pile TSX 17-20	SY13 = 0 pile présente et en service SY13 = 1 pile absente ou hors service
SY14	Visualisation mémoire Memory Display TSX 17-20	Selon l'état du bit, la signification des voyants de la face avant de l'automate de base est différente : SY14 = 0 voyant MD éteint visualisation des sorties, SY14 = 1 voyant MD allumé visualisation des 16 bits de SW16 (poids faible à droite, poids fort à gauche).
SY15	Fin de cycle comptage rapide TSX 17-20	Normalement à l'état 0, ce bit est positionné par le système à 1 dès que la valeur courante de FC atteint la valeur de présélection. Il doit être remis à 0 par programme à la fin de la tâche rapide.
SY16	Réarmement sorties statiques TSX 17-20	Normalement à l'état 0, est mis à l'état 1 par l'utilisateur pour une demande de réarmement toutes les 10 s, des sorties statiques déclenchées sur surintensité ou sur court-circuit.
SY17	Dépassement de capacité (carry)	Normalement à l'état 0, est mis à l'état 1 : <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 en cas de dépassement.

Bits système	Fonction	Désignation
SY18	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 - 32 768, • Division par 0. <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>
SY19	Inhibition tâche rapide TSX 17-20 TSX 27/47 V≥3	<p>Normalement à l'état 1:</p> <p>SY19 = 1 inhibition tâche rapide SY19 = 0 validation tâche rapide</p> <p>La gestion de ce bit est à la charge de l'utilisateur.</p>
SY20	Débordement index (tâche) TSX 17-20 TSX 27/47 V>3	<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 127 (> à 1023 dans le cas d'un TSX 17-20 VL>2, avec SY2=1)</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> <p>Fonctionnement : voir sous chapitre 2.5-2 "débordement d'index".</p>
SY21	Initialisation GRAFCET (général)	<p>Normalement à l'état 0, est mis à l'état 1 par :</p> <ul style="list-style-type: none"> • reprise à froid, SY0=1, • le programme utilisateur uniquement dans le traitement préliminaire (PRE), • le terminal (demande d'initialisation par la touche dynamique INIT) ou modification de programme en mode connecté. <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 en début d'exécution du traitement séquentiel.</p>
SY22	Remise à zéro GRAFCET	<p>Normalement à l'état 0, ne peut être mis à l'état 1 par programme que dans le traitement préliminaire (PRE).</p> <p>A l'état 1, provoque la désactivation des étapes actives sur l'ensemble du GRAFCET.</p> <p>Est remis à l'état 0 par le système en début d'exécution du traitement séquentiel.</p>

Bits système	Fonction	Désignation
SY23	Prépositionnement et gel GRAFCET	<p>Normalement à l'état 0, ne peut être mis à l'état 1 par le programme utilisateur que dans le traitement préliminaire (PRE).</p> <p>A l'état 1, il permet de valider le prépositionnement du GRAFCET.</p> <p>Maintenu à l'état 1, il permet de maintenir le GRAFCET dans un état donné (gel du graphe).</p> <p>Est remis à 0 par le système en début d'exécution du traitement séquentiel.</p>

Tableau récapitulatif

Bit	Fonction	Etat init.	Gestion (*)	TSX		
				17-20	27	47
SY0	1 = démarrage à froid (reprise secteur avec perte des données)	0	S ou U → S	x	x	x
SY1	1 = démarrage à chaud (reprise secteur sans perte des données)	0	S ou U → S	x	x	x
SY2	0 = adres. indexé W0 à W127 1 = adres. indexé W0 à W1023	0	U	V≥2		
SY3	BT tâche rapide 0 = 10ms ; 1 = 5 ms	0	U	V≥3 V≥3		
	1 = gel de l'horodateur	0	U	x		
SY5	BT = 100ms		S	x	x	x
SY6	BT = 1s		S	x	x	x
SY7	BT = 1mn		S	x	x	x
SY8	0 = maintien des sorties sur STOP automate	1	U	x	x	x
SY9	1 = mise à zéro des sorties automate en RUN	0	U	x	x	x
SY10	0 = défaut E/S	1	S	x	x	x
SY11	1 = rafraîchissement variables COM réseau émises par station	0	S → U			x
SY12	1 = réseau et coupleur, station réseau OK	0	S			x
SY13	1 = pile absente ou défectueuse	0	S	x		
SY14	1 = visualisation du contenu de SW16	0	U	x		
SY15	1 = le compteur rapide C31 est égal à sa valeur de présélection	0	S → U	x		
SY16	1 = demande de réarmement toutes les 10s sorties statiques déclenchées sur surintensité ou court-circuit	0	U → S	x		
SY17	1 = débord. sur calcul non signé ou décal. circulaire	0	S → U	x	x	x
SY18	1 = débordement sur calcul (16 bits)	0	S → U	x	x	x
SY19	1 = inhibition tâche rapide	1	U	x	V≥3	V≥3
SY20	1 = débordement d'index	0	S → U	x	V≥3.1	V≥3.1
SY21	1 = initialisation du GRAFCET provoque la mise à 0 des étapes et la mise à 1 des étapes initiales	0	U → S	x	x	x
SY22	1 = RAZ de GRAFCET	0	U → S	x	x	x
SY23	1 = validation du prépositionnement du GRAFCET. Maintenu à 1 provoque le figeage du GRAFCET	0	U → S	x	x	x

(*) abréviations utilisées :

S = mis à 1 ou 0 par le système,

U = mis à 1 ou 0 par l'utilisateur,

S → U = mis à 1 par le système, remis à 0 par l'utilisateur,

U → S = mis à 1 par l'utilisateur, remis à 0 par le système.

6.2 Mots système

Les automates TSX 17-20/47 disposent de 60 mots système (SW0 à SW59) décrits ci-après :

6.2-1 Mots système TSX 47

Mots système	Fonction	Désignation
SW0	Mise à jour des mots communs TELWAY	Chaque bit de ce mot (0 à F) représente une station du réseau (0 à 15). L'état 1 d'un bit indique que la station correspondante à émis ses mots communs. Ce mot permet de s'assurer que l'automate de cette station est en RUN. Ce mot est réinitialisé à la valeur 0 seulement par programme.
SW1	_____	Mot système réservé.
SW2	Numéro de station TELWAY	Indique en décimal le numéro de la station codé sur le bornier de raccordement du coupleur TELWAY.
SW3	Table des stations TELWAY	Chaque bit (0 à F) de ce mot représente une station du réseau (0 à 15). L'état 1 d'un bit indique que le coupleur de la station correspondante fonctionne. Il est remis à jour par le système.

6.2-2 Mots système TSX 17-20

Mots système	Fonction	Désignation
SW0	Mise à jour des mots communs	Chaque bit de ce mot (0 à 15) représente une station du réseau (0 à 15). L'état 1 d'un bit indique que la station correspondante a émis ses mots communs. Ceci permet de s'assurer que l'automate de cette station est en RUN. Ce mot est réinitialisé à 0 seulement par programme.
SW1		SW1,15 = Présence coupleur (1 = coupleur présent) SW1,14 = Etat coupleur (1 = coupleur disponible. Ce bit n'est significatif que si SW1,15 = 1). SW1,13 = Echange des mots communs. (1 = fonction active. Significatif que si SW1,14 = 1). Autres bits SW1,● = réservés.
SW2	Numéros de réseau et de station	Ce mot indique le numéro de réseau et de station FIPWAY. Il correspond au codage effectué sur la partie avant du coupleur. Le numéro de réseau est indiqué par les bits de poids forts (8 à 15), le numéro de station est indiqué par les bits de poids faibles (0 à 7).
SW3 SW4 SW5 SW6	Table des stations FIPWAY	Chaque bit de ces mots représente une station du réseau. L'état 1 d'un bit indique que le coupleur de la station correspondante appartient au réseau. Ce mot, remis à jour par le système, n'indique pas que l'automate correspondant est en RUN.
SW16	Mot visualisable en mode visualisation mémoire (MD)	Si SY14 = 1, le contenu de SW16 est affiché en binaire par les voyants de la face avant du module de base. Lors de l'utilisation de l'afficheur numérique TSX 17 ACC2, le contenu de SW16 est affiché en BCD.
SW17	Checksum RAM	Contient le checksum de la RAM programme du TSX 17-20
SW18	Checksum cartouche EPROM ou EEPROM TSX 17-20	Contient le checksum (voir SW17) du programme utilisateur contenu dans la cartouche. Ce mot est mis à 0 si le commutateur de la cartouche EEPROM est en position "WORK". La gestion par programme de SW17 et SW18 permet de protéger le programme utilisateur contre les duplications, en imposant la présence de la cartouche automate en RUN (voir chapitre : Aide à l'exploitation – intercal. D).
SW24	Adresse UNI-TELWAY prise terminal	Adresse UNI-TELWAY de la prise terminal. Adresse définie par configuration et chargée dans SW24 lors d'une reprise à froid (SY0=1)

Mots système	Fonction	Désignation
SW40	Durée du dernier temps d'exécution programme	Durée (multiple de la base de temps 10 ms) du dernier temps d'exécution programme application (sur TSX 17-20 VL>2).
SW41	Durée du temps d'exécution programme le	Durée (multiple de la base de temps 10 ms) du temps d'exécution du programme application le plus long, depuis le dernier "démarrage à froid".
SW46 SW47 SW48	Codes modules d'extension	code module d'extension 1 (0 si absent) code module d'extension 2 (0 si absent) code module d'extension 3 (0 si absent)
SW50 SW51 SW52 SW53	Fonction Horodateur	Mots système contenant les valeurs courantes de la date et de l'heure (en BCD) : SW50 : siècle et année, SW51 : mois et jour, SW52 : heure et minute, SW53 : secondes et jour de la semaine.
SW54 SW55 SW56 SW57	Fonction Horodateur	Mots système contenant la date et l'heure du dernier défaut secteur ou arrêt automate (en BCD) : SW54 : siècle et année, SW55 : mois et jour, SW56 : heure et minute, SW57 : secondes et code du défaut.
SW58	Heure courante	Contient l'heure courante en minute dans le jour courant (0 à 1439 en décimal)
SW59	Date courante	Contient le numéro du jour courant dans l'année courante (1 à 366 en décimal). Les mots SW58 et SW59 permettent d'effectuer des opérations de calcul de durées.

Tableau récapitulatif

Mot	Fonction	TSX		Gestion (*)	Explication
		17-20	47		
SW0	Mise à jour des mots communs	x	x	S → U	Bits 0 à F = N° station 1 = mise à jour effectuée
SW1	Gestion du réseau	x	x	S	Réservé pour extension future
SW2	N° de réseau et de station	x	x	S	Bits 8 à F : n° réseau Bits 0 à 7 : n° station
SW3	Coupleur réseau OK	x	x	S	bits 0 à F = N° station 1 = coupleur station OK
SW4		x			
SW5					
SW6					
SW16	Mot visualisable en face avant de l'automate	x		U	Si SY14 = 1 affichage du contenu de SW16 sur visu automate
SW17	Checksum RAM programme	x		S	
SW18	Checksum cartouche "back up" = 0 si pas de backup	x		S	
SW24	Adresse UNI-TELWAY prise terminal	x		S	adresse définie par configuration. Chargée dans SW24 lors d'une reprise à froid (SY0 = 1)
SW40	Durée (base 10ms) des derniers temps d'exécution programme application	x		S	Sur TSX 17-20 VL ≥ 2
SW41	Durée (base 10ms) du temps de cycle le plus long depuis le dernier démarrage à froid	x	S		
SW46	Codes modules d'extension	x	S		mod 1 (0 si absent)
SW47					mod 2 (0 si absent)
SW48					mod 3 (0 si absent)

(*) abrégations utilisées :

S = mot écrit par le système,

U = mot écrit par l'utilisateur,

S → U = bits du mot mis à 1 par le système, mise à 0 à charge de l'utilisateur.

Tableau récapitulatif (suite)

Mot	Fonction	TSX 17-20 47	Gestion (*)	Explication
SW50	Contient le siècle et l'année en BCD	x	S	si option horodateur contiennent la date et l'heure courante
SW51	Contient le mois et le jour en BCD	x	S	
SW52	Contient l'heure et la minutes en BCD	x	S	
SW53	Contient les secondes, et le jour de la semaine en BCD	x	S	
SW54	Contient le siècle et l'année en BCD	x	S	si option horaodateur contiennent la date et l'heure de la dernière mise hors tension ou passage en STOP de l'automate
SW55	contient le mois et le jour en BCD	x	S	
SW56	Contient l'heure et la minute en BCD	x	S	
SW57	Contient les secondes, et le code d'arrêt en BCD	x	S	
SW58	Heure courante en minutes dans le jour courant (0 à 1439 en décimal)	x	S	si option horodateur SW58 et SW59 permettent de faire des opérations arithmétiques pour effectuer du calcul de durée
SW59	Numéro du jour courant dans l'année (1 à 366 en décimal)	x	S	

(*) abréviations utilisées :

S = mot écrit par le système,

U = mot écrit par l'utilisateur,

S → U = bits du mot mis à 1 par le système, mise à 0 à charge de l'utilisateur.



Sous chapitre	Page
7.1 Principe	7/2
7.1-1 Fonctionnement de l'automate à la coupure secteur	7/2
7.1-2 Sauvegarde du contexte système et application	7/3
7.1-3 Fonctionnement de l'automate à la reprise secteur	7/3
7.2 Comportement sur reprise secteur en structure monotâche	7/4
7.2-1 Comportement de l'automate sur reprise à chaud	7/4
7.2-2 Comportement de l'automate sur reprise à froid	7/6
7.3 Comportement sur reprise secteur en structure bitâche	7/8
7.3-1 Comportement sur reprise à froid	7/8
7.3-2 Comportement sur reprise à chaud après coupure en tâche maître	7/8
7.3-3 Comportement sur reprise à chaud après coupure en tâche rapide	7/8
7.4 Programmation de la reprise secteur	7/10
Ce chapitre se termine à la page	7/10

7.1 Principe

Pour éviter tout aléa de fonctionnement de l'automatisme en cas de coupure secteur ou de sous tension anormale, les modules alimentation des automates TSX 17-20/27/47 disposent d'un ensemble détectant la présence de la tension secteur.

L'alimentation du TSX 47 est équipée d'un dispositif de surveillance de la disparition secteur (dispositif au fonctionnement optionnel connecté sur le primaire des alimentations). Ce dispositif permet de traiter chaque coupure secteur en supprimant la prise en compte d'états d'entrées erronés suite à la disparition secteur (autonomie d'alimentation capteur inférieure à l'autonomie d'alimentation automate), et ceci tout en réduisant au maximum les délais de reprise de l'exécution du programme utilisateur. D'origine ce dispositif de surveillance secteur est en service.

7.1-1 Fonctionnement de l'automate à la coupure secteur

Durée de la coupure secteur	Comportement de l'automate
Inférieur à 10 ms	<ul style="list-style-type: none"> Exécution normale du programme
Supérieur à 10ms (2 ms avec module d'alimentation 24/48 VCC), et inférieure à l'autonomie de l'alimentation (environ 200 ms)	<p>TSX 47 avec circuit de surveillance en service</p> <ul style="list-style-type: none"> Arrêt de la scrutation du programme avant la disparition de la tension sur les entrées. Sauvegarde du contexte système et application. Suivant l'état du bit SY8 (sécurité) sorties forcées à zéro (SY8=1) ou maintenues à l'état (SY8=0). <p>TSX 47 avec circuit de surveillance hors service et TSX 17-20/27</p> <ul style="list-style-type: none"> Exécution normale du programme après reprise immédiate.
Supérieure à l'autonomie de l'alimentation (200ms).	<p>TSX 47 avec circuit de surveillance en service</p> <ul style="list-style-type: none"> Processeur arrêté hors tension. <p>TSX 17-20/27 et TSX 47 avec circuit de surveillance hors service</p> <ul style="list-style-type: none"> Arrêt de scrutation. Sauvegarde contexte⁽¹⁾. Processeur hors tension.

⁽¹⁾ sur TSX 17-20, le contexte n'est sauvegardé qu'une heure si l'automate n'est pas équipé d'une pile.

7.1-2 Sauvegarde du contexte système et application

Cette sauvegarde consiste pendant l'autonomie de l'alimentation à :

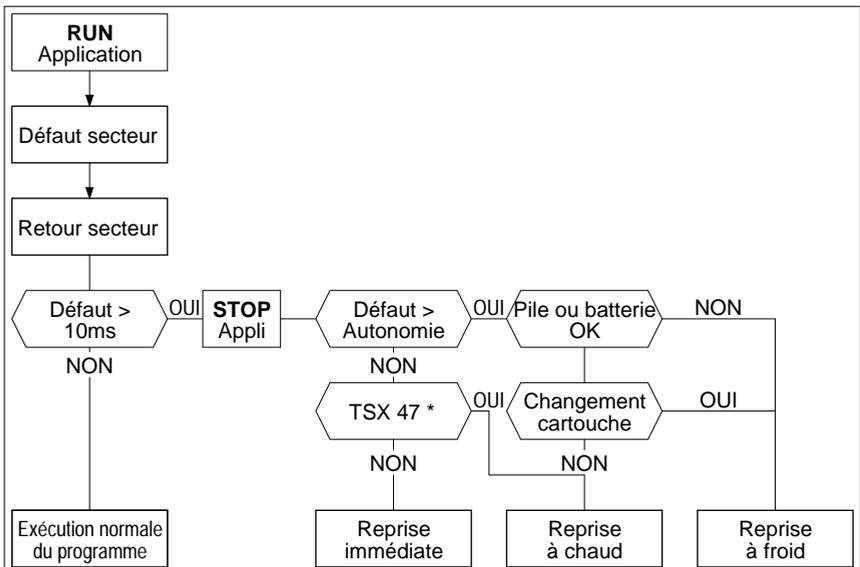
- calculer la valeur du checksum de la mémoire RAM du processeur (pointeur d'instruction, registres internes,...),
- transférer le résultat de ce calcul (checksum, etc) dans la mémoire RAM secourue du processeur de l'automate.

A la réapparition du secteur, cette sauvegarde est comparée au contexte présent et selon le résultat, le type de la reprise secteur est immédiat, à chaud ou à froid.

7.1-3 Fonctionnement de l'automate à la reprise secteur

Type de reprise secteur	Comportement de l'automate
Reprise après coupure inférieure à l'autonomie d'alimentation (200ms)	TSX 47 avec circuit de surveillance en service <ul style="list-style-type: none"> • Reprise à chaud, avec contexte cohérent vis-à-vis de l'application (programme, données et entrées/sorties sauvegardées).
Reprise après coupure supérieure à l'autonomie de l'alimentation : <ul style="list-style-type: none"> • batteries ou pile correctes et cartouches inchangées (SY1=1) • batteries ou piles défectueuses ou changement de cartouche (SY0=1) 	TSX 17-20/27 et TSX 47 avec circuit de surveillance hors service <ul style="list-style-type: none"> • Reprise à chaud, avec contexte cohérent vis-à-vis de l'application • Reprise à froid, avec perte du contexte : initialisation des données du langage

Synoptique

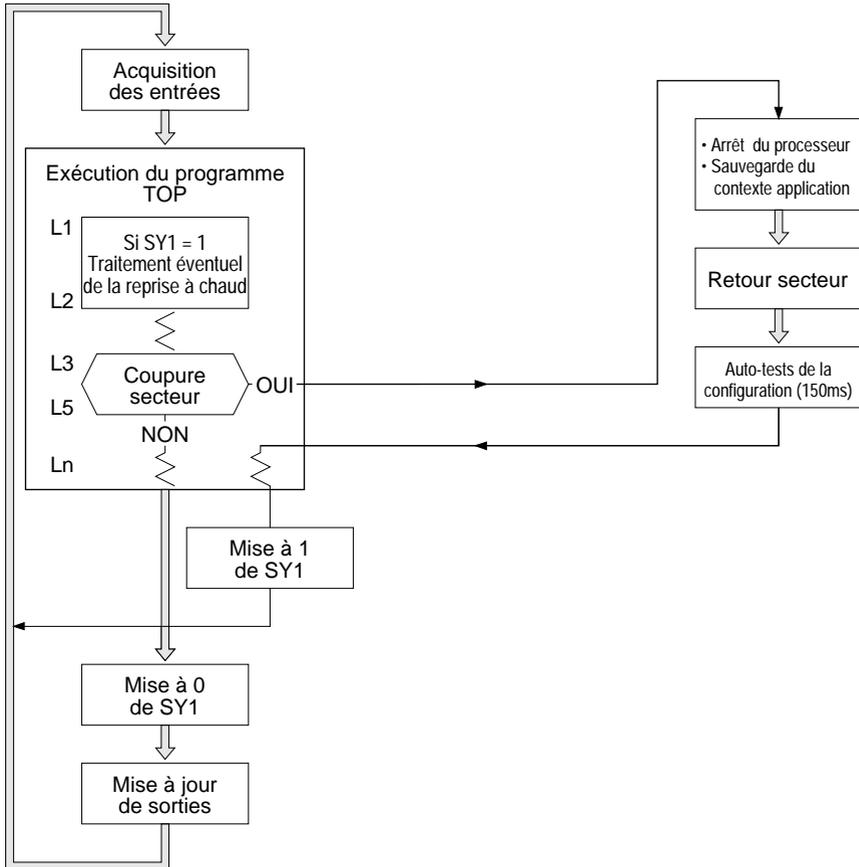


* TSX 47 avec circuit de surveillance en service

7.2 Comportement sur reprise secteur en structure monotâche

7.2-1 Comportement de l'automate sur reprise à chaud : SY1=1

Synoptique



Conséquences de la reprise à chaud :

- mise à 1 du bit système SY1,
- mise à 0 des bits de sorties, des bits internes non sauvegardés et des mots de registres IW/OW,
- initialisation des blocs textes.

Reprise de l'exécution du programme :

- après une coupure dans le programme utilisateur, l'exécution du programme reprend à partir de l'élément où a eu lieu la coupure secteur, sans mise à jour des sorties à la fin de ce cycle (voir illustration ci-contre).
- après une coupure en dehors du programme utilisateur, l'exécution reprend en début de cycle avec mise à jour des sorties, après mise à 0 de SY1.

Evolution des sorties

Pendant une coupure secteur, l'état des sorties dépend de l'état dans lequel l'utilisateur aura forcé le bit système SY8.

SY8 = 0 Maintien des sorties pendant l'autonomie de l'alimentation,

SY8 = 1 Mise à l'état 0 des sorties à partir de l'arrêt du processeur.

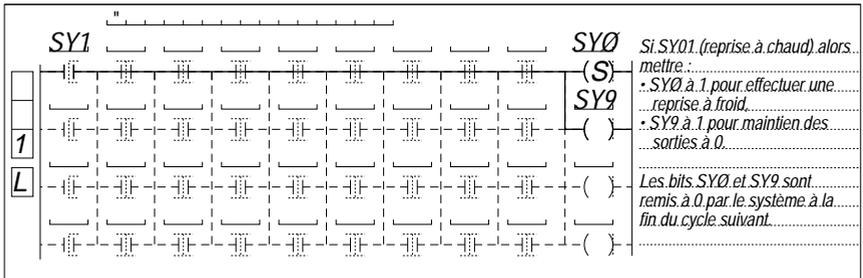
Traitement de la reprise à chaud

Si l'utilisateur désire un traitement particulier vis-à-vis de l'application en cas de reprise à chaud, il doit alors écrire en début de programme, sur test de SY1 à l'état 1, le programme correspondant (voir chapitre 2 - intercalaire D - traitement des modes de marche et d'arrêt).

• **Cas des TSX 17-20/TSX 27**

Bien que le contexte soit cohérent sur reprise à chaud, il y a lieu de s'assurer (vis à vis de la sécurité de l'application), que le processeur n'a pas pris en compte l'état d'entrées erronées à la disparition du secteur. Ceci du fait que le processeur continue pendant la réserve d'énergie l'acquisition des entrées alors que l'alimentation de celles-ci a disparu (voir manuel de mise en oeuvre TSX 17).

Sur certains types de TSX 17-20 (voir manuel de mise en oeuvre), ainsi que sur le TSX 27, toute reprise à chaud (SY1=1) doit entraîner une reprise à froid et le maintien des sorties à 0 (SY9=1).



• **Cas du TSX 47 avec circuit de surveillance en service**

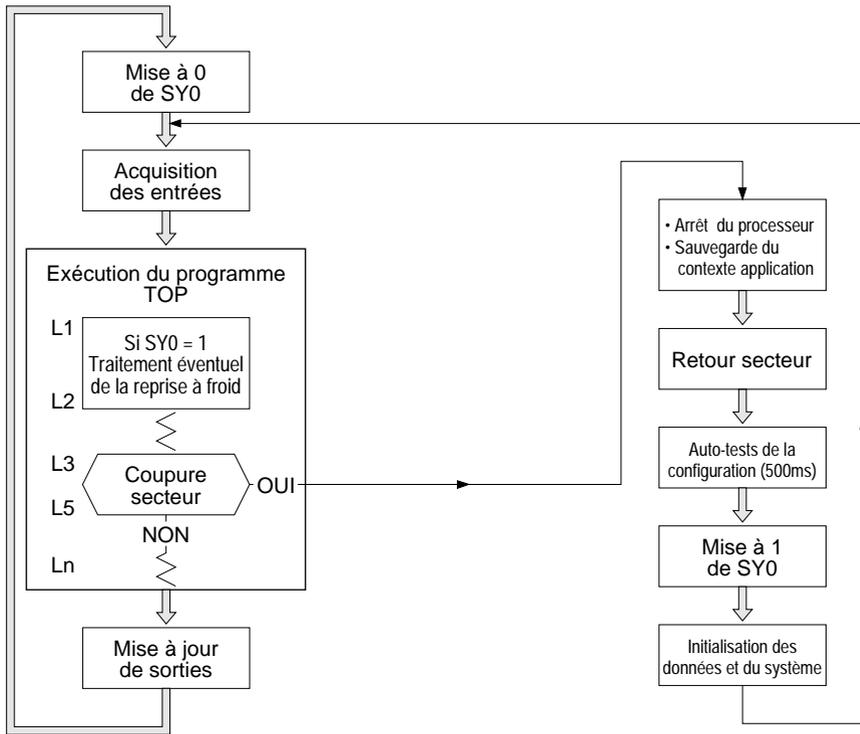
Aucun risque d'acquisition d'état d'entrées erronées. L'utilisateur doit tenir compte uniquement du traitement particulier vis à vis de l'application.

• **Cas du TSX 47 avec circuit de surveillance hors service**

voir cas des TSX 17-20/27.

7.2-2 Comportement de l'automate sur reprise à froid : SY0=1

Synoptique



Initialisation des données et du système

Ce qui correspond à :

- remise à zéro dans la mémoire bits des bits I/O, Bi,
- initialisation des bits et mots système,
- mise à l'état 1 du bit système SY21 pour initialisation du GRAFCET lors de l'exécution du traitement séquentiel,
- mise à 0 des valeurs courantes des blocs fonctions,
- mise à 0 des piles de registres,
- mise à zéro de l'espace données mots (Wi, Xi, V, I/OW, COMi, j , etc),
- perte des valeurs de présélection des blocs temporisateurs, monostables et compteurs, etc, acquise en mode REGLAGE,
- annulation des forçages de bits et des blocages d'étapes,
- initialisation des blocs textes et des programmeurs cycliques.

Les mots constants et les paramètres de l'horodateur (sur TSX 17-20 uniquement), ne sont pas modifiés. Sur reprise à froid, l'automate repart toujours en RUN si la mémoire contient un programme exécutable.

Traitement de la reprise à froid

Si l'utilisateur désire un traitement particulier vis-à-vis de l'application en cas de reprise à froid, il doit alors écrire en début de programme, sur test de SY0 à l'état 1, le programme correspondant (voir chapitre 2 - intercalaire D - traitement des modes de marche et d'arrêt).

Lors d'un démarrage à froid, le système positionne SY0 à 1, initialise les données, puis démarre en début de cycle (gestion système). Le bit SY0 est remis à 0 lors de la gestion système du cycle suivant.

Evolution des sorties

Quelque soit l'état du bit système SY8, les sorties sont initialisées à l'état 0 et seront positionnées lors de la mise à jour de celles-ci.

7.3 Comportement sur reprise secteur en structure bitâche

Le principe de fonctionnement des coupures et reprises secteur est alors affecté aux deux tâches (tâche maître et tâche rapide), de façon homogène.

Lors d'une reprise secteur (à chaud ou à froid), la tâche rapide demeure désactivée durant le tour de reprise et le premier tour de cycle complet. En effet, elle ne peut être exécutée que lorsque SY0 et SY1 sont à 0.

La tâche rapide ne pourra être lancée qu'à partir du tour de cycle suivant. Elle sera alors exécutée dès l'appel de son top d'horloge ou de l'apparition d'un événement.

7.3-1 Comportement sur reprise à froid

Il est identique à celui décrit au sous chapitre 7.2-2. La tâche maître est la seule tâche relancée.

7.3-2 Comportement sur reprise à chaud après coupure en tâche maître

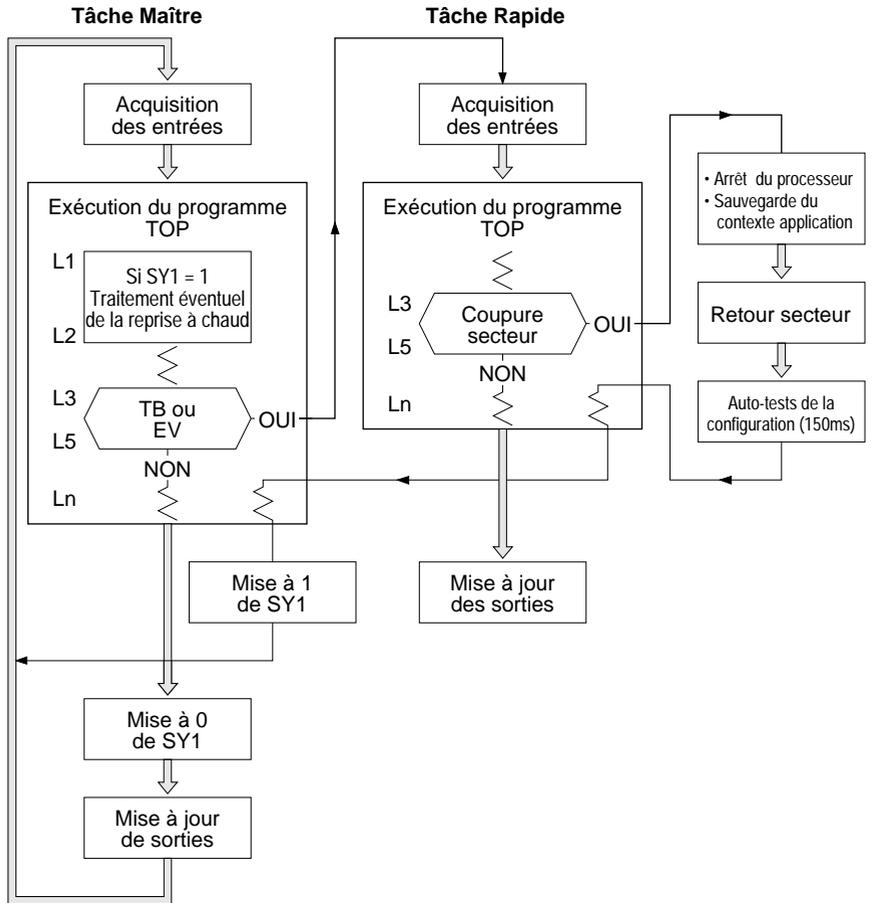
Le comportement de l'automate est identique à celui décrit au sous chapitre 7.2-1 si la coupure secteur survient lors de l'exécution de la tâche maître.

7.3-3 Comportement sur reprise à chaud après coupure en tâche rapide

Reprise de l'exécution du programme : le comportement lors de la reprise à chaud est homogène avec celui décrit au sous-chapitre 7.2-1.

- Interruption dans la tâche maître durant l'exécution du programme utilisateur :
 - Coupure en tâche rapide durant l'exécution de son programme : la reprise s'effectue là où a eu lieu la coupure sans mise à jour des sorties, puis retour en tâche maître sans mise à jour des sorties (voir illustration ci-contre).
 - Coupure en tâche rapide en dehors du programme utilisateur : retour en tâche maître et exécution de la suite du programme, sans mise à jour des sorties.
- Interruption dans la tâche maître en dehors du programme utilisateur :
 - Coupure en tâche rapide durant l'exécution de son programme : la reprise s'effectue là où a eu lieu la coupure sans mise à jour des sorties, puis retour en tâche maître, en début de cycle.
 - Coupure en tâche rapide en dehors du programme utilisateur : retour en début de cycle de la tâche maître.

Synoptique



Lors d'une reprise secteur (à chaud ou à froid), la tâche rapide demeure désactivée durant le tour de reprise et le premier tour de cycle complet pendant lequel SY0 ou SY1 est à 1.

7.4 Programmation de la reprise secteur

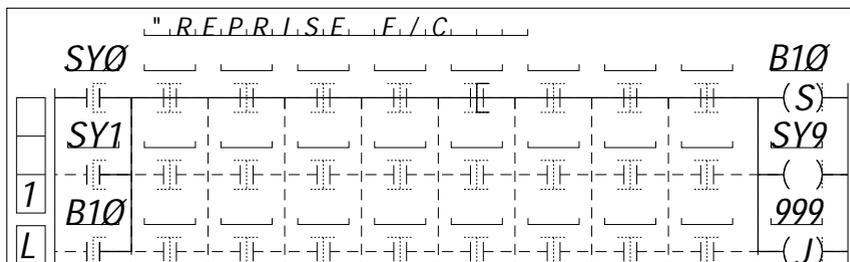
Lors de la reprise secteur, l'utilisateur peut souhaiter deux comportements de son automatisme :

- un démarrage automatique de l'application, comme s'il n'y avait pas eu de coupure,
- imposer une intervention humaine pour vérifier que tous les éléments de l'automatisme sont en ordre pour démarrer (solution recommandée par les normes de sécurité).

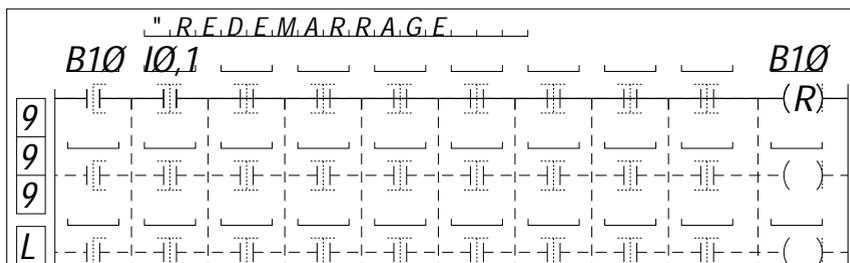
Les automates TSX 17-20/27/47 permettent l'une ou l'autre des solutions :

- reprise automatique : l'utilisateur doit veiller à la bonne charge de ses batteries ou piles.
- intervention humaine : rajouter dans le programme utilisateur les tests et actions suivantes (voir schéma ci-dessous). La mise à 1 de SY0 ou SY1 interrompt la scrutation du programme, force les sorties à 0 et attend une validation de la reprise par un personnel de surveillance.

Reprise secteur avec intervention humaine



Programme



Dès la mise à 1 de l'un des deux bits système, le programme met le bit B10 à 1 pour mémoriser cette information, positionne SY9 à 1, puis effectue un saut vers le dernier réseau du programme: entrée I0,1 (départ cycle) qui provoque la mise à 0 de B10. SY0 et SY1 étant remis à 0 par le système et le B10 étant à 0, le saut n'est plus effectué et le programme est donc exécuté entièrement.

Nota : I0,1 peut être un contact du contacteur KA destiné à l'asservissement de l'alimentation des sorties (voir manuels de mise en oeuvre).



Chapitre	Page
1 <i>Présentation du langage à contacts PL7-2</i>	1/1
1.1 Principe	1/2
1.2 Le langage à contacts : traitement monotâche et bitâche	1/3
2 <i>Les éléments du langage à contacts PL7-2</i>	2/1
2.1 Les éléments graphiques de base	2/2
2.2 Objets bits adressables	2/4
2.3 Objets mots adressables	2/4
2.4 Les blocs fonction	2/4
2.5 Le bloc comparaison	2/7
2.6 Les blocs opération	2/7
3 <i>Réseau de contacts</i>	3/1
3.1 Structure d'un réseau de contacts	3/2
3.2 Règles d'exécution d'un réseau de contacts	3/9
4 <i>Blocs fonction</i>	4/1
4.1 Bloc fonction : Temporisateur	4/2
4.2 Bloc fonction : Monostable	4/8
4.3 Bloc fonction : Compteur/Décompteur	4/11
4.4 Bloc fonction : Compteur/Temporisateur rapide (TSX 17-20)	4/14
4.5 Bloc fonction : Registre	4/18
4.6 Bloc fonction : Programmateur cyclique	4/21
4.7 Bloc fonction : Comparateur	4/24
5 <i>Bloc opération</i>	5/1
5.1 Bloc opération "calcul et transfert"	5/2
5.2 Opérations sur chaînes de bits	5/3
5.3 Opérations sur chaînes de mots	5/7



Chapitre	Page
6 <i>Blocs fonction texte</i>	6/2
6.1 Présentation	6/2
6.2 Description de l'échange	6/2
6.3 Caractéristiques	6/6
6.4 Fonctionnement	6/8
7 <i>Bloc fonction horodateur (TSX 17-20)</i>	7/1
7.1 Présentation	7/2
7.2 Programmateur temporel	7/2
7.3 Consignateur temporel	7/6
7.4 Mesure de durée	7/8
7.5 Exemple de mise à l'heure de l'horodateur	7/9



Sous-chapitre	Page
1.1 Principe	1/2
1.2 Le langage à contacts : traitement monotâche et bitâche	1/3
1.2-1 Traitement monotâche	1/3
1.2-2 Traitement bitâche	1/3
 Ce chapitre se termine à la page	 1/4

1.1 Principe

Un programme écrit en langage à contacts se compose d'une suite de réseaux exécutés séquentiellement par l'automate.

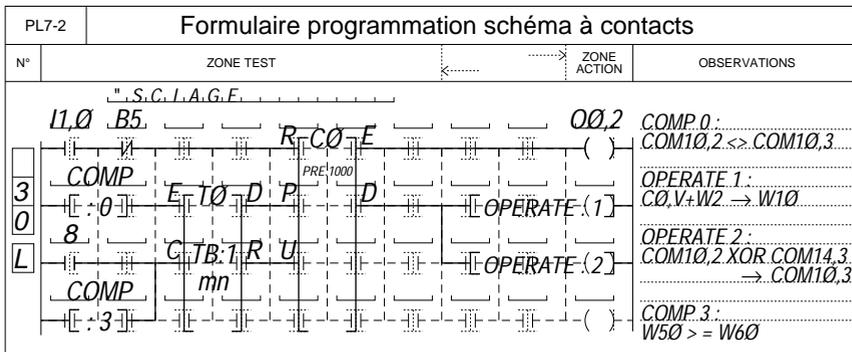
Dessiné entre deux barres de potentiel, un réseau est un ensemble d'éléments graphiques représentant :

- les entrées/sorties de l'automate (boutons-poussoirs, détecteurs, relais, voyants...),
- des fonctions d'automatismes (temporisateurs, compteurs...),
- des opérations arithmétiques et logiques et des opérations de transfert,
- les variables internes de l'automate (bits, mots,etc).

Ces éléments graphiques sont reliés entre eux par des connexions horizontales et verticales, définissant ainsi des "réseaux de contacts".

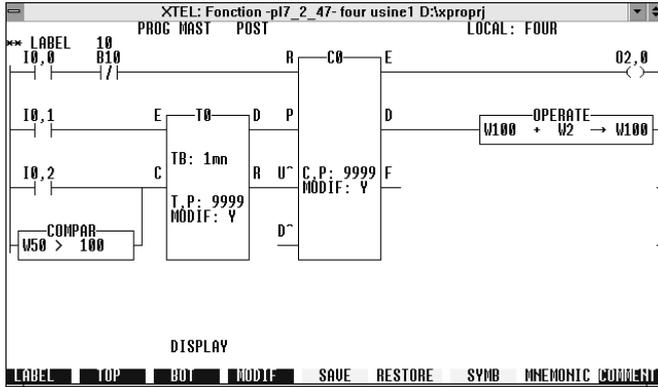
Chaque réseau ainsi constitué comporte au plus 4 lignes et 10 colonnes et s'organise en 2 zones :

- la zone test, dans laquelle figurent les conditions nécessaires à une action,
- la zone action, qui sanctionne le résultat consécutif à un enchaînement de test.



Des formulaires de programmation "schéma à contacts" sont prévus pour la représentation des réseaux. Ils sont conformes à la structure des réseaux, pour la programmation en langage PL7-2 : un réseau à la fois peut être visualisé sur l'écran du terminal.

Exemple de présentation sur écran terminal, FTX 417/507 ou compatible PC

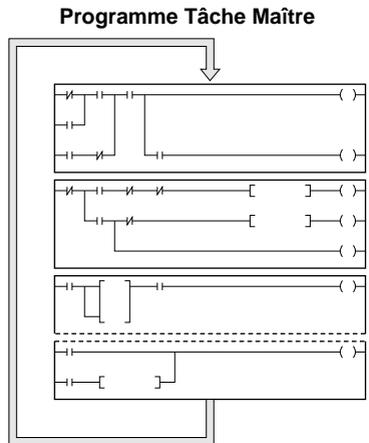


1.2 Le langage à contacts : traitement monotâche et bitâche

1.2-1 Traitement monotâche

La tâche maître est exécutée périodiquement de façon cyclique.

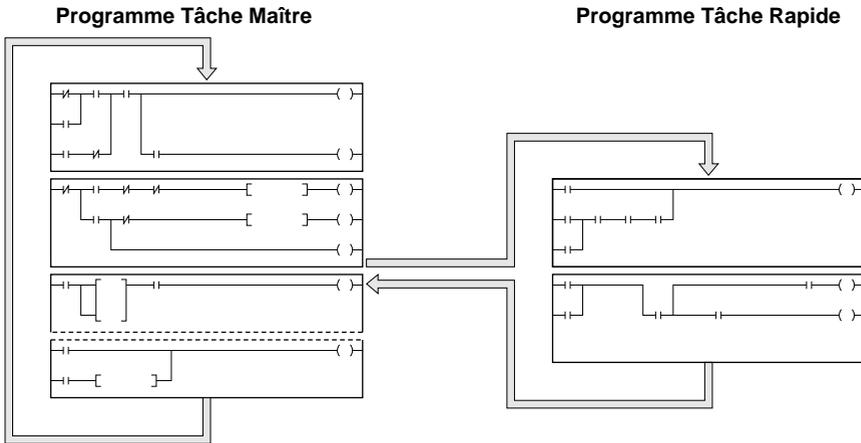
Elle comprend un programme constitué d'une suite de réseaux de contacts (maximum 999).



1.2-2 Traitement bitâche

En traitement bitâche, le système gère deux programmes distincts :

- un programme dans la tâche maître (voir 1.2-1),
- un programme dans la tâche rapide, constitué aussi d'une suite de réseaux de contacts. Le nombre de réseaux doit être réduit (< 20), afin de garantir un temps de cycle de la tâche rapide < 2 ms.





Sous-chapitre	Page
2.1 Les éléments graphiques de base	2/2
2.2 Les objets bits adressables	2/4
2.3 Les objets mots adressables	2/4
2.4 Les blocs fonctions	2/4
2.4-1 Blocs fonctions d'automatisme	2/5
2.4-2 Bloc fonction de communication	2/6
2.5 Le bloc Comparaison	2/7
2.6 Les blocs opérations	2/7
2.6-1 Transfert	2/7
2.6-2 Décalage	2/7
2.6-3 Transcodage	2/8
2.6-4 Opérations logiques	2/8
2.6-5 Opérations arithmétiques	2/9
2.6-6 Opérations sur chaînes de bits	2/9
2.6-7 Opérations sur tableaux de mots	2/10
Ce chapitre se termine à la page	2/10

2.1 Les éléments graphiques de base

Désignation		Graphes	Fonctions
Eléments de test	• Contact à fermeture (direct)		Contact passant quand l'objet bit qui le pilote est à l'état 1.
	• Contact à ouverture (indirect)		Contact passant quand l'objet bit qui le pilote est à l'état 0.
Eléments de liaison	• Connexion horizontale		Permet de relier en série les éléments graphiques de test et d'action entre les deux barres de potentiel.
	• Connexion verticale		Permet de relier en parallèle les éléments graphiques de test et d'action.
Eléments d'action	• Bobine directe		L'objet bit associé prend la valeur du résultat logique de la zone test.
	• Bobine inverse		L'objet bit associé prend la valeur inverse du résultat logique de la zone test.
	• Bobine d'enclenchement		L'objet bit associé est mis à 1 et garde cet état, lorsque le résultat de la zone test est à 1. Il est remis à 0 par la bobine de déclenchement.
	• Bobine de déclenchement		L'objet bit associé est mis à 0 et garde cet état, lorsque le résultat de la zone test est à 1. Il est remis à 1 par la bobine d'enclenchement.
	• Bobine saut à un autre réseau (JUMP) maximum 42		Permet un branchement à un réseau étiqueté, amont ou aval. Les sauts ne sont effectifs qu'au sein d'une même entité de programmation (programme principal, traitement préliminaire...) L'exécution d'un saut provoque : <ul style="list-style-type: none"> • l'arrêt de la scrutation du réseau en cours, • l'exécution du réseau étiqueté demandé, • la non scrutation de la partie du programme située entre l'action de saut et le réseau désigné. L'instruction de saut n'est pas permise en tâche rapide.

Désignation		Grappe	Fonctions
Eléments d'action (suite)	• Bobines sauvegardées	$\neg(\)^*$	Bobines sauvegardées en cas de coupure secteur.
		$\neg(\ /)^*$	Les états des objets bits associés sont sauvegardés durant le premier tour de cycle.
		$\neg(S)^*$	
		$\neg(R)^*$	

Nota : les bobines non sauvegardées sont maintenues à 0 pendant le premier tour de cycle de reprise.

2.2 Les objets bits adressables

Le langage à contacts permet de manipuler les objets bits :

- contenus dans la mémoire bits :
 - bits d'entrées/sorties,
 - bits internes,
 - bits associés aux étapes GRAFCET,
 - bits système.
- extraits de l'espace mémoire de données :
 - bits d'état des modules,
 - bits extraits de mots communs,
 - bits extraits de mots,
 - bits associés aux blocs fonctions.
- extraits de l'espace mémoire "constantes" :
 - bits extraits de mots constants.

Ces bits peuvent, selon le cas, être lus ou écrits par l'utilisateur.

L'adressage et la liste des objets bits sont définis dans "Généralités" intercalaire A, chapitre 2.

2.3 Les objets mots adressables

Le langage à contacts accède aux objets mots lors de l'utilisation de blocs fonctions d'automatismes ou de blocs opérations.

Ces mots, selon leur nature, peuvent être lus et/ou écrits.

L'adressage et la liste des objets mots sont définis dans "Généralités" intercalaire A, chapitre 2.

2.4 Les blocs fonctions

Chacun des blocs fonctions représentés ci-après utilise :

- des entrées et des sorties permettant de le relier à des éléments graphiques dans un réseau,
- des paramètres de configuration, accessibles ou non par le programme utilisateur,
- des paramètres courants (objets bits de sortie, mot de valeur courante, mot d'entrée), lus ou écrits par l'utilisateur (voir "Généralités" intercalaire A, chapitre 2).

2.4-1 Blocs fonctions d'automatisme

Désignation (Nb)	Graphes	Fonctions
Temporisateur TSX 17-20 (32) TSX 27/47 (16)		E entrée "armement" C entrée "contrôle" Ti,D bit de sortie temporisation écoulée Ti,R bit de sortie temporisation en cours Ti,P mot valeur de présélection Ti,V mot valeur courante
Monostable TSX 17-20 (8) TSX 27/47 (8)		S entrée sur front "départ" Mi,R bit de sortie monostable en cours Mi,P mot valeur de présélection Mi,V mot valeur courante
Compteur/ décompteur TSX 17-20 (31) TSX 27/47 (16)		R entrée "remise à zéro" P entrée "présélection" U entrée sur front "comptage" D entrée sur front "décomptage" Ci,E bit de sortie "débordement décomptage" Ci,D bit de sortie "présélection atteinte" Ci,F bit de sortie "débordement comptage" Ci,P mot valeur de présélection Ci,V mot valeur courante
Registre LIFO/FIFO TSX 17-20 (4) TSX 27/47 (4)		R entrée "remise à zéro" I entrée sur front "stockage" O entrée sur front "déstockage" Ri,F bit de sortie "registre plein" Ri,E bit de sortie "registre vide" Ri,I mot d'entrée du registre Ri,O mot de sortie du registre L nombre de mots du registre (999 max)
Programmateur cyclique TSX 17-20 (8) TSX 27/47 (8)		R entrée "Remise A Zéro" U entrée sur front "Avance de pas" Di,F bit de sortie "dernier pas en cours" Di,S mot "numéro du pas actif" Di,V mot "temps enveloppe" Di,Wj mot 16 bits d'ordres

Bloc fonction d'automatisme (suite)

Désignation (Nb)	Graphe	Fonctions
Compteur/ Temporisateur rapide TSX 17-20 (1)		R entrée "remise à zéro" P entrée validation "présélection" V entrée validation "lecture à la volée" C entrée validation comptage FC,E bit de sortie "Remise A Zéro externe" FC,D bit de sortie "présélection atteinte" FC,F bit de sortie "comptage en cours" Ci,P mot valeur de présélection Ci,V mot valeur courante
HORODATEUR TSX 17-20 (illimité)		E entrée "Validation fonctionnement" (ENABLE) < bit de sortie "Avant" = bit de sortie "Pendant" > bit de sortie "Après"

2.4-2 Bloc fonction de communication

Désignation (Nb)	Graphe	Fonctions
Bloc fonction texte TSX 17-20 (8) TSX 27/47 (8)		R entrée validation arrêt échange S entrée sur front "lancement échange" O entrée sur état "émission" I entrée sur état "réception" TXTi,D bit de sortie "transfert terminé et reçu" TXTi,E bit de sortie "erreur de transfert" TXTi,L mot "longueur du message" (en octets) TXTi,S mot status : compte-rendu de l'échange TXTi,M mot adresse coupleur et numéro de voie TXTi,C mot code requête TXTi,R mot compte rendu échange TXTi,T num. bloc TXT en communi. TXTi,A adresse station réseau

2.5 Le bloc comparaison

Désignation (Nb)	Graphe (*)	Fonctions
Bloc comparaison (illimité)		OP : opérateur : <, <=, =, >, >=, <> OP1 : objet mot pouvant être lu OP2 : objet mot pouvant être lu ou valeur immédiate

(*) sur terminal de programmation TSX T407, le bloc comparaison est illustré comme suit : 

2.6 Les blocs opérations

Ce sous chapitre a pour but de rappeler les différentes possibilités d'écriture des expressions simples dans les blocs opérations.

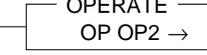
Ces blocs peuvent être utilisés en quantité non limitée, celle-ci dépend de la taille de la mémoire "utilisateur".

Les blocs opérations énumérés ci-dessous sont décrits et détaillés dans le chapitre 4 : "Blocs fonctions".

2.6-1 Transfert

Désignation (Nb)	Graphe (*)	Fonctions
Bloc-opération (illimité)		OP2 : objet mot pouvant être lu ou valeur immédiate → : objet mot de rangement

2.6-2 Décalage

Désignation (Nb)	Graphe (*)	Fonctions
Bloc-opération (illimité)		OP : type de décalage : SLC n : décalage circulaire à gauche de n bits SRC n : décalage circulaire à droite de n bits OP2 : objet mot de 16 bits → : objet mot de rangement du résultat

(*) sur terminal de programmation TSX T407, le bloc opération est illustré comme suit : 

2.6-3 Transcodage

Désignation (Nb)	Graphe	Fonctions
Bloc- opération (illimité)		OP : type de conversion : BIN : conversion BCD vers BINaire BCD : conversion BINaire vers BCD OP2 : objet mot de 16 bits → : objet mot de rangement du résultat
		ATB : conversion ASCII vers BINaire OP2 : tableau de 3 mots → : objet mot de rangement du résultat
		BTA : conversion BINaire vers ASCII OP2 : objet mot → : tableau de 3 mots pouvant être écrits

2.6-4 Opérations logiques

Désignation (Nb)	Graphe	Fonctions
Bloc- opération (illimité)		OP1 : objet mot pouvant être lu OP : code opération logique AND : "ET" logique OR : "OU" inclusif XOR : "OU" exclusif CPL : complément logique OP2 : objet mot pouvant être lu ou valeur immédiate → : objet mot de rangement du résultat

2.6-5 Opérations arithmétiques

Désignation (Nb)	Graphe	Fonctions
Bloc- opération (illimité)		OP1 : objet mot pouvant être lu OP : code opération arithmét. + : addition - : soustraction * : multiplication / : division (quotient) REM : division (reste) pour TSX 17–20 uniquement OP2 dans ce cas doit être différent de 0 OP2 : objet mot pouvant être lu ou valeur immédiate → : objet mot de rangement du résultat

2.6-6 Opérations sur chaînes de bits

Bit i, Bit j : les objets bits manipulés peuvent être des bits internes, des bits extraits de mots, des bits d'entrées-/sorties.

[n] : constante entière positive, dépend de la configuration et des objets manipulés. Définit la longueur de la chaîne de bits.

Désignation (Nb)	Graphe	Fonctions
Transfert chaîne de bits → chaîne de bits		
Bloc- opération (illimité)		OP2 : Bi[n] → : Bj[n]
Transfert chaîne de bits → mot		
Bloc- opération (illimité)		OP2 : Bi[n] → : objet mot de rangement pou- vant être écrit
Transfert chaîne de bits ou mot → chaîne de bits		
Bloc- opération (illimité)		OP2 : objet mot pouvant être lu ou valeur immédiate → : Bj[n]

2.6-7 Opérations sur tableaux de mots

Un tableau de mots, ou chaîne de mots, est une suite continue de mots de même type.

MOT i [MOT j] : MOT i : mot début de la table

MOT j : mot index

(voir Généralités, intercalaire A, chapitre 2.5: indexation des objets).

Désignation (Nb)

Graphe

Fonctions

Transfert indexé

Bloc-
opération
(illimité)



OP2 : MOT i [MOT j]

→ : MOT I [MOT k]



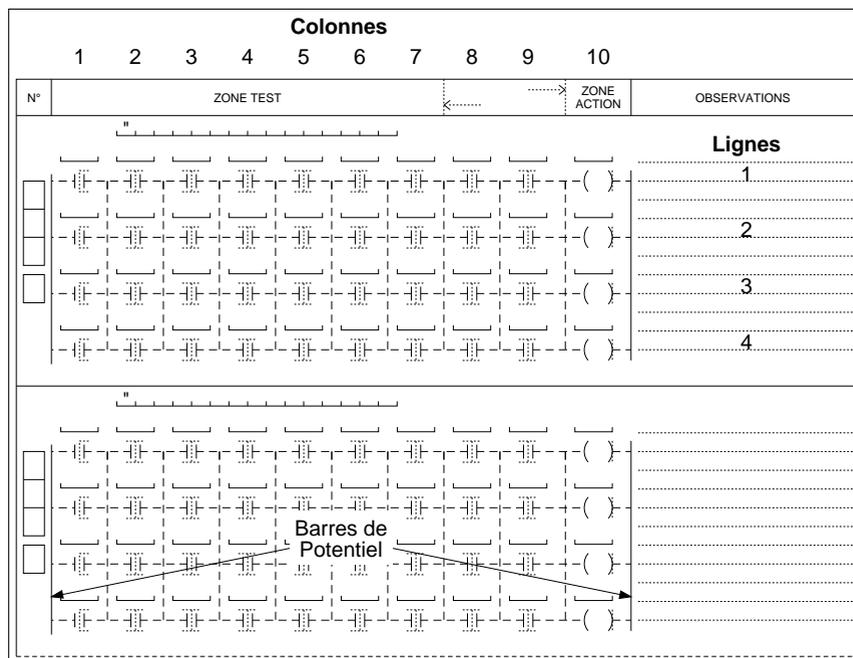
Sous-chapitre	Page
3.1 Structure d'un réseau de contacts	3/2
3.1-1 Généralités	3/2
3.1-2 Etiquette ou label	3/3
3.1-3 Commentaire	3/4
3.1-4 Réseau de contacts	3/5
3.1-5 Réseau de contacts avec blocs fonctions	3/7
3.2 Règles d'exécution d'un réseau de contact	3/9
3.2-1 Principe de scrutation d'un programme	3/9
3.2-2 Principe d'exécution d'un réseau	3/9
3.2-3 Saut à un autre réseau (JUMP)	3/10
3.2-4 Exécution d'un réseau comportant des blocs fonctions	3/11
3.2-5 Scrutation d'un réseau comportant des blocs opérations	3/12
Ce chapitre se termine à la page	3/12

3.1 Structure d'un réseau de contacts

3.1-1 Généralités

Un réseau s'inscrit entre deux barres de potentiel et est composé d'un ensemble d'éléments graphiques reliés entre eux par des liaisons horizontales ou verticales. Un réseau est limité à la dimension d'un écran du terminal, soit 4 lignes et 10 colonnes maximum réparties en deux zones : la zone test et la zone action.

Les éléments graphiques énumérés précédemment peuvent s'implanter soit dans la zone action, soit dans la zone test sur l'ensemble des lignes et colonnes d'un réseau.



Chaque réseau doit être repéré par une étiquette, et peut être précédé par un commentaire.

Nota : Un formulaire de programmation schéma à contacts comporte 4 réseaux.

3.1-2 Etiquette (ou label)

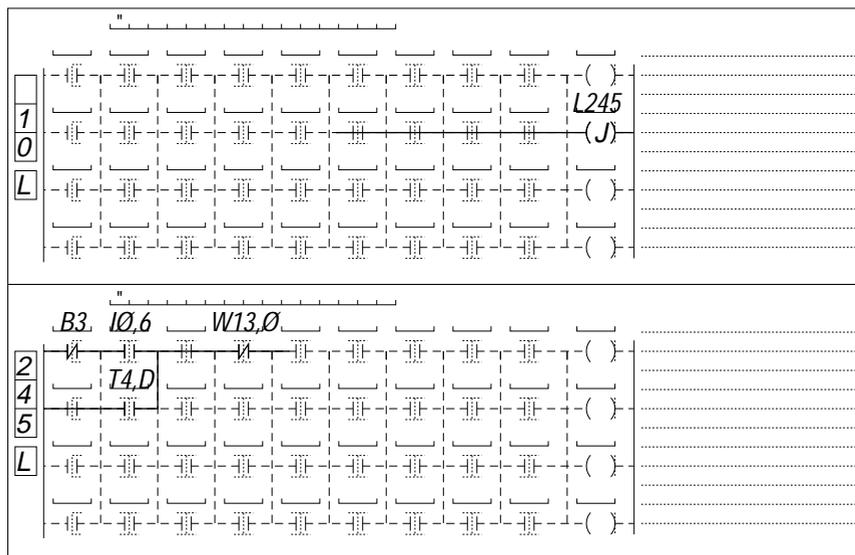
L'étiquette ou label permet de repérer un réseau dans une entité de programme :

- programme langage à contacts en tâche maître : label L,
- programme langage à contacts en tâche rapide : label F,
- programme langage GRAFCET en tâche maître :
 - schémas à contacts en traitement préliminaire : label p,
 - schémas à contacts en traitement postérieur: label P.

Cette étiquette comprend au maximum 3 caractères numériques (1 à 999) et se positionne à la partie supérieure gauche devant la barre de potentiel. Un réseau sous étiquette ne pourra être validé lors de la saisie sur terminal.

Un repère d'étiquette ne peut être affecté qu'à un seul réseau au sein d'une même entité de programme.

L'étiquette permet entre autre le branchement après un saut de programme.



3.1-3 Commentaire

Le commentaire est intégré au réseau et comprend 15 caractères alphanumériques au maximum. Il facilite l'interprétation du réseau auquel il est affecté, mais n'est pas obligatoire.

PL7-2		Formulaire programmation schéma à contacts			
N°	ZONE TEST			ZONE ACTION	OBSERVATIONS
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>					Zone commentaires.....
					non mémorisés dans.....
	Commentaire mémorisé dans mémoire automate				l'automate.....
					(sur formulaire.....
					uniquement).....

En cas de suppression d'un réseau, le commentaire qui lui est associé est également supprimé.

Les commentaires sont mémorisés dans l'automate et sont accessibles à tout moment à l'utilisateur.

3.1-4 Réseau de contacts

Sa représentation est proche de celle d'un schéma électrique à relais.

Les éléments graphiques simples de test et d'action occupent chacun une ligne et une colonne au sein d'un réseau.

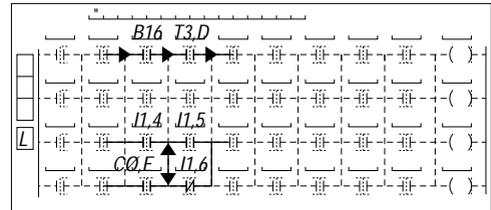
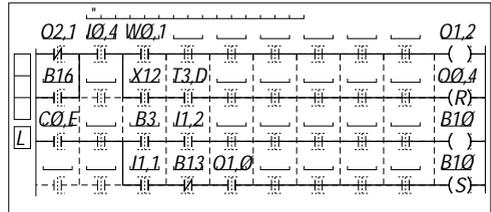
Toute ligne de contact commencée à gauche doit conduire à droite.

Les tests sont toujours situés sur les colonnes 1 à 9.

Les actions sont toujours situées sur la colonne 10.

Le sens de circulation du courant est le suivant :

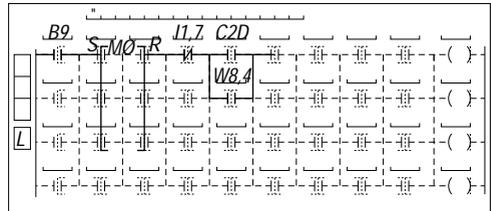
- pour les liaisons horizontales, de la gauche vers la droite,
- pour les liaisons verticales, dans les deux sens.



• Zone test

On y trouve :

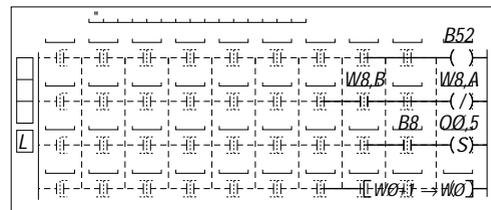
- les contacts, auxquels peuvent être affectés tous les objets bits définis précédemment,
- les blocs fonctions,
- les blocs comparaison.



• Zone action

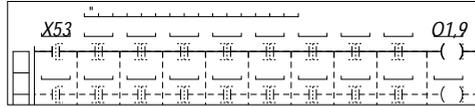
On y trouve :

- les bobines directes, inverses, d'enclenchement et de déclenchement, qui peuvent être affectées de tous les objets bits pouvant être écrits par l'utilisateur.
- les blocs opérations.

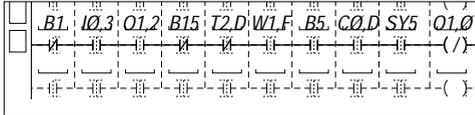


Réseau simple

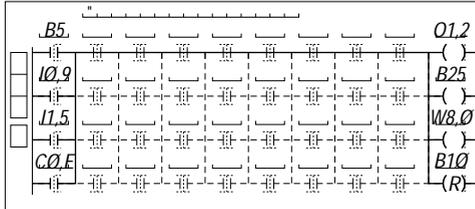
Pilotage d'une bobine conditionnée par un contact.



Utilisation de 9 contacts maximum en série sur une ligne.



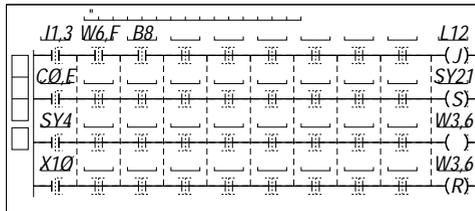
Possibilité de tester 4 contacts maximum en parallèle sur une colonne, de placer 4 bobines en parallèle.



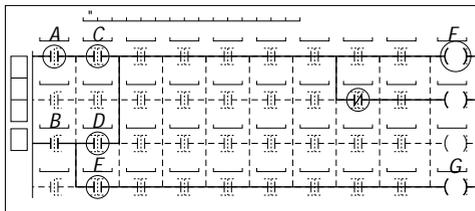
Réseau utilisant plusieurs chaînes de contacts

Un réseau peut être divisé en plusieurs chaînes de contacts indépendantes pilotant des bobines indépendantes

4 lignes indépendantes de 9 contacts.



Réseaux utilisant les différents principes énoncés. Les éléments à l'état logique 1 sont cerclés. Le courant peut passer des éléments A et C vers la bobine F, il ne peut pas passer de l'élément C vers les éléments D et E. La bobine G n'est pas activée.

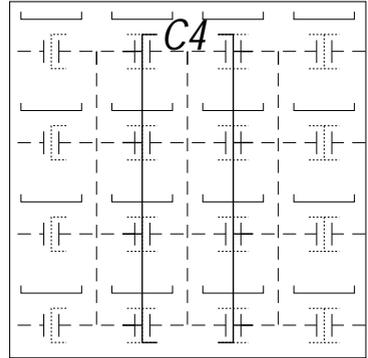


3.1-5 Réseau de contacts avec blocs fonctions et opérations

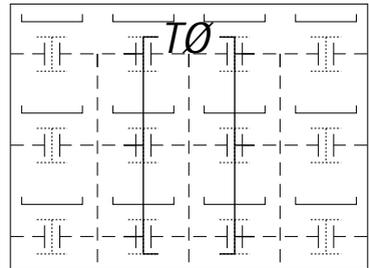
- Les blocs fonctions se positionnent dans la zone test et s'insèrent dans un réseau de contacts.

Il existe trois dimensions de graphisme pour représenter l'ensemble des autres blocs fonctions du langage PL7-2 :

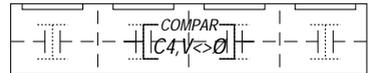
Compteur/Décompteur 2 colonnes
 Compteur/Temporisateur rapide 4 lignes
 Bloc texte



Temporisateur Monostable 2 colonnes
 3 lignes
 Registre
 Programmeur cyclique
 Horodateur



Comparateur 2 colonnes
 1 ligne



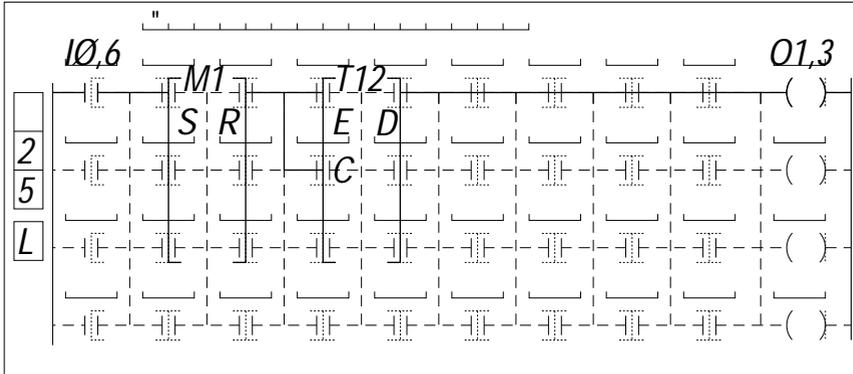
- Les blocs opérations se positionnent toujours en zone action. Ils utilisent 1 ligne et 3 colonnes et sont toujours directement reliés à la barre de potentiel droite.



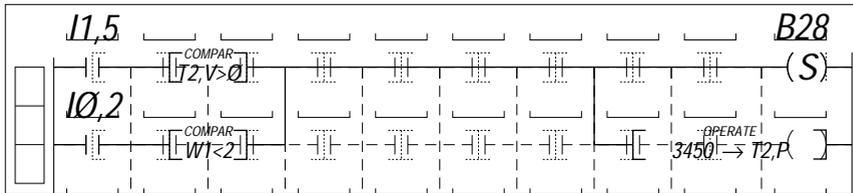
• Les blocs fonctions sont "cascadables"

Comme pour les éléments graphiques du type contacts, il est possible de réaliser des combinaisons de blocs fonctions.

Mise en série de blocs fonctions :



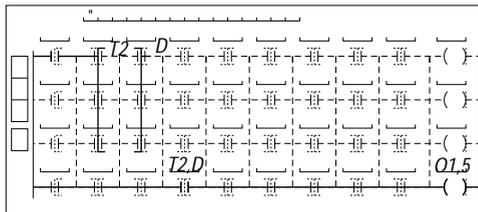
• Les blocs fonctions et blocs opérations sont mixables



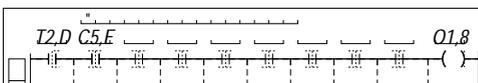
• Autres possibilités des blocs fonctions

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

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



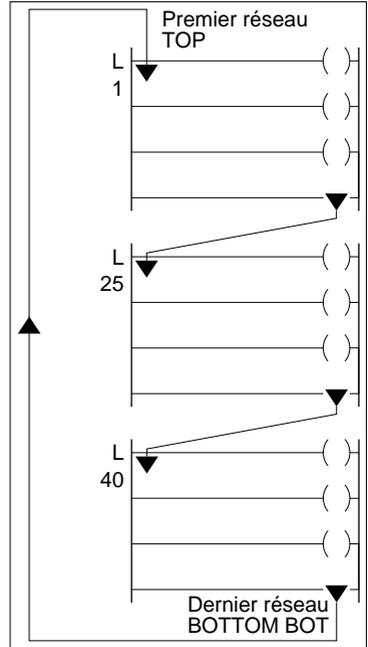
- sorties testables : les sorties des blocs fonctions sont accessibles à l'utilisateur sous forme d'objet bit.



3.2 Règles d'exécution d'un réseau de contacts

3.2-1 Principe de scrutation d'un programme

Le programme est exécuté réseau par réseau de manière séquentielle selon l'ordre d'écriture des réseaux en mémoire. La lecture se fait de haut en bas sauf lors de l'exécution d'un saut.

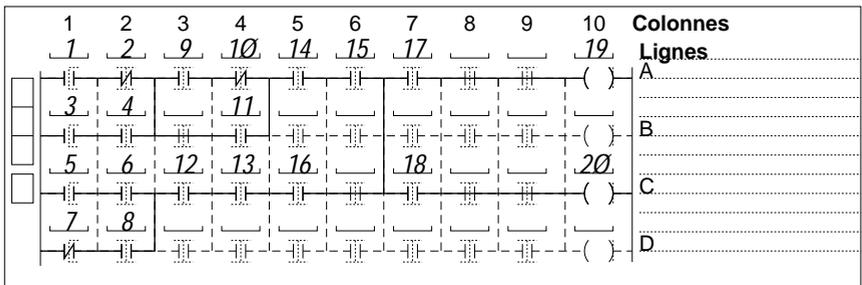


3.2-2 Principe d'exécution d'un réseau

L'exécution d'un réseau s'effectue séquence par séquence, du haut vers le bas, de gauche à droite. Une séquence est délimitée par une connexion verticale ou un bloc fonction.

Exemple : L'ordre d'exécution des séquences de ce réseau est le suivant :

(1-2), (3-4), (5-6), (7-8), (9-10), 11, (12-13), (14-15), 16, 17, 18, 19 et 20.



Les bobines sont exécutées après le traitement de la zone test.

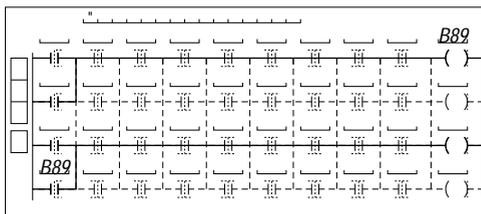
En respectant cet ordre d'exécution, le système met à jour :

- l'état logique de chaque contact en fonction de l'état de l'entrée associée, acquise en début de cycle,
- l'état logique des objets bits associés aux bobines afin de mettre à jour les sorties en fin de cycle.

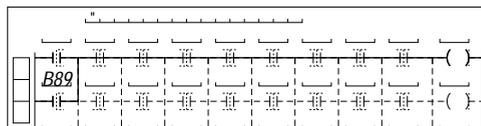
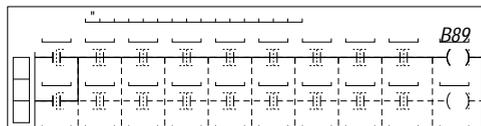
Conséquences : lorsqu'un objet est utilisé plusieurs fois dans un même réseau, son état logique est, selon le cas, celui acquis pendant le cycle en cours ou celui acquis dans le cycle précédent.

Dans l'exemple suivant B89 est utilisé deux fois dans le même réseau.

La valeur du contact B89 n'est pas celle donnée par la bobine 89 dans le cycle en cours mais celle acquise au cycle précédent.



Pour que la bobine et le contact aient la même valeur dans un cycle, il faut décomposer le réseau précédent en deux réseaux distincts.

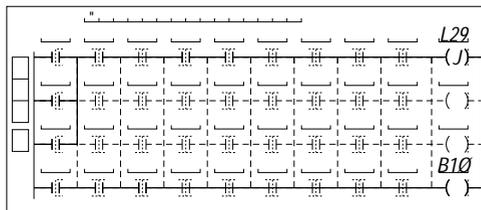


Remarque : une bobine ne doit jamais être écrite plusieurs fois dans un programme. En effet, son état à la fin d'un cycle sera défini par la dernière écriture.

3.2-3 Saut à un autre réseau (JUMP)

Lors du traitement de la zone action, 2 cas sont possibles :

- la bobine "J" est activée, le saut au réseau concerné est immédiat, la bobine B 10 n'est pas activée,
- la bobine "J" n'est pas activée : la bobine B10 est traitée normalement.

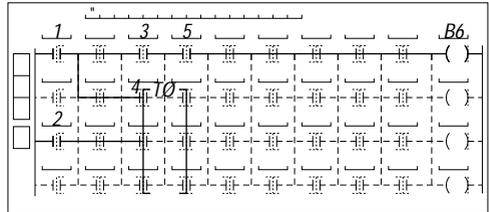


Pour qu'une bobine soit traitée indépendamment de l'état d'une bobine "J" située dans le même réseau, il convient donc de la placer sur une ligne située au dessus de la bobine "J".

3.2-4 Exécution d'un réseau comportant des blocs fonctions

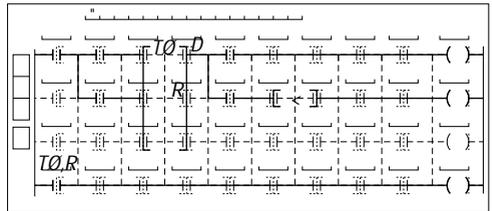
Les fonctions associées à ce type de bloc sont exécutées au moment de la scrutation du coin supérieur gauche du bloc concerné.

Le bloc fonction délimite aussi une séquence (voir page 3/9).

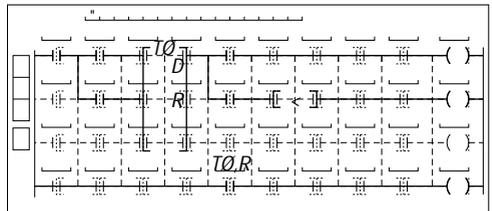


Conséquence : lorsque le bloc fonction et l'un des objets bits qui lui sont associés sont utilisés dans le même réseau, cet objet peut prendre la valeur donnée par le bloc dans le cycle précédent ou dans le cycle en cours :

L'état de l'objet bit T0,R est celui acquis dans le cycle précédent.



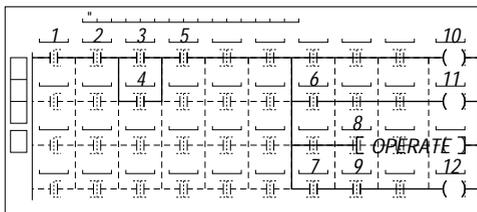
L'état de l'objet bit T0,R est celui acquis dans le cycle en cours.



3.2-5 Scrutation d'un réseau comportant des blocs opérations

Le bloc opération, situé en zone action, suit les mêmes règles que les blocs fonctions.

Exemple : Ce réseau s'exécute dans l'ordre présenté ci-contre.



L'opération associée au bloc n'est lancée qu'au moment de la scrutation de la partie gauche du bloc.

Tout bloc opération est donc exécuté :

- avant toutes les bobines (colonne 10),
- avant les contacts des colonnes 9 lorsque ceux-ci sont situés au-dessus du bloc,
- avant les contacts des colonnes 8 et 9 lorsque ceux-ci sont situés en-dessous du bloc.

Rappel : En structure bitâche, dans tous les cas, le système finit la scrutation du réseau en cours avant de démarrer l'exécution de la tâche rapide.

L'apparition d'un "appel horloge" ou d'un événement prioritaire sur l'exécution de la tâche en cours n'a donc pas d'incidence sur les principes énoncés dans les paragraphes précédents.



Sous chapitre	Page
4.1 Bloc fonction : Temporisateur	4/3
4.1-1 Présentation	4/3
4.1-2 Caractéristiques	4/3
4.1-3 Fonctionnement	4/4
4.1-4 Cas types	4/5
4.2 Bloc fonction : Monostable	4/9
4.2-1 Présentation	4/9
4.2-2 Caractéristiques	4/9
4.2-3 Fonctionnement	4/10
4.3 Bloc fonction : Compteur	4/12
4.3-1 Présentation	4/12
4.3-2 Caractéristiques	4/12
4.3-3 Fonctionnement	4/13
4.4 Bloc fonction : Compteur/Temporisateur rapide (TSX 17-20)	4/15
4.4-1 Présentation	4/15
4.4-2 Caractéristiques	4/15
4.4-3 Fonctionnement	4/16
4.4-4 Utilisation de FC en compteur/temporisateur MULTIPRESELECTION	4/17
4.4-5 Raccordements du compteur rapide	4/18
4.5 Bloc fonction : Registre	4/20
4.5-1 Présentation	4/20
4.5-2 Caractéristiques	4/20
4.5-3 Fonctionnement	4/21

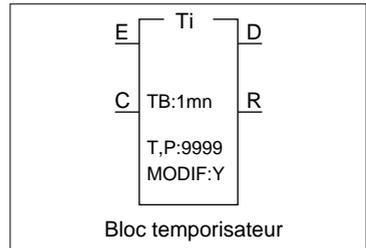
Sous chapitre	Page
4.6 Bloc fonction : Programmateur cyclique	4/23
4.6-1 Présentation	4/23
4.6-2 Caractéristiques	4/23
4.6-3 Fonctionnement	4/24
4.7 Bloc fonction : comparateur	4/26
4.7-1 Présentation	4/26
4.7-2 Caractéristiques	4/26
Ce chapitre se termine à la page	4/26

4.1 Bloc fonction : Temporisateur

4.1-1 Présentation

Le bloc fonction temporisateur permet de commander avec retard des actions spécifiques.

La valeur de ce retard est programmable et peut être modifiable ou non en mode REGLAGE.



4.1-2 Caractéristiques

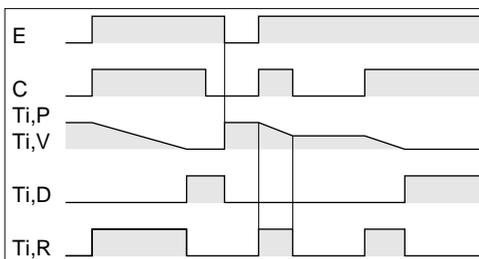
Numéro temporisateur	R (Running)	TSX 17-20	TSX 27/47
Nombre		32 (T0 à T31)	16 (T0 à T15)
Base de temps	TB	1mn, 1s, 100ms, 10ms (1mn par défaut) Plus la base de temps est faible, plus la précision du temporisateur sera grande.	
Valeur courante	Ti,V	Mot qui décroît de Ti,P vers 0 sur écoulement du temporisateur. Peut être lu, testé, mais non écrit ⁽¹⁾ .	
Valeur de présélection	Ti,P	0 < Ti,P < 9999. Mot qui peut être lu, testé, et écrit par programme. Est mis à la valeur 9999 par défaut	
Modification MODIF	YES/NO	YES : possibilité de modification de la valeur de présélection Ti,P en mode REGLAGE. NO : pas d'accès en mode réglage.	
Entrée "Armement"	E (Enable)	Sur état 0 réinitialise le temporisateur Ti,V = Ti,P	
Entrée "Contrôle"	C (Control)	Sur état 0 gèle la valeur courante Ti,V. Doit être à 1 durant l'écoulement.	
Sortie "Temporisateur écoulé"	D (Done)	Le bit associé Ti,D = 1, si temporisation écoulée Ti,V = 0	
Sortie "Temporisateur en cours"	R(Running)	Le bit associé Ti,R = 1 si temporisateur Ti,P > Ti,V > 0 et si entrée C est à l'état 1	
Dimension dans un réseau de contacts		2 colonnes de contacts sur 3 lignes	

⁽¹⁾ Ti,V peut être modifiée par terminal en mode REGLAGE.

4.1-3 Fonctionnement

Le temporisateur évolue lorsque ses 2 entrées (E et C) sont à l'état 1. Il se comporte comme un décompteur.

- la valeur courante Ti,V décroît de la présélection Ti,P vers 0, d'une unité à chaque impulsion de la base de temps TB.
- le bit de sortie Ti,R (Temporisateur en cours) associé à la sortie R est alors à l'état 1, le bit de sortie Ti,D (Temporisateur écoulé) associé à la sortie D est à l'état 0.
- lorsque la valeur courante $Ti,V = 0$, Ti,D passe à l'état 1 et Ti,R repasse à l'état 0.



E	0	0	1	1
C	0	1	0	1
Ti,P Ti,V	$Ti,V = Ti,P$	$Ti,V = Ti,P$	Ti,V gelée	Ti,V décroît de $Ti,P \rightarrow 0$
Ti,D	0	0	0	1 si Tempo écoulée
Ti,R	0	0	0	1 si Tempo en cours

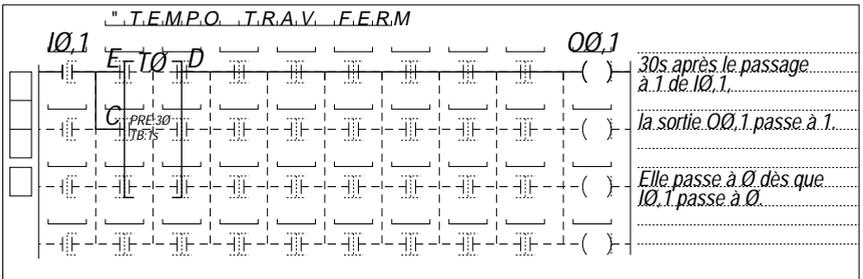
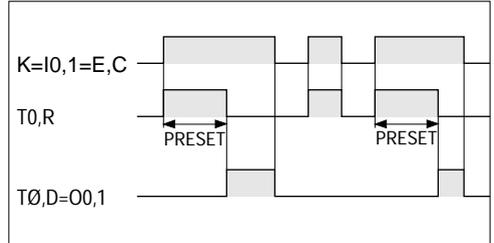
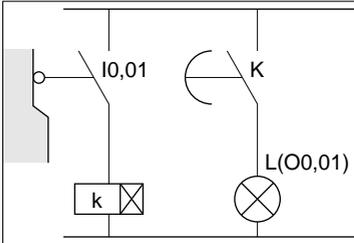
Cas spécifiques

- **Incidence d'une reprise "à froid"** : (SY0=1) provoque le chargement de la valeur de présélection (définie en mode CONFIGURATION) dans la valeur courante et la mise à 0 de la sortie Ti,D , la valeur de présélection éventuellement modifiée par le terminal (mode REGLAGE ou DONNEES) étant perdue,
- **Incidence d'une "reprise à chaud"** : (SY1) n'a pas d'incidence sur la valeur courante du temporisateur,
- **Incidence d'un passage en stop** : le passage en stop de l'automate ne fige pas la valeur courante. Il en va de même lorsque la tâche en cours est désactivée (structure bitâche) ou lors de l'exécution d'un point d'arrêt,
- **Incidence d'un saut de programme** : le fait de ne pas scruter le réseau où est programmé le bloc temporisateur ne fige pas la valeur courante Ti,V qui continue à décroître vers 0,
De même les bits Ti,D et Ti,R associés aux sorties D et R du bloc temporisateur conservent leur fonctionnement normal et peuvent être ainsi testés dans un autre réseau.
Par contre les bobines directement "raccordées" aux sorties du bloc ne seront pas activées puisque non scrutées par l'automate,
- **Test des bits Ti,D et Ti,R** : ces bits peuvent changer d'état en cours de cycle (voir gestion des bases de temps intercalaire A, page 4/6).

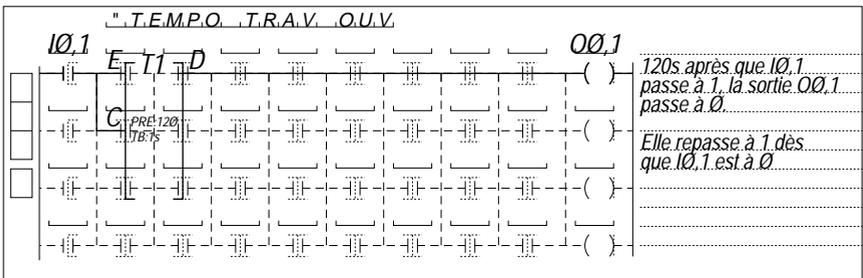
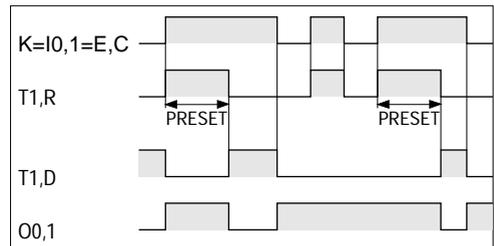
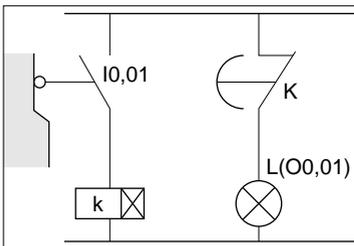
4.1-4 Cas types

Selon sa programmation, le bloc fonction "Temporisateur" peut réaliser les fonctions suivantes :

• Temporisateur Travail à Fermeture

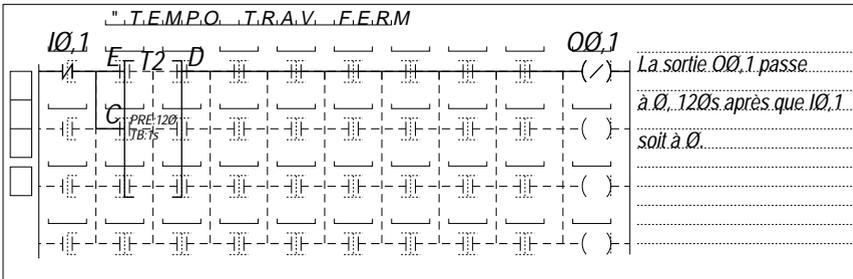
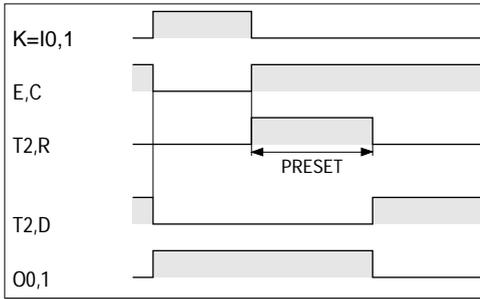
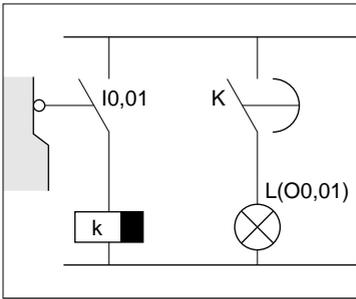


• Temporisateur Travail à Ouverture

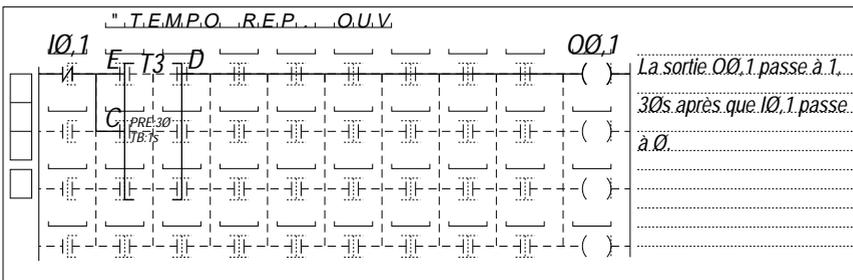
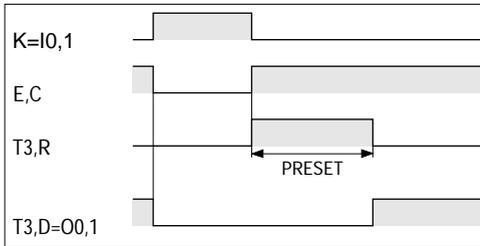
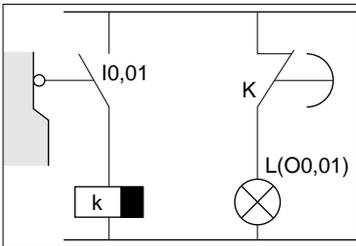


B

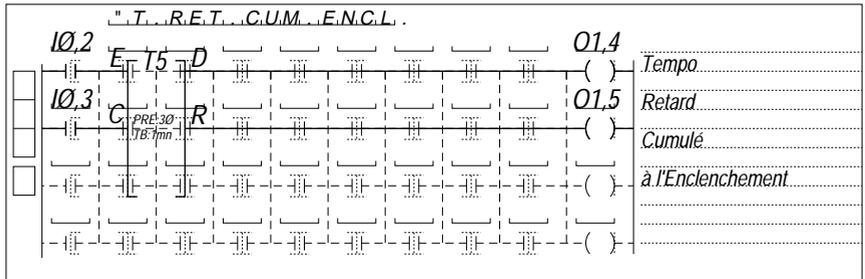
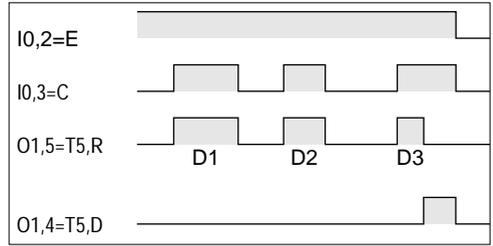
• Temporisateur Repos à Fermeture



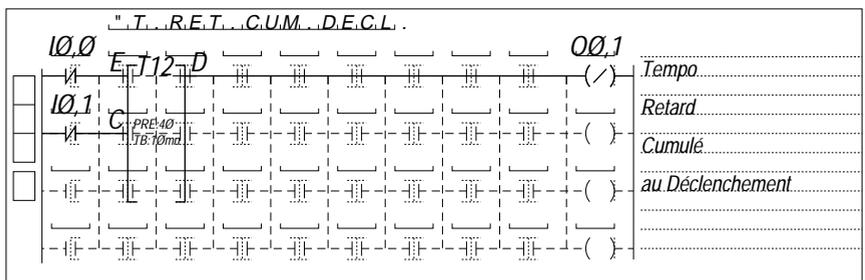
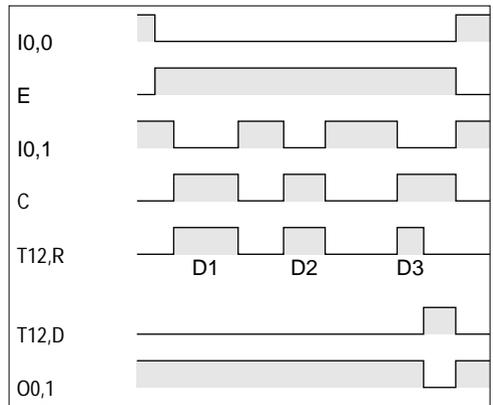
• Temporisateur Repos à Ouverture



• Temporisateur Travail à Fermeture : retard cumulé à l'enclenchement



• Temporisateur Repos à Fermeture : retard cumulé au déclenchement



Exemple d'utilisation

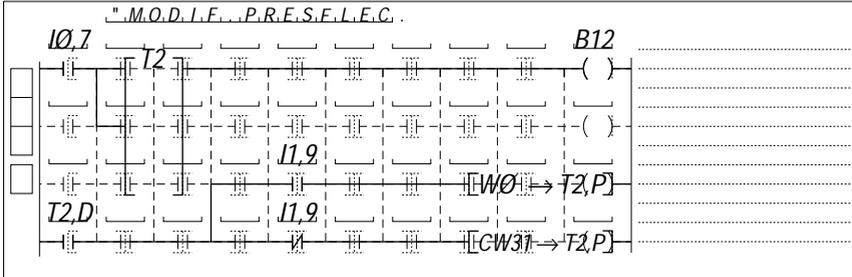
• **Modification de présélection en fonction d'un événement extérieur :**

La connaissance des blocs opérations "comparaison" et "transfert" est nécessaire à la compréhension de cet exemple.

Si T2,D=1 et I1,9=1 alors ranger le contenu du mot interne W0 dans T2,P.

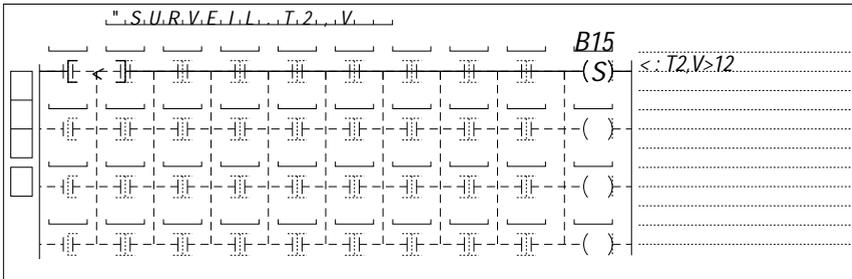
Si T2,D=1 et I1,9=0 alors ranger le contenu du mot constant CW31 dans T2,P.

Nota : la modification ne s'effectue que si le temporisateur n'est pas en cours T2,D=1 et T2,R=0.



• **Surveillance de la valeur courante d'un temporisateur :**

Si T2,V>12s, alors mettre le bit interne B15 à l'état 1.

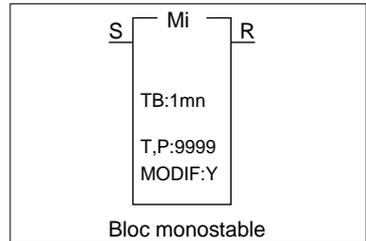


4.2 Bloc fonction : Monostable

4.2-1 Présentation

Le bloc fonction monostable permet d'élaborer une impulsion de durée précise.

Cette durée est programmable et peut être modifiable ou non en mode REGLAGE par les terminaux.



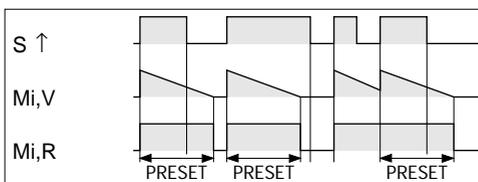
4.2-2 Caractéristiques

Numéro monostable	Mi	TSX 17-20	TSX 27/47
Nombre		8	8
Base de temps	TB	1mn, 1s, 100ms, 10ms (1mn par défaut)	
Valeur courante	Mi,V	Mot qui décroît de Mi,P vers 0 sur écoulement du monostable. Peut être lu, testé mais non écrit.	
Valeur de présélection	Mi,P	0 < Mi,P < 9999, mot pouvant être lu, testé, écrit.	
Modification MODIF	YES/NO	YES : possibilité de modification de la valeur de présélection en mode REGLAGE NO : pas d'accès en mode réglage	
Entrée "départ"	S(start)	Sur front montant Mi,V=Mi,P puis Mi,V décroît vers 0.	
Sortie "monostable en cours"	R(Running)	Le bit associé Mi,R est à 1 si Mi,V > 0 (écoulement du monostable). Mi,R=0 si Mi,V=0	
Dimension dans un réseau de contacts		2 colonnes de contacts sur 3 lignes.	

4.2-3 Fonctionnement

Lors de l'apparition d'un état 1 (front montant) sur l'entrée S du monostable

- la valeur courante Mi,V prend la valeur de présélection Mi,P et décroît vers 0 d'une unité à chaque impulsion de la base de temps TB .



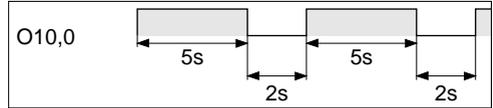
- Le bit de sortie Mi,R (running) associé à la sortie R passe à l'état 1 dès que la valeur courante Mi,V est différente de 0.
- Lorsque la valeur courante $Mi,V=0$, le bit de sortie Mi,R repasse à l'état 0.

Cas spécifiques

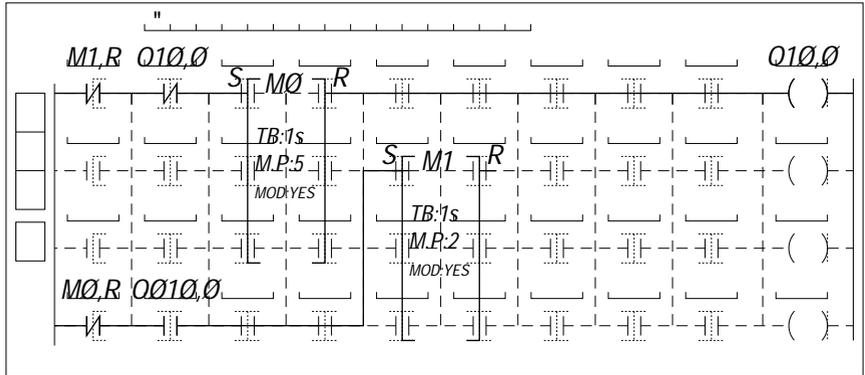
- Incidence d'une reprise à froid** : ($SY0=1$) provoque le chargement de la valeur de présélection (définie en programmation) dans la valeur courante Mi,V ; la valeur de présélection éventuellement modifiée par le terminal (mode REGLAGE ou DONNEES) étant perdue.
- Incidence d'une reprise à chaud** : ($SY1$) n'a pas d'incidence sur la valeur courante Mi,V du monostable.
- Incidence d'un passage en stop : le passage en STOP de l'automate ne fige pas la valeur courante. Il en va de même lorsque la tâche en cours est désactivée (structure bitâche) ou lors de l'exécution d'un point d'arrêt.
- Incidence d'un saut de programme** : le fait de ne plus scruter le réseau où est programmé le bloc monostable ne fige pas la valeur courante Mi,V qui continue de décroître vers 0.
De même le bit Mi,R associé à la sortie du bloc monostable conserve son fonctionnement normal et peut être ainsi testé dans un autre réseau; par contre les bobines directement "raccordées" à la sortie du bloc (Ex:O10,0) ne seront pas activées car non scrutées par l'automate.
- Test du bit Mi,R** : ce bit peut changer d'état en cours de cycle (voir gestion des bases de temps intercalaire A, page 4/6).

- **Exemple d'utilisation** : clignotant à période cyclique variable.

La valeur de présélection de chaque monostable définit la durée de chaque impulsion.



Dans l'exemple ci-dessous, la sortie O10,0 est mise à l'état 1 pendant 5s (M0,P) et remise à l'état 0 pendant 2s (M1,P)

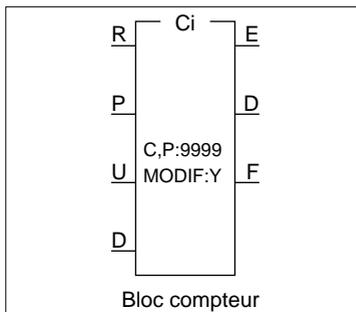


B

4.3 Bloc fonction : Compteur

4.3-1 Présentation

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



4.3-2 Caractéristiques

Numéro de compteur	Ci	TSX 17-20	TSX 27/47
Nombre		31 (C0 à C30)	16 (C0 à C15)
Valeur courante	Ci,V	Mot incrémenté ou décrémenté en fonction des entrées U et D. Peut être lu, testé mais non écrit.	
Valeur de préselection	Ci,P	0 < Ci,P < 9999. Mot pouvant être lu, testé, écrit.	
Modification MODIF	YES/NO	YES: possibilité de modification de la valeur de préselection en mode REGLAGE. NO : pas d'accès en mode REGLAGE.	
Entrée remise à zéro	R (Reset)	Sur état 1 : Ci,V = 0.	
Entrée préselection	P (Preset)	Sur état 1: Ci,V = Ci, P.	
Entrée comptage	U (Up)	Incrémente Ci,V sur front montant.	
Entrée décomptage	D (Down)	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.	
Sortie préselection	D (Done)	Le bit associé Ci,D=1, lorsque Ci,V=Ci,P. atteinte	
Sortie débordement	F (Full)	Le bit associé Ci,F =1 lorsque Ci,V passe de 9999 à 0.	
Dimension à l'écran		2 colonnes de contacts sur 4 lignes	

4.3-3 Fonctionnement.

- **Présélection** : si l'entrée "présélection" est à l'état 1 et l'entrée "remise à zéro" à l'état 0, la valeur courante $C_{i,V}$ prend la valeur $C_{i,P}$ et la sortie $C_{i,D}$ prend la valeur 1.
- **Remise à zéro** : dès la mise à l'état 1 de cette entrée, la valeur courante $C_{i,V}$ est forcée à 0, les sorties $C_{i,E}$, $C_{i,D}$ et $C_{i,F}$ sont à 0. L'entrée "remise à zéro" est prioritaire.
- **Comptage** : à l'apparition d'un front montant sur l'entrée comptage U, la valeur courante est incrémentée d'une unité. lorsque cette valeur est égale à la valeur de présélection $C_{i,P}$, le bit de sortie $C_{i,D}$ "présélection atteinte" associé à la sortie D passe à l'état 1. le bit de sortie $C_{i,F}$ (débordement comptage) passe à l'état 1 pendant un tour de programme lorsque $C_{i,V}$ passe de 0 à 9999.
- **Décomptage** : à l'apparition d'un front montant sur l'entrée "décomptage" D, la valeur courante $C_{i,V}$ est décrémentée d'une unité. Le bit de sortie $C_{i,E}$ (débordement décomptage) passe à l'état 1 pendant un tour de programme lorsque $C_{i,V}$ passe de 0 à 9999.
- **Comptage/Décomptage** : pour utiliser simultanément les fonctions comptage et décomptage, il est nécessaire de commander les deux entrées correspondantes U et D; ces deux entrées étant scrutées successivement. Si les deux entrées sont à 1 simultanément, la valeur courante reste inchangée.

Cas spécifiques.

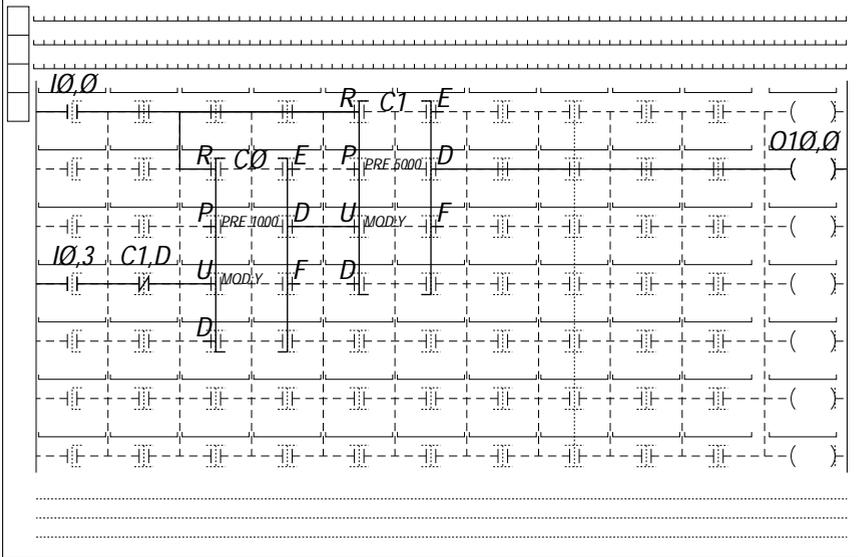
- **Incidence d'une "reprise à froid"** : ($SY0=1$) provoque la mise à zéro de la valeur courante $C_{i,V}$. La valeur de présélection éventuellement modifiée par le terminal (mode REGLAGE ou DONNEES) étant perdue. Les bits de sorties $C_{i,E}$, $C_{i,D}$ et $C_{i,F}$ sont à 0.
- **Incidence d'une reprise à chaud (SY1) ou d'un passage en STOP** : n'a pas d'incidence sur la valeur courante du compteur ($C_{i,V}$).

Exemples d'utilisation.

- Comptage d'un nombre de pièces > 9999.
Deux blocs fonctions compteur montés en cascade sont nécessaires. Chaque impulsion sur l'entrée I0,3 provoque l'incrémentation du compteur C0 et ce jusqu'à la valeur de présélection final du compteur C1 (bit C1,D=1).
La remise à zéro des compteurs est provoquée par l'entrée I0,0.

Formulaire programmation schéma à contacts

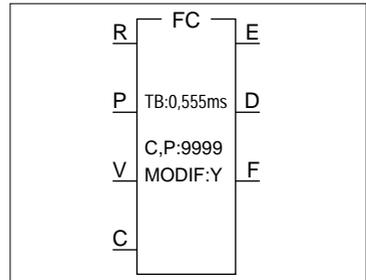
<input type="checkbox"/> IT	<input type="checkbox"/> FAST	<input type="checkbox"/> MAST	<input type="checkbox"/> AUX	<input type="checkbox"/> ASY
<input type="checkbox"/> MAIN	<input type="checkbox"/> SR	<input type="checkbox"/> PRE	<input type="checkbox"/> POST	<input type="checkbox"/> CHART
				<input type="checkbox"/> XM



4.4 Bloc fonction : Compteur/Temporisateur rapide (TSX 17-20 uniquement)

4.4-1 Présentation

Les automates TSX 17-20 possèdent un compteur rapide FC, qui selon sa configuration logicielle peut être utilisé en compteur rapide ou en temporisateur rapide. La valeur de présélection est définie en mode PROGRAMMATION (PRESET statique) et peut être modifiée par programme utilisateur⁽¹⁾ ou avec le terminal en mode REGLAGE (PRESET dynamique).



4.4-2 Caractéristiques

Compteur rapide (configuration par défaut)

Sa fréquence maximum est de 2 KHz, avec un cycle de comptage de 10 000 points. Le Compteur FC reçoit les signaux nécessaires à son fonctionnement par un connecteur spécialisé, qui comprend :

- 2 entrées physiques de comptage : 5 ou 24 V,
- 2 entrées physiques de remise à zéro du compteur : 5 ou 24 V.

Temporisateur rapide

Une base de temps interne de 0,555 ms remplace les entrées physiques de comptage, qui deviennent inopérantes. Un front montant sur l'entrée physique de remise à zéro provoque la mise à 0 de la valeur courante FC,V.

Compteur/temporisateur FC (C31 en mode REGLAGE)

Base de temps	TB	0,555 ms (si FC configuré en tempo)
Valeur courante capturée	FC,V	0 < FC,V < FC,P valeur courante capturée par lecture à la volée de la valeur courante interne si l'entrée V est à l'état logique 1. Peut être lu, testé, mais non écrit.
Valeur de présélection	FC,P	0 < FC,P < 9999. Mot qui peut être lu, testé, et écrit par programme. Est à la valeur 9999 par défaut.
Modification MODIF	YES/NO	YES : possibilité de modification de la valeur de présélection FC,P de C31 en mode REGLAGE. NO : pas d'accès en mode réglage.
Entrée "Remise à Zéro"	R (RESET)	Sur état 1 positionne FC,V à 0 Entrée prioritaire.

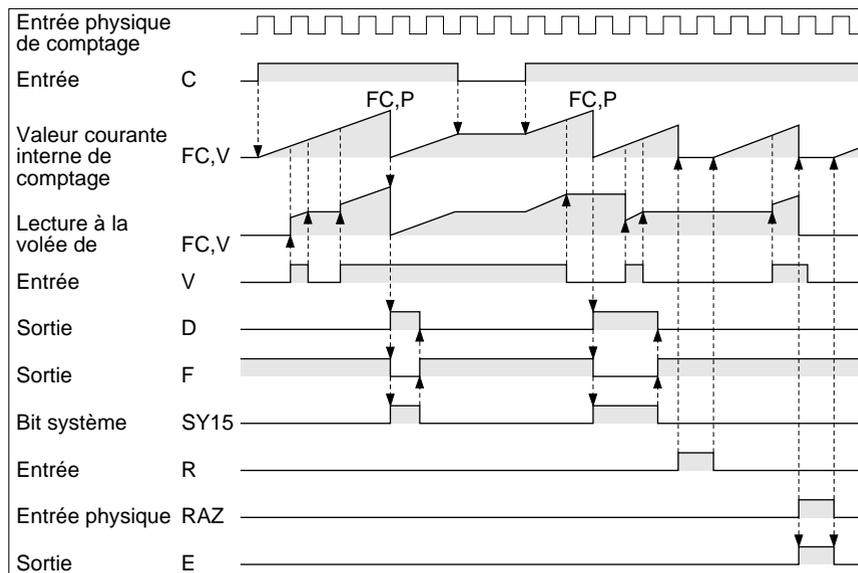
⁽¹⁾ utilisation de FC en compteur/temporisateur multiprésélection.

Caractéristiques (suite)

Entrée PRESELECTION ⁽¹⁾	P (PRESET)	Sur état 1 chargement du PRESET dynamique (registre d'attente).
Entrée "Lecture à la volée"	V	Sur état 1, lecture à la volée de la valeur courante FC,V.
Entrée "Validation comptage"	C	Sur état 1, valide le comptage (évolution de FC,V).
Sortie "Image entrées physiques RAZ"	E (External RELOAD)	Le bit associé FC,E = 1 si une RAZ physique est provoquée.
Sortie "Présélection atteinte"	D (DONE)	Le bit associé FC,D = 1 si FC,V=FC,P Ce bit reste à 1 jusqu'à la mise à 0 de SY15.
Sortie "Présélection non atteinte"	F	Le bit associé FC,F = 1 si $0 < FC,V < FC,P$.
Dimension		2 colonnes de contacts sur 4 lignes dans un réseau de contacts

4.4-3 Fonctionnement

Diagramme

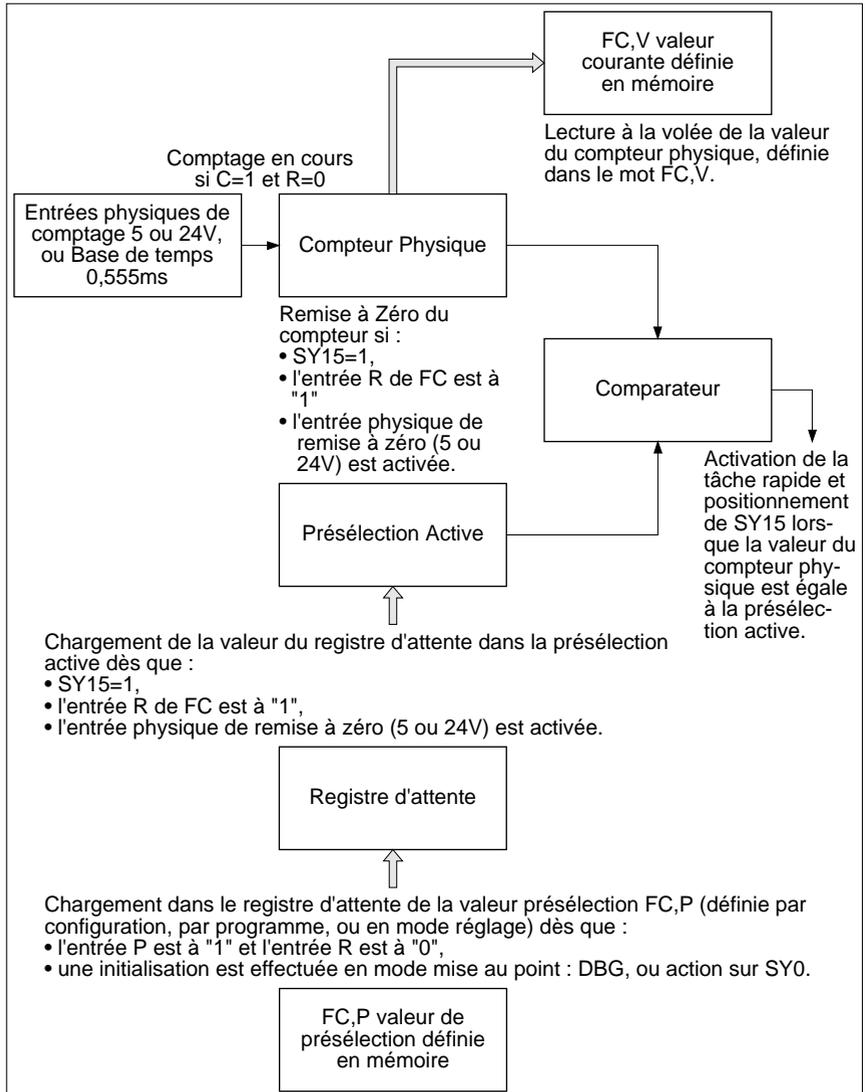


Bit système SY15

Lorsque FC atteint la valeur de présélection active (FC,P = FC,V) :

- Le bit SY15 passe à 1.
- La tâche rapide est lancée (si SY19=0). Le bit SY15 doit être remis à 0 par programme, dans la tâche rapide.
- Le compteur est remis à 0.
- La valeur présente dans le registre d'attente est recopiée dans la présélection active.

Architecture



La mise à zéro de FC par une entrée physique "RAZ" ou par l'entrée R, provoque la mise à zéro de la valeur courante interne du compteur physique et de la valeur courante capturée FC,V (si l'entrée V=1).

La valeur de présélection dynamique est conservée.

Le bloc fonction FC décrit précédemment, ne réalise pas la fonction de comptage. Il effectue uniquement le contrôle des éléments physiques (présélection active, registre d'attente, etc). La fonction comptage est assurée par le compteur physique, dont le fonctionnement est indépendant du cycle de l'automate.

4.4-4 Utilisation de FC en compteur/temporisateur MULTIPRESELECTION

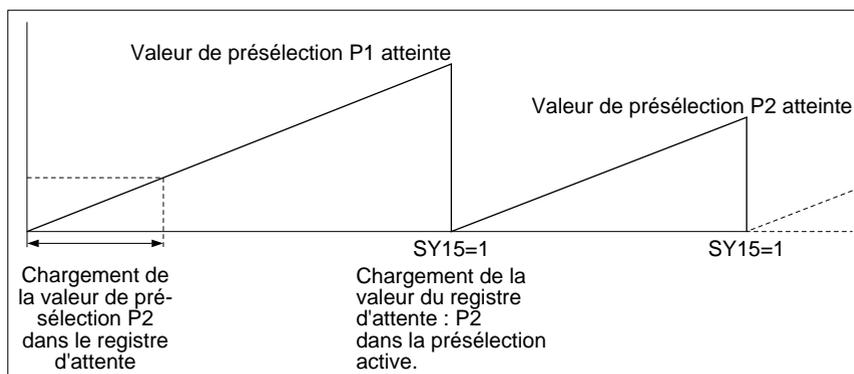
Rôle de l'entrée P : La mise à 1 de cette entrée permet le chargement par le programme d'une valeur de présélection (PRESET dynamique). Le PRESET statique étant la valeur définie par configuration, en mode PROGRAMMATION.

Pour charger une nouvelle valeur de présélection par programme, il suffit de transférer (en utilisant un bloc opération) le contenu d'un mot Wi ou CWi ($0 < i < 127$) dans le mot FC,P. Cette opération doit être programmée dans le réseau de contacts précédant celui contenant le bloc fonction FC.

La modification de la valeur de présélection par transfert d'un mot dans FC,P (P2, voir figure ci-dessous), n'est prise en compte qu'une fois que le compteur a atteint la valeur de présélection initialement définie (P1).

Cette fonctionnalité du compteur/temporisateur rapide permet la multiprésélection.

Le retour au PRESET statique se fera par chargement de la valeur équivalente ou action sur SY0.



4-4.5 Raccordement du compteur rapide

Le connecteur qui transmet les signaux de comptages et de remise à zéro au compteur physique est situé sur la face avant de l'automate de base TSX 17-20, en haut à gauche.

Ce connecteur, raccordé à un câble (référence: TSX CCB 020), possède 2 entrées isolées, sans point commun :

- **Entrée physique de comptage :**

0V = broche 1 (fil Bleu)

5V = broche 7 (fil Noir)

24V= broche 6 (fil Gris)

- **Entrée physique de remise à zéro :**

0V = broche 5 (fil Jaune)

5V = broche 8 (fil Orange)

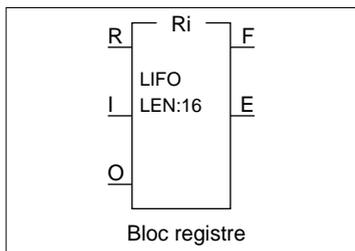
24V= broche 9 (fil Blanc)

4.5 Bloc fonction : Registre

4.5-1 Présentation

Un registre est un ensemble de mots de 16 bits permettant de stocker des informations de deux manières différentes :

- file d'attente (premier entré, premier sorti) ou pile FIFO,
- pile (dernier entré, premier sorti) ou pile LIFO.



4.5-2 Caractéristiques

Numéro Registre	Ri	TSX 17-20 4 (R0 à R3)	TSX 27/47 4 (R0 à R3)
Type	FIFO LIFO	File d'attente (First In, First Out). Pile (Last In, First Oust)	
Longueur	L	Nombre de mots 16 bits $1 \leq L \leq 255$. (16 par défaut)	
Mot d'entrée	Ri,I	Mot d'accès au registre. Peut être lu, testé, écrit.	
Mot de sortie	Ri,O	Mot de sortie du registre. Peut être lu, testé, écrit	
Entrée "Stockage"	I (In)	Sur front montant provoque le stockage du contenu du mot Ri,I dans le registre.	
Entrée "Déstockage"	O (Out)	Sur front montant provoque le rangement d'un mot d'information dans le mot Ri,O.	
Entrée "Remise à zéro"	R (Reset)	Sur état 1 initialise le registre.	
Sortie "Plein"	F (Full)	Le bit Ri,F associé indique que le registre est plein. Peut être testé.	
Sortie "Vide"	E (Empty)	Le bit Ri,E associé indique que le registre est vide. Peut être testé.	
Dimensions à l'écran		2 colonnes de contacts sur 3 lignes	

4.5-3 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), le contenu du mot d'entrée Ri,I préalablement chargé est stocké au plus haut de la file (fig a et b).

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

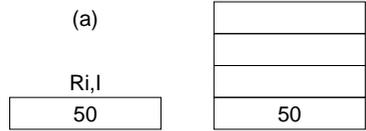
Lorsqu'une demande de sortie est prise en compte (front montant sur entrée O) stockage du contenu de Ri,I le mot d'information le plus bas de la file est rangé dans le mot de sortie Ri,O et le contenu du registre est décalé d'un pas vers le bas (fig. c).

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.

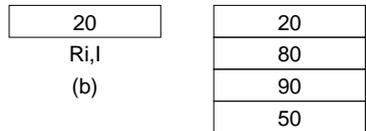
La file peut être réinitialisée à tout moment (état 1 sur l'entrée R).

Exemple :

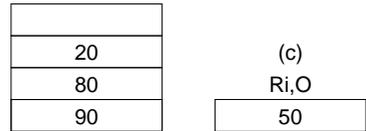
Stockage du contenu de Ri,I dans un registre vide.



Stockage du contenu de Ri,I au plus haut de la file.



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



LIFO (Last In, First Out)

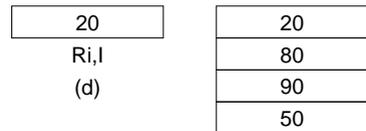
La dernière information entrée est la première à être sortie.

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

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

Exemple :

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

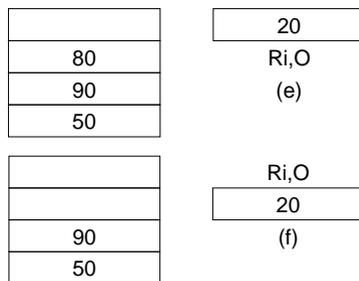


Lorsqu'une demande de sortie est prise en compte (front montant sur l'entrée O), le mot d'information le plus haut (dernière information entrée) est rangé dans le mot Ri,O (fig e et f).

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). L'élément pointé est alors le plus haut dans la pile.

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



Cas spécifiques

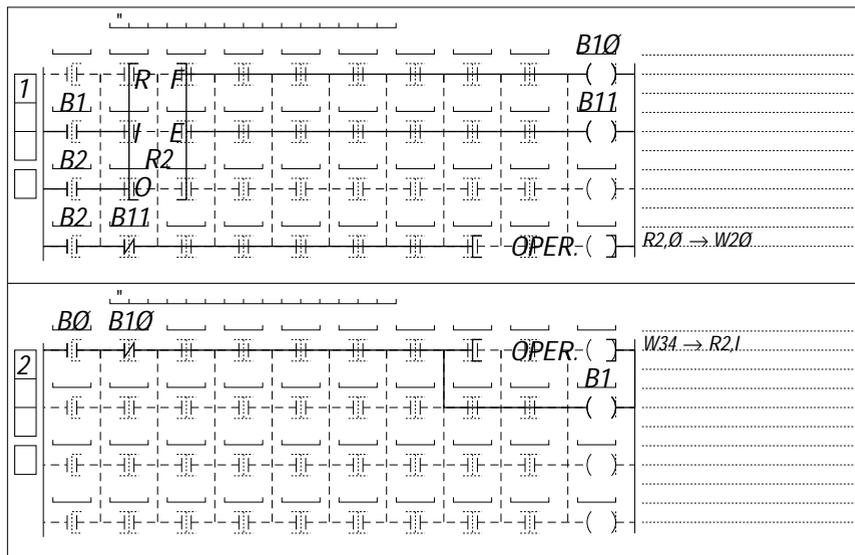
- Incidence d'une reprise "à froid" : (SY0=1) provoque l'initialisation du contenu du registre. Le bit de sortie Ri,E associé à la sortie E est mis à 1.
- Incidence d'une reprise "à chaud" ou "immédiate" : (SY1 ou SY2) n'a pas d'incidence sur le contenu du registre ainsi que sur l'état des bits de sorties.

Exemple

La connaissance des blocs opérations "calcul et transfert" est nécessaire pour la compréhension de cet exemple.

Le programme ci-dessous montre le chargement de R2,I par le mot W34 et la demande d'entrée B1 dans le registre R2 si l'événement B0 est présent et si le registre R2 n'est pas plein (B10=0).

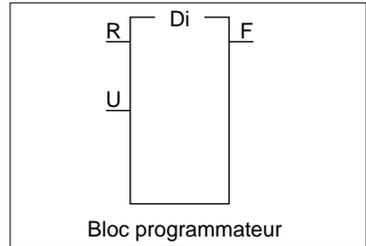
La demande de sortie est faite par le bit B2 et le rangement de R2,O dans W20 s'effectue si le registre n'est pas vide (B11=0).



4.6 Programmateur cyclique

4.6-1 Présentation

D'un principe de fonctionnement similaire au programmateur à cames, le programmateur 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 programmateur cyclique, ces points hauts seront symbolisés par un état 1 au niveau de chaque pas.



4.6-2 Caractéristiques

Numéro	Di	TSX 17-20	TSX 27/47
Nombre		8 (D0 à D7)	8 (D0 à D7)
Nombre de pas	L	1 < L < 16 (16 par défaut)	
Base de temps	TB	1mn, 1s, 100ms, 10ms. (1mn par défaut)	
Temps enveloppe ou durée du pas en cours	Di,V	0 < Di,V < 9999. Mot remis à zéro à chaque changement de pas. Peut être lu, testé mais non écrit.	
Numéro du pas en cours	Di,S	0 < Di,S < 15. Mot pouvant être lu et testé. Ne peut être écrit qu'à partir d'une valeur immédiate.	
Entrée "retour au pas 0"	R (RESET)	Sur état 1 initialise le programmateur au pas 0.	
Entrée "avance"	U (UP)	Sur front montant provoque l'avance d'un pas du programmateur et la mise à jour des bits d'ordres.	
Sortie	F (FULL)	Indique que le dernier pas défini est en cours. Le bit Di,F associé peut être testé (Di,F=1 si Di,S=L-1).	
Etat d'un pas	Di,Wj	Mot de 16 bits définissant les états du pas j du programmateur i. Peut être lu, testé mais non écrit.	

4.6-3 Fonctionnement

Le programmeur cyclique se compose:

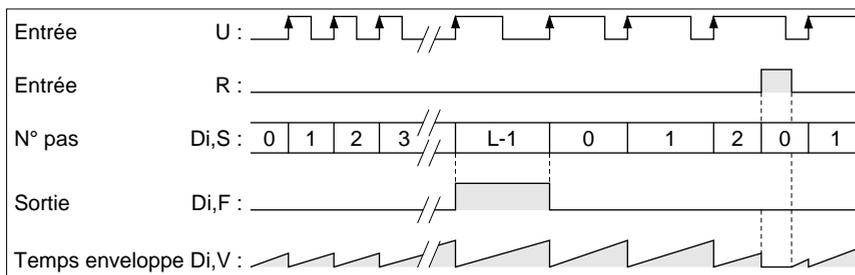
- d'une matrice de données constantes (les cames) organisée en 16 pas de 0 à 15 et de 16 informations binaires (état du pas) rangées en colonnes repérées de 0 à F.
- d'une liste de bits d'ordres (1 par colonne) correspondant à des sorties Ox,y ou à des bits internes Bi. Lors du pas en cours les bits d'ordres prennent les états binaires définis pour ce pas. dans l'exemple ci-dessous le pas 5 étant en cours, les bits d'ordre O3,1; O4,3 et O1,5 sont mis à l'état 1; les bits d'ordres O4,5; B5 et O3,0 sont mis à l'état 0.

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

Colonne	F	E	D	2	1	0	
Bits d'ordre :	O4,5	O3,1	O4,3	B5	O1,5	O3,8	
Configuration de 0 à 15 bits	Pas 0	0	0	1	1	0	= Di,W0
	Pas 1	1	0	1	1	0	= Di,W1
N° de pas en cours	Pas 5	0	1	1	0	1	Di,W5
	Pas 13	0	1	0	0	0	
Nombre de pas du programmeur	Pas 14	0	1	1	0	1	= Di,W14
	Pas 15	1	1	1	1	0	= Di,W15

Le numéro du pas en cours est incrémenté à chaque front montant sur l'entrée U. Ce numéro peut être modifié par programme.

Diagramme de fonctionnement



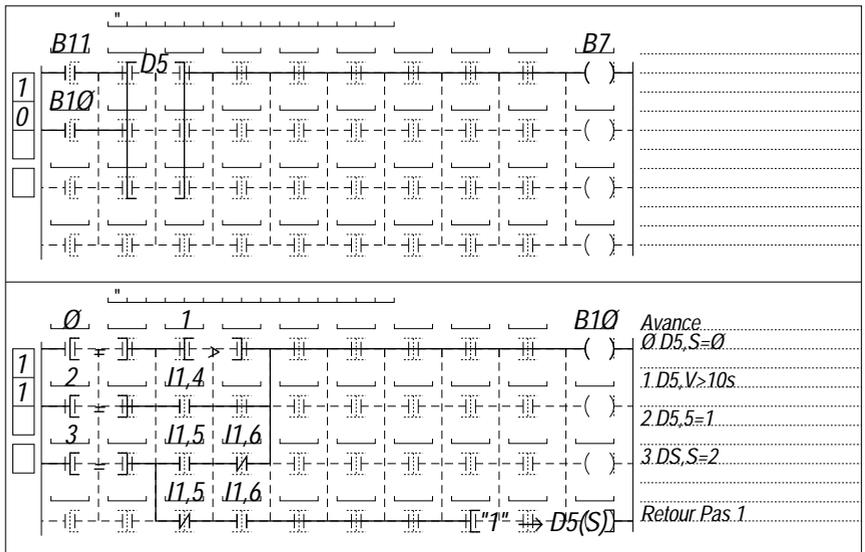
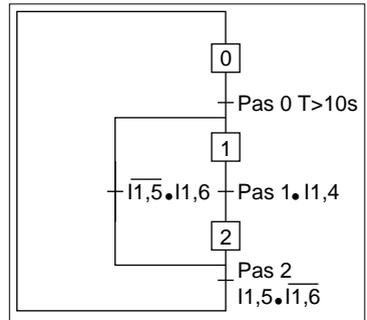
Cas spécifiques

- Incidence d'une "reprise à froid" : (SY0=1) provoque la réinitialisation du programmateur au pas 0.
- Incidence d'une "reprise à chaud" : (SY1) provoque la mise à jour des bits d'ordres.
- Incidence d'un saut de programme : le fait de ne pas scruter le réseau où est programmé le programmateur ne provoque pas de remise à 0 des bits d'ordres.
- Mise à jour des bits d'ordres: ne s'effectue que lors d'un changement de pas ou lors d'une reprise à chaud.

Exemple d'utilisation

- Commande d'avance sur évènement extérieur et temporisation, saut de pas.
La connaissance des blocs opérations "comparaison", "calcul et transfert" est nécessaire. Le programme ci-dessous traduit le grafcet en schéma à contacts.

Si le pas 0 du programmateur est en cours (C0) et si le temps enveloppe du pas est > à 10s (C1) passage au pas 1 (B10).
 Si le pas 1 est en cours (C2) et si I1,4 = 1: passage au pas 2 (B10)
 Si le pas 2 est en cours (C3), si I1,5 = 1 et I1,6 = 0: passage au pas 0 (la longueur du programmateur D5 (L) ayant été défini à 3 pas).
 Si le pas 2 est en cours (C3), I1,5 = 0 et I1,6 = 1 retour au pas 1.



4.7 Bloc fonction : comparateur

4.7-1 Présentation

Le bloc "comparaison" permet d'effectuer les tests suivants :

- supérieur >
- inférieur <
- égal =
- supérieur ou égal >=
- inférieur ou égal <=
- différent <>

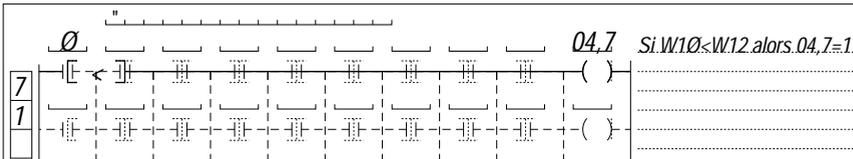


4.7-2 Caractéristiques

Nombre		TSX 17-20	TSX 27/47
		Non limités et programmables en zone test.	
Objet à comparer	OP1	Tout objet mot (opérande 1)	
Type de comparaison	SIGNE	>, <, >=, <=, <>.	
Valeur de comparaison	OP2	Tout objet mot ou valeur immédiate (opérande 2)	
Dimension à l'écran		2 colonnes de contacts sur 1 ligne.	

Exemple

L'exemple ci-dessous montre la comparaison entre deux mots





Sous chapitre	Page
5.1 Blocs opérations "calcul et transfert"	5/2
5.1-1 Présentation	5/2
5.1-2 Caractéristiques	5/2
5.2 Opérations sur chaînes de bits	5/3
5.2-1 Transfert d'une chaîne de bits dans une autre	5/3
5.2-2 Transfert d'une chaîne de bits dans un mot	5/4
5.2-3 Transfert d'un mot dans une chaîne de bits	5/5
5.2-4 Transfert d'une valeur immédiate dans une chaîne de bits	5/6
5.3 Opérations sur mots	5/7
5.3-1 Transfert	5/7
5.3-2 Décalage circulaire	5/10
5.3-3 Comparaison	5/12
5.3-4 Opérations arithmétiques	5/13
5.3-5 Opérations logiques	5/17
5.3-6 Transcodage	5/20
Ce chapitre se termine à la page	5/24

5.1 Bloc opération "calcul et transfert"

5.1-1 Présentation

Le bloc opération "calcul et transfert" permet d'effectuer les opérations suivantes :

- opérations arithmétiques +, -, *, / et REM sur TSX 17-20
- opérations logiques AND, OR, XOR,
- complément CPL
- transcodage de mot BIN, BCD, ATB, BTA
- décalage circulaire des bits SRC, SLC
- transfert



5.1-2 Caractéristiques

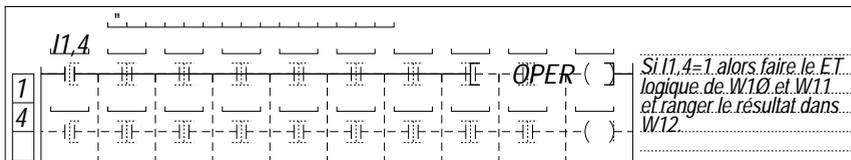
Nombre		TSX 17-20	TSX 27/47
		Non limités et programmables en zone action.	
Premier opérande	OP1	Tout objet mot	
Opération	OP	Code opération (+, -, *, /, ...)	
Deuxième opérande	OP2	Tout objet mot ou valeur immédiate.	
Résultat	→	Wi, Ti, P, OWxy, i, ...	
Dimension à l'écran		les 3 dernières colonnes d'une ligne.	

Utilisation des paramètres

	OP1	OP	OP2	→
Opérations arithmétiques et opérations logiques	X	X	X	X
Complément, transcodage de mot et décalage circulaire		X	X	X
Transfert			X	X

Exemple

L'exemple ci-dessous montre le ET logique de deux mot.



5.2 Opérations sur chaînes de bits

5.2-1 Transfert d'une chaîne de bits dans une autre.

Cette opération consiste à recopier l'état de n bits d'une chaîne origine dans les n bits d'une chaîne destination.

Structure

Le transfert est réalisé par un bloc opération constitué de la manière suivante :



Opérandes

Objet	Chaîne de bits		Exemple
	origine	destination	
Bit interne Bi	X	X	B14[16]
Bit d'entrée Ix,i	X	X ⁽¹⁾	I3,0[7]
Bit de sortie Ox,i	X	X	O5,0[3]

⁽¹⁾ voir règles d'utilisation

Exemple

Recopie du contenu des bits I0,0 à I0,5 dans les bits internes B20 à B25 :

OP2 I0,0[6]
→ B20[6]

Règles d'utilisation

- Longueur des chaînes de bits: le nombre de bits à transférer et le nombre de bits destinataires doivent être égaux et inférieur ou égal à 16.
- L'utilisation de chaîne de bits d'entrée comme destination est possible mais doit être employée avec précaution (modification de l'état réel des entrées jusqu'à la prochaine acquisition des entrées par l'automate).
- Transfert d'une chaîne de bits d'entrée/sorties à tout emplacement de module 4, 8 ou 16 bits :

Le transfert d'une chaîne de bits dont l'adresse de début majorée de sa longueur dépasse 16, provoque un chevauchement sur l'emplacement suivant.

Exemple : I1,6[16] adresse les bits I1,6 à I1,F (quelque soit le module 1) et les bits I ou O 2,0 à I ou O 2,5.

Pour des raisons de sécurité, il est conseillé de vérifier qu'il n'y a pas chevauchement entre 2 modules.

5.2-2 Transfert d'une chaîne de bits dans un mot

Cette opération consiste à recopier l'état de n bits d'une chaîne dans un mot. La longueur de la chaîne de bits ne devra pas être supérieure à 16.

Structure

Le transfert est réalisé par un bloc opération constitué de la manière suivante :



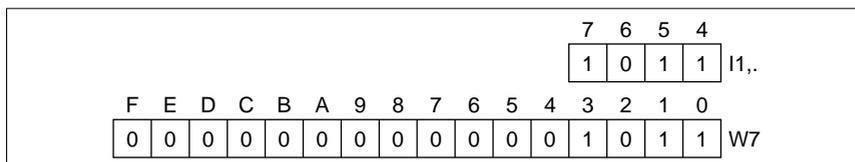
Opérandes

Objet	Chaîne de bits origine	Mot de destination	Exemple
Bit interne Bi	X		B14[16]
Bit d'E/S I/Oxy,i	X		I3,5[12]
Mot interne Wi		X	W12

Exemple

Transfert du contenu des bits d'entrée I1,4 à I1,7 dans le mot W7 :

OP2 I1,4[4]
→ W7



Règles d'utilisation

- Les bits de la chaîne sont transférés dans le mot en commençant par la droite (premier bit de la chaîne dans bit 0 du mot).
- Les bits d'un mot non concernés par le transfert (longueur < à 16) sont positionnés à l'état 0.

5.2-3 Transfert d'un mot dans une chaîne de bits

Cette opération consiste à recopier bit à bit les 16 bits d'un mot dans une chaîne de bits. Lorsque la chaîne de bits est de longueur inférieure à 16, seul les bits de poids faible sont recopiés.

Structure

Le transfert est réalisé par un bloc opération constitué de la manière suivante :



Opérande

Objet	Mot origine	Chaîne de bits destination	Exemple
Bit interne Bi		X	B14[16]
Bit d'entrée Ix,i		X ⁽¹⁾	I3,5[7]
Bit de sortie Ox,i		X	O2,5[5]
Mot interne Wi	X		W12
Mot constant CWi	X		CW10

⁽¹⁾ voir règles d'utilisation.

Exemple

Transfert des bits 0 à F du mot constant CW14 dans les bits internes B10 à B25 :

OP2 CW14
→ B10[16]

Règles d'utilisation

L'utilisation de bits d'entrées dans la chaîne de bits destination est possible mais doit être employée avec précautions (inhibition de l'état réel des entrées jusqu'à la prochaine acquisition des entrées par l'automate).

Les bits du mot origine sont transférés à partir de la droite (bit 0 du mot dans le premier bit de la chaîne).

Cas spécifiques

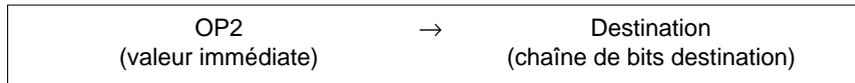
Transfert d'un mot dans une chaîne de bits de sorties: ce type de transfert affecte une chaîne de bits de longueur 16. Dans le cas où le module de sortie n'est pas un module 16 bits ou que le premier bit de la chaîne n'est pas le bit 0 du module, il y a débordement sur le module suivant.

5.2-4 Transfert d'une valeur immédiate dans une chaîne de bits

Cette opération consiste à recopier bit à bit la représentation binaire sur 16 bits d'une valeur immédiate dans une chaîne de bits.

Structure

Le transfert est réalisé par un bloc opération constitué de la manière suivante :



Opérandes

La chaîne de bits destination peut être constituée par des bits d'entrées/sorties. L'utilisation de bits d'entrées est possible mais doit être employée avec précaution (inhibition de l'état réel des entrées jusqu'à la prochaine acquisition des entrées par l'automate).

La valeur immédiate est nécessairement comprise entre -32768 et +32767.

Exemple

Mise à l'état 1 des bits B30 à B37.

OP2 H'00FF'
→ B30[8]

5.3 Opérations sur mots

5.3-1 Transfert

Les opérations de transfert suivantes peuvent être réalisées :

- mot → mot
- mot → mot indexé
- mot indexé → mot
- mot indexé → mot indexé
- valeur immédiate → mot
- valeur immédiate → mot indexé
- table de mots → table de mots

Structure

Le transfert est réalisé par un bloc opération constitué de la façon suivante :



Opérandes

Objet	Mot origine	Mot destination	Exemple	Exemple indexé
Mot constant CWi	X		CW3	CW3(W45)
Mot interne Wi	X	X	W12	W12(W11)
Mot commun COMi,j	X	X	COM3,2	
Mot registre IW/OWxy,i	X	X	OW7,5	
Mot système SWi	X	X	SW4	
Mot extrait de blocs fonctions (ex.Ti,P)	X	X ⁽¹⁾	T3,P	
Valeur immédiate	X		3476	
Temps d'activité d'étape Xi,V	X		X15,V	

(1) sauf valeur courante de bloc fonction.

Exemples

Transfert d'une valeur immédiate dans un mot.

OP2 127
→ T2,P

Transfert de la valeur immédiate 127 dans le mot T2,P (présélection du temporisateur T2).

Transfert d'un mot dans un mot

Transfert du mot W31 dans le mot W40

OP2 W31
→ W40

Transfert d'un mot dans un mot indexé.

Transfert du mot W12 dans le mot W50(W26)

Le contenu du mot W26 utilisé comme index pouvant être modifié, le mot W12 peut être transféré dans différents mots à partir de W50.

OP2 W12
→ W50(W26)

Transfert d'un mot indexé dans un mot.

Transfert du mot indexé CW5(W11) dans le mot W34.

Le contenu du mot W11 pouvant être modifié, différents mots à partir de CW5 peuvent être transférés dans W34.

OP2 CW5(W11)
→ W34

Transfert d'un mot indexé dans un mot indexé.

Transfert du mot CW30(W22) dans le mot W30(W22).

OP2 CW30(W22)
→ W30(W22)

Attention : Le mot utilisé comme index ne doit jamais être négatif.

Le contenu de l'index ne doit jamais dépasser la valeur 127 diminuée de l'adresse du mot interne ou constant sauf sur TSX 17-20 si SY2 = 1.

Ex. : W80(W127) W127 ne doit pas dépasser la valeur
127 - 80 = 47.

W0 (W127) W127 ne doit pas dépasser la valeur
127 - 0 = 127.

Transfert de tables.

Cette opération permet l'accès à la totalité des 1024 mots internes et des 1024 mots constants.

Une table de mots est un ensemble de mots de même type placés les uns à la suite des autres. Elle est caractérisée par :

- le type de mots : mots internes W ou mots constants CW
- l'adresse du premier mot
- la longueur (nombre de mots de la table) : ≤ 128 .

Les 1024 mots constants CW_i (configurables) et les 1024 mots internes sont structurés par pages de 128 mots.

La table origine peut être constituée par des mots internes ou constants.

La table destinataire est toujours constituée par des mots internes.

Seul la page des mots internes 0 à 127 est accessible en lecture/écriture et constitue une page de travail. La lecture ou l'écriture des autres pages s'effectue en les transférant dans la page de travail.

Chaque page étant une entité, une page ne peut pas chevaucher deux pages.

Exemple : W200[100] cette table n'est pas prise en compte car il y a chevauchement entre la table W128 - W255 et la table W256 - W383

0	-	127
128	-	255
256	-	383
384	-	511
512	-	639
640	-	767
768	-	895
896	-	1023

Transfert de 60 mots internes W512 à W571 dans le 60 mots internes W0 à W59.

OP2 W512[60]

→ W0[60]

Transfert des 128 mots constants CW256 à CW383 dans les 128 mots internes W0 à W127.

OP2 CW256[128]

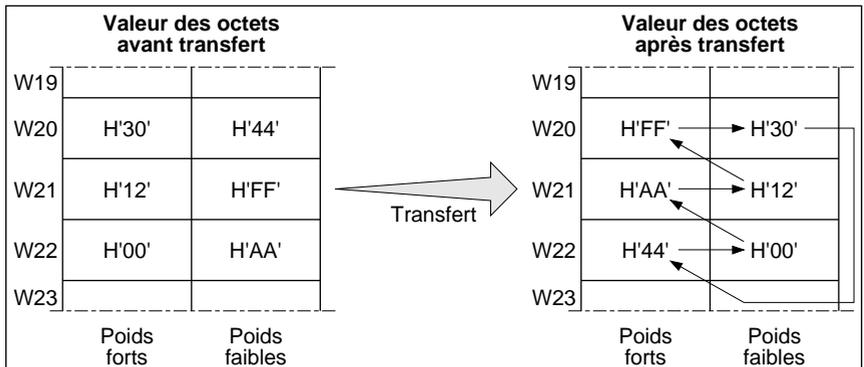
→ W0[128]

Cas particuliers de transfert de table :

Transfert de table dans laquelle la table origine est identique à la table destination. Dans ce cas, le transfert provoque un décalage global d'un octet de la table concernée.

Exemple :

W20 [3] → W20 [3]



5.3-2 Décalage circulaire.

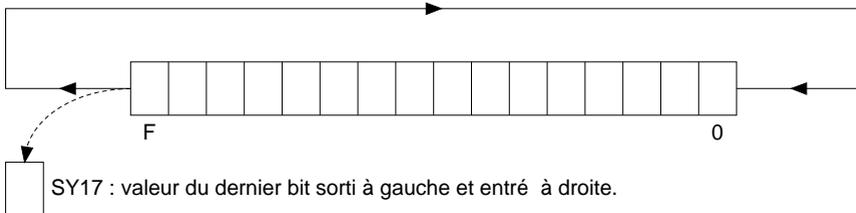
L'opération de décalage consiste à déplacer les bits contenus dans un mot d'un certain nombre de positions vers la droite ou vers la gauche.

Le décalage circulaire peut être réalisé dans les deux sens à l'aide des deux instructions :

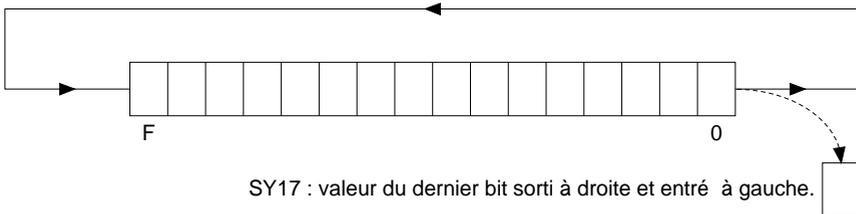
- SLCi : décalage circulaire à gauche de i positions (Shift Left Circular).
- SRCi : décalage circulaire à droite de i positions (Shift Right Circular).

L'opérande à décaler étant nécessairement un mot de 16 bits, la variable i sera comprise entre 0 et 16 ($1 \leq i \leq 15$).

Décalage circulaire à gauche : SLCi



Décalage circulaire à droite : SRCi



L'ensemble des bits est décalé de i positions. l'état du dernier bit sorti, donc du dernier bit entré est mémorisé par le bit système SY17.

Structure

Dans tous les cas une opération de décalage est caractérisée par :

- son type : SLC, SRC
- le nombre de positions de décalage i ($1 \leq i \leq 15$)
- l'opérande à décaler

Le décalage est réalisé par un bloc opération constitué de la manière suivante :

OP type de décalage SLCi ou SRCi $1 \leq i \leq 15$	OP2 mot à décaler	→ mot résultat
--	----------------------	-------------------

Opérandes

Objet	Mot à décaler	Mot résultat	Exemple	Exemple indexé
Mot constant CWi	X		CW3	CW3(W45)
Mot interne Wi	X	X	W12	W12(W11)
Mot commun COMi,j	X	X	COM3,2	
Mot registre IW/OWxy,i	X	X	OW7,5	
Mot système SWi	X	X	SW4	
Valeur immédiate	X		3476	
Temps d'activité d'une étape Xi,V	X		X67,V	
Mot extrait de bloc fonction (Ex: Ti,P)	X	X ⁽¹⁾	T3,P	

⁽¹⁾ sauf valeur courante de bloc fonction

Exemple

- Décalage circulaire du mot W1 de 3 positions vers la gauche et rangement du résultat dans le mot W14.

```
OP  SLC3
OP2 W1
→  W14
```

- Décalage circulaire du mot W1 de 4 positions vers la droite et rangement du résultat dans le mot W14

```
OP  SRC4
OP2 W1
→  W14
```



5.3-3 Comparaison

L'instruction de comparaison permet de comparer deux opérandes. Les différentes instructions de comparaison sont les suivantes :

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

Structure

La comparaison est réalisée par un bloc comparaison qui est toujours situé en zone test. Sa sortie est à 1 lorsque la comparaison demandée est vraie.

Ce bloc est constitué de la manière suivante

OP1 opérande 1	TYPE DE COMPARAISON <, <=, >, >=, =, <>	OP2 opérande 2
-------------------	--	-------------------

Opérandes

Objet	Opérande		Exemple	Exemple indexé
	OP1	OP2		
Mot constant CWi	X	X	CW3	CW3(W10)
Mot interne Wi	X	X	W10	W10(W11)
Mot commun COMi,j	X	X	COM3,5	
Mot registre IW/OWxy,i	X	X	OW7,5	
Mot système SWi	X	X	SW4	
Valeur immédiate	X	X	5425	
Temps d'activité d'une étape Xi,V	X	X	X56,V	
Mot extrait de bloc fonction (Ex.Ti,P)	X	X	T3,P	

Exemple

Comparaison du mot W241 avec la valeur immédiate 100.

La sortie du bloc est à 1 lorsque la comparaison est vraie (contenu du mot W241 supérieur à 100)

OP1 W241
OP >
OP2 100

5.3-4 Opérations arithmétiques

Les instructions associées permettent de réaliser une opération arithmétique entre deux opérandes. Ces instructions sont :

- +** : réalise l'addition de deux opérandes,
- : réalise la soustraction de deux opérandes,
- *** : réalise la multiplication de deux opérandes,
- /** : réalise la division de deux opérandes,
- REM** : réalise la division de deux opérandes et permet d'en exprimer le reste (uniquement sur TSX 17-20).

Structure

Une opération arithmétique est réalisé par un bloc opération constitué de la manière suivante :

OP1	OP	OP2	→
opérande 1	type d'opération +, -, *, /, REM	opérande 2	mot de rangement

Opérandes

Objet	Opérandes		Mot de rangement	Exemple	Exemple indexé
	OP1	OP2			
Mot constant CWi	X	X		CW3	CW3(W45)
Mot interne Wi	X	X	X	W12	W12(W11)
Mot commun COMi,j	X	X	X	COM3,2	
Mot registre IW/OWxy,i	X	X	X	OW7,5	
Mot système SWi	X	X	X	SW4	
Valeur immédiate	X	X		6742	
Temps d'activité d'une étape Xi,V	X	X		X45,V	
Mot extrait de blocs fonctions (Ex.Ti,P)	X	X	X ⁽¹⁾	T3,P	

⁽¹⁾ sauf valeur courante de bloc fonction

Exemple

Addition du contenu de CW12 et de la valeur immédiate 1300. Le résultat de l'addition est rangé dans le mot de de présélection du compteur C5.

```
OP1  CW12
OP   +
OP2  1300
→    C5,p
```

Règles sur les priorités

- l'addition et la soustraction ont la même priorité,
- la multiplication, la division et le calcul du reste ont la même priorité et sont prioritaires sur l'addition et la soustraction.

Valeurs limites des opérandes

Codes opération +, -	Valeurs limites des opérandes (décimale) -32768 <= valeurs <= +32767
*, /	0 <= valeurs <= 255 (TSX 27/47) -32768 < valeurs < +32767 (TSX 17-20)
REM	-32768 <= valeurs <= +32767 (uniquement sur TSX 17-20)

Règles d'utilisation

- **Addition** : opérande 1 + opérande 2

Dépassement de capacité pendant l'opération :

Dans le cas où le résultat dépasse la longueur du plus long des opérandes (-32768 à +32767), le bit SY18 (overflow) est mis à l'état 1. Le résultat est donc non significatif. La gestion du bit SY18 est à la charge du programme utilisateur.

Exemple : $W0 + W1 \rightarrow W2$

0	1	0	1	1	0	1	0	1	1	0	0	1	0	0	1	W0	23241	
+																		
0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	W1	21853	
1	1	0	1	1	0	0	0	0	0	0	1	0	0	1	1	0	W2	-20442

SY18

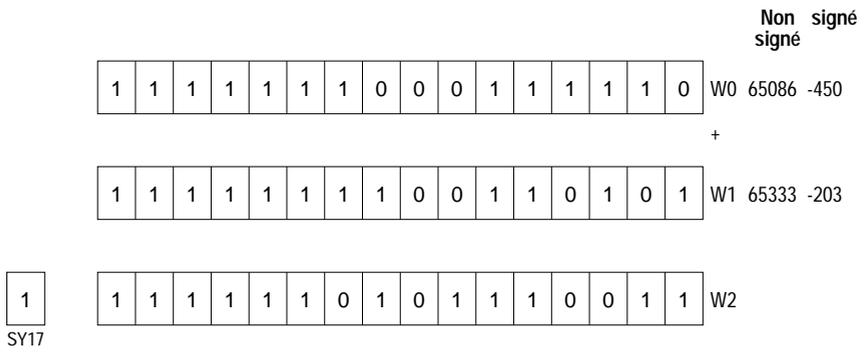
Le résultat réel (45094) ne peut pas être exprimé dans un mot de 16 bits, le bit système SY18 est mis à l'état 1 et le résultat obtenu (-20442) est erroné.

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

Lors de certains calculs, il est parfois 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 SY17 (carry) qui représente la valeur 65536. La gestion de SY17 est à la charge du programme utilisateur.

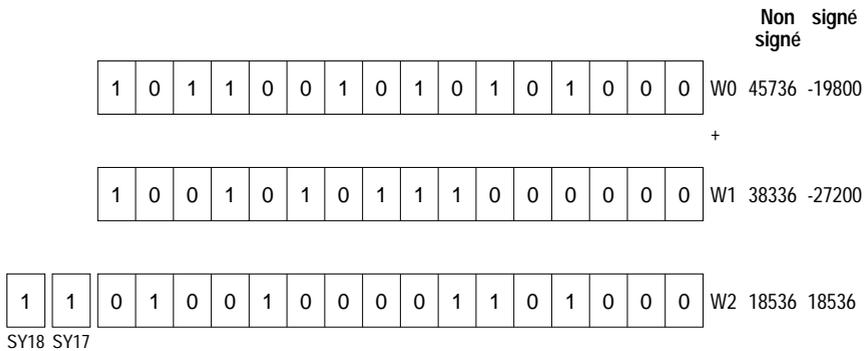
Exemple : $W0 + W1 \rightarrow W2$



Le bit SY17 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.

Les deux cas décrits ci-dessus peuvent se produire simultanément. Dans ce cas les deux bits système SY17 et SY18 sont mis à l'état 1. La gestion de ces bits est à la charge du programme utilisateur.

Exemple : $W0 + W1 \rightarrow W2$



Les deux bits système SY17 et SY18 sont mis à l'état 1. Le résultat en arithmétique signée (+18536) est erroné. En arithmétique non signée, le résultat (18536 + valeur de SY17 soit 84072) est correct.

-
- **Soustraction** : opérande 1 - opérande 2

Résultat négatif

Dans le cas où le résultat de la soustraction est inférieur à 0, le bit système SY17 est mis à l'état 1. La gestion de ce bit est à la charge de l'utilisateur.

- **Multiplication** : opérande 1 * opérande 2

Signe du résultat

Il dépend du signe des opérandes à multiplier :

$$2 * 3 \rightarrow 6$$

$$2 * -3 \rightarrow -6$$

$$-2 * 3 \rightarrow -6$$

$$-2 * -3 \rightarrow 6$$

Débordement de capacité pendant l'opération

Dans le cas où le résultat dépasse la capacité du mot de rangement, le bit SY18 (overflow) est mis à l'état 1 et le résultat est non significatif. La gestion de ce bit est à la charge du programme utilisateur.

- **Division** : opérande 1 / opérande 2

Opérandes

Le dividende est représenté par l'opérande 1 et le diviseur par l'opérande 2. Le quotient peut être transféré dans un opérande 3.

Signe du résultat

Le signe du quotient dépend des signes du dividende (OP1) et du diviseur (OP2) :

$$8 / 2 \rightarrow 4$$

$$8 / -2 \rightarrow -4$$

$$-8 / 2 \rightarrow -4$$

$$-8 / -2 \rightarrow 4$$

Division par 0

Dans le cas où le diviseur est égal à 0, la division est impossible et le bit système SY18 est mis à l'état 1, le résultat sera donc erroné. La gestion de ce bit est à la charge du programme utilisateur.

Débordement de capacité pendant l'opération

Dans le cas où le quotient de la division dépasse la capacité du mot de rangement, le bit SY18 est mis à l'état 1. La gestion de ce bit est à la charge du programme utilisateur.

- **Reste de la division** : opérande 1 REM opérande 2
(uniquement avec TSX 17-20)

Opérandes

Le dividende est représenté par l'opérande 1 et le diviseur par l'opérande 2. Le reste peut être transféré dans un opérande 3.

Signe du résultat

Le signe du reste est toujours identique à celui du dividende.

Division par 0

Dans le cas ou le diviseur est égal à 0, la division est impossible et le bit système SY18 est mis à l'état 1; le résultat sera donc erroné. La gestion du bit SY18 est à la charge du programme utilisateur.

5.3-5 Opérations logiques

Les instructions associées permettent de réaliser une opération logique entre deux opérandes; ces instructions sont :

- AND : réalise le ET (intersection) entre deux opérandes,
- OR : réalise le OU inclusif (union) entre deux opérandes,
- XOR : réalise le OU exclusif (disjonction) entre deux opérandes,
- CPL : réalise le complément logique d'un opérande.

Une opération logique entre deux opérandes est effectuée bit à bit (bit 0 à F) sur le contenu des deux opérandes.

Structure

Une opération logique est réalisée par un bloc opération constitué de la manière suivante.

Opérations AND, OR, XOR

OP1	OP	OP2	→
Opérande 1	Type d'opération AND,OR,XOR	Opérande 2	Mot de rangement

Complément logique CPL

OP	OP2	→
Type d'opération CPL	Opérande	Mot de rangement

Opérandes

Le complément logique ne s'effectue que sur des mots internes Wi.

Objet	Opérandes		Mot de rangement	Exemple	Exemple indexé
	OP1	OP2			
Mot constant CWi	X	X		CW3	CW3(W45)
Mot interne Wi	X	X	X	W12	W12(W11)
Mot commun COMi,j	X	X	X	COM3,2	
Mot registre IW/OWxy,i	X	X	X	OW7,5	
Mot système SWi	X	X	X	SW4	
Valeur immédiate	X	X		3476	
Temps d'activité d'une étape Xi,V	X	X		X67,V	
Mot extrait de blocs fonction Ti,P	X	X	X ⁽¹⁾	T3,P	

⁽¹⁾ sauf valeur courante de bloc fonction

Exemples

ET logique : le résultat bit à bit est égal à 1 si les deux bits correspondants sont à l'état 1.

Rappel Exemple

```

0 AND 0 → 0
0 AND 1 → 0
1 AND 0 → 0
1 AND 1 → 1

```

OP1 W32 1100 1100 1101 1011
OP AND
OP2 W12 0100 1011 1111 0000

→ W15 0100 1000 1101 0000

OU logique : le résultat bit à bit est égal à 1 si au moins un des deux bits correspondants est à l'état 1.

Rappel

```

0 OR 0 → 0
0 OR 1 → 1
1 OR 0 → 1
1 OR 1 → 1

```

Exemple

```

OP1 W32 1100 1100 1101 1011
OP OR
OP2 W12 0100 1011 1111 0000

```

→ W15 1100 1111 1111 1011

OU exclusif logique : le résultat bit à bit est égal à 1 si un seul des bits correspondants est à l'état 1.

Rappel	Exemple
0 XOR 0 → 0	OP1 W32 1100 1100 1101 1011
0 XOR 1 → 1	OP XOR
1 XOR 0 → 1	OP2 W12 0100 1011 1111 0000
1 XOR 1 → 0	→ W15 1000 0111 0010 1011

Complément logique : le résultat est l'inversion bit à bit du mot à compléter.

Rappel	Exemple
CPL 0 → 1	OP CPL
CPL 1 → 0	OP2 W12 0100 1011 1111 0000
	→ W15 1011 0100 0000 1111

Exemple d'utilisation : masquage de tout ou partie de mot

L'exemple ci-dessous montre le ET logique entre le mot W94 et la valeur immédiate H"00FF"; le résultat étant stocké dans le mot W95. Cette opération permet de masquer les 8 bits de poids forts du mot W94.

OP1 W94	0100 1110 1010 1001
OP AND	
OP2 H"00FF"	0000 0000 1111 1111
→ W95	0000 0000 1010 1001



5.3-6 Transcodage

Une valeur numérique peut être représentée de plusieurs manières dans un mot; ces différentes représentations sont appelées codes. Le passage d'une représentation à une autre est appelée transcodage.

Cette opération peut être associée à une opération de transfert afin de transférer le contenu transcodé dans un mot ou une table de mots.

Les transcodages possibles sont les suivants.

- BIN : transcodage BCD → binaire
- BCD : transcodage binaire → BCD
- ATB : transcodage ASCII → binaire
- BTA : transcodage binaire → ASCII

Structure

Un transcodage est réalisé par un bloc opération constitué de la manière suivante :

OP	OP2	→
Type de transcodage BIN, BCD, ATB, BTA	Opérande à convertir	Mot de rangement

Opérandes

Objet	Opérande OP2	Mot de rangement	Exemple	Exemple indexé
Mot constant CWi	X		CW3	CW3(W45)
Mot interne Wi	X	X	W12	W12(W11)
Mot commun COMi,j	X	X	COM3,2	
Mot registre IW/OWxy,i	X	X	OW7,5	
Mot système SWi	X	X	SW4	
Valeur immédiate	X		3476	
Temps d'activité d'une étape Xi,V	X	X	67,V	
Mot extrait de bloc fonction Ti,P	X	X ⁽¹⁾	T3,P	

⁽¹⁾ sauf valeur courante de bloc fonction

Transcodage BCD → Binaire et Binaire → BCD• **Rappel sur le code BCD :**

Le code BCD (Binary Coded Décimal) 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 donc contenir un nombre N exprimé sur 4 décades ($0 \leq N \leq 9999$).

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

Exemples:• **Transcodage BCD → Binaire (BIN)**

La valeur du mot W1, exprimée en code BCD est convertie en code binaire puis rangée dans le mot W2.

```
OP   BIN
OP2  W1      0001 0111 0101 1001
→    W2      0000 0110 1101 1111
```

• **Transcodage Binaire → BCD (BCD)**

La valeur du mot W10 exprimée en code binaire est convertie en code BCD puis rangée dans le mot W21.

```
OP   BCD
OP2  W10     0000 0110 1101 1111
→    W21     0001 0111 0101 1001
```

Règles d'utilisation• **Transcodage BCD → Binaire**

Le transcodage s'effectue sur l'ensemble des bits de l'opérande à transcoder. Un contrôle est effectué automatiquement par l'automate sur chaque quartet (4 bits) de l'opérande. Chaque quartet ne doit pas représenter une valeur décimale supérieure à 9. Cette vérification a pour but d'effectuer l'opération de transcodage sur un opérande contenant bien une valeur exprimée en code BCD; dans le cas où cette valeur n'est pas une valeur BCD, le bit système SY18 (débordement) est positionné à l'état 1. La gestion de ce bit est à la charge du programme utilisateur.

• **Transcodage Binaire → BCD**

L'opération de transcodage ne peut s'effectuer que sur un opérande dont le contenu est compris entre 0 et 9999. Dans le cas contraire, le résultat du transcodage est erroné et le bit système SY18 est positionné à l'état 1. La gestion de ce bit est à la charge du programme utilisateur.

Transcodage Binaire → ASCII et ASCII → Binaire

• Rappel sur le code ASCII

L'ensemble des 256 caractères alphanumériques et de contrôles peut être codé sur 8 bits. Ce code appelé ASCII (American Standard Code for Information Interchange) est compatible avec la notion d'octets. Tout mot de 16 bits peut donc être formé par 2 codes ASCII définissant 2 caractères.

• Transcodage Binaire → ASCII

L'opération de transcodage Binaire → ASCII consiste à convertir la valeur binaire d'un mot (5 chiffres plus signe) en code ASCII sur 3 mots.

Exemple

Conversion de la valeur binaire du mot W5 et rangement dans les mots W20 à W22.

OP BTA
OP2 W5
→ W20

Valeur de W5 = +128

0	0
1	0
8	2

H30	H30
H31	H30
H38	H32

W20
W21
W22

Valeur de W5 = -128

0	-
1	0
8	2

H30	H2D
H31	H30
H38	H32

Valeur de W5 = +32567

3	0
15	2
7	6

H33	H30
H35	H30
H37	H32

W20
W21
W22

Valeur de W5 = +0

0	0
0	0
0	0

H30	H30
H30	H30
H38	H30

Règles d'utilisation

Le contenu d'un mot étant compris entre -32768 et +32767 soit 5 chiffres plus signe, la destination du transcodage sera toujours une tables de 3 mots soit 6 octets.

Le code H2D (-) doit être rangé dans l'octet du poids faible du premier des 3 mots de rangement.

Le rangement des caractères est réalisé de manière à effectuer aisément un échange de caractères via une liaison série asynchrone.

Transcodage ASCII → Binaire

L'opération de transcodage ASCII → Binaire consiste à convertir les codes ASCII représentant les chiffres 0 à 9 en code binaire.

Exemple

Conversion des codes ASCII rangés dans les mots W30 à W32 et rangement dans le mot W40.

OP ATB
 OP2 W30
 → W40

W30	H30	H30
W31	H31	H30
W32	H38	H32

} W40 = +128

W30	H30	H2D
W31	H31	H30
W32	H38	H32

} W40 = -128

W30	H33	H30
W31	H37	H32
W32	H37	H36

} W40 = +32767

W30	H30	H30
W31	H30	H30
W32	H30	H30

} W40 = 0

Règles d'utilisation

Les codes ASCII à convertir sont toujours rangés dans une table de 3 mots.

Le nombre exprimé en ASCII rangé dans une table de 3 mots doit obligatoirement être compris entre -32767 et +32768 afin d'être rangé dans un mot résultat. Dans le cas contraire, le mot résultat ne peut contenir la valeur et le bit système SY18 (overflow) est mis à l'état 1. La gestion de ce bit est à la charge du programme utilisateur.

Le code H30 (0) est le code neutre.

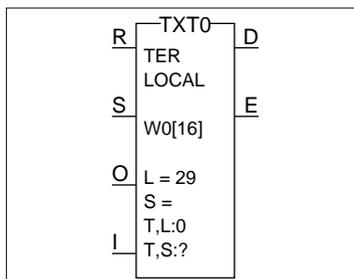




Sous-chapitre	Page
6.1 Présentation	6/2
6.2 Description de l'échange	6/2
6.2-1 Tables de données	6/4
6.2-2 Rangement des données dans les tables	6/6
6.3 Caractéristiques	6/7
6.4 Fonctionnement	6/9
Ce chapitre se termine à la page	6/10

6.1 Présentation

Le bloc "Texte" est un bloc fonction qui véhicule des messages et permet ainsi à l'automate de communiquer avec le milieu extérieur.



6.2 Description de l'échange

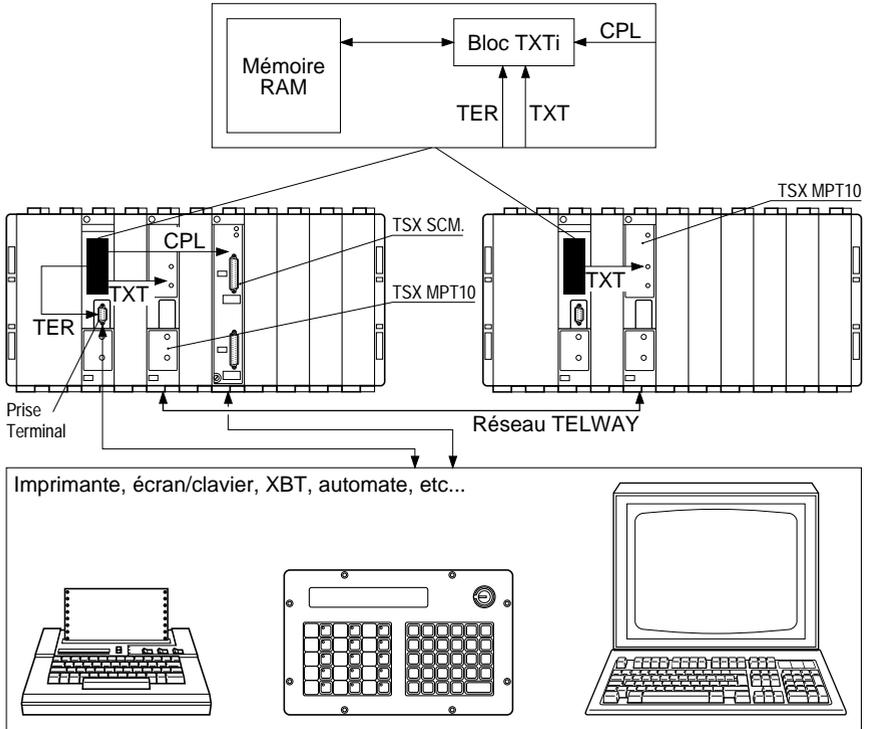
Toute communication entre un automate et un périphérique ou un autre automate nécessite l'emploi d'un bloc texte comme "enveloppe" et suppose la connaissance des éléments suivants :

- Le type d'échange : avec qui veut-on communiquer? :
 - TER (communication avec un périphérique via la prise Terminal),
 - CPL (communication avec un périphérique, via un module coupleur de l'automate),
 - TXT (communication avec un autre automate, via le réseau FIPWAY ou TELWAY.
Le message émis par un bloc texte de la station A est reçu par un bloc texte de la station B),
- Le support de communication : par quel moyen veut-on communiquer? :
 - LOCAL (localement, vers prise Terminal ou module coupleur),
 - RESEAU FIPWAY (TSX17-20) ou TELWAY (TSX47)
 - BUS UNI-TELWAY,
- Les tables de données : que veut-on communiquer? :
 - longueur et adresse de début de la table d'émission,
 - longueur de la table de réception.

Ces éléments sont définis lors de la saisie du bloc texte et ne sont modifiables que par le programme utilisateur.

La figure et le tableau de la page ci-contre, illustrent les différents échanges, suivant le type de blocs textes.

B



Type de Bloc Texte	Echange				Moyen d'échange de données
	TSX 17-20	TSX 27	TSX 47-J/10	TSX 47-20	
TER	Programme utilisateur avec un périphérique, via la prise terminal				Local
TXT	Programme utilisateur d'un automate A avec un programme utilisateur d'un automate B		Programme utilisateur d'un automate A avec un programme utilisateur d'un automate B		Réseau
CPL	Programme utilisateur avec un serveur UNI-TELWAY, via la prise Terminal				Bus de Liaison Multipoint UNI-TELWAY
	Programme utilisateur avec un coupleur intelligent			Programme utilisateur avec un coupleur intelligent	Local

Les échanges de données entre le programme utilisateur et les coupleurs intelligents peuvent être effectués avec les coupleurs suivants :

TSX 17-20	Coupleur réseau FIPWAY	: TSX FPG 10
TSX 17-20	Coupleur de communication par BUS UNI-TELWAY Module de communication Chaîne de caractères	: TSX SCG 116 : TSX SCG 113
TSX 47-J TSX 47-10/20	Coupleur réseau TELWAY	: TSX MPT10
TSX 47-20	Coupleur de communication par bus UNI-TELWAY	: TSX SCM 21.6
TSX 47-20	Coupleurs liaisons asynchrones Coupleurs Chaîne de mesure	: TSX SCM 2.. : TSX AEM

Chaque coupleur ayant des fonctionnalités bien définies (paramètres, commandes, ...), les échanges correspondants sont explicités dans des manuels spécifiques.

Les périphériques connectés à la prise terminal peuvent être une imprimante, un écran,... mais aussi un terminal TSX T317/407 ou FTX 417/507. Ces derniers devront être utilisés en mode "terminal".

Le périphérique (du type imprimante), peut être aussi connecté à la prise "périphérique" du terminal T407, lui même relié à la prise terminal de l'automate. Dans ce cas la validation de la liaison série du terminal T407 est provoquée par l'envoi du code H'12' par le bloc texte. Le blocage de cette liaison est obtenu par l'envoi du code H'14'.

6.2-1 Tables de données

Un message véhiculé par un bloc texte (échangé entre un automate et un périphérique), est un ensemble de caractères alphanumériques contenu dans une table de mots. Chaque caractère composant ce message, est codé sur 8 bits.

Les blocs textes du type TER véhiculent généralement des messages codé en ASCII (American Standard Code Information Inter-change). Ce code est compatible avec la notion d'octets (8 bits). Exemple le code ASCII de la lettre "A", exprimé en Hexadécimal est H'41'.

Tout mot de 16 bits peut donc contenir 2 caractères ASCII.

Les blocs textes du type CPL ou TXT véhiculent en général des messages définis en binaire pur, sans codage (valeurs numériques, etc).

Les données échangées à l'aide de la fonction "Texte" sont organisées sous forme de tables de mots, définies dans la mémoire de données et structurées de la façon suivante :

La table d'émission

Elle permet d'envoyer des données au destinataire (coupleur, prise terminal,...). Elle peut être constituée par des mots internes W_i ou des mots constants $CW_i^{(1)}$ et est définie par sa longueur L représentant le nombre de caractères (2 caractères/mot).

$L_{max} = 30$ caractères soit 15 mots.

La table de réception

Elle permet de recevoir des données envoyées par l'émetteur. Elle est nécessairement constituée par des mots internes W_i et est définie par son adresse de début W_i et sa longueur n représentant le nombre de caractères (2 caractères/mot).

$n_{max} = 30$ caractères soit 15 mots.

Les tables de réception et d'émission sont juxtaposées et constituent une table dont les paramètres sont : $W_i[n]$ et L .

Exemple de configuration de table: $W_8[10]$ et $L = 12$

- table de réception : W_8 à W_{12} (10 caractères, donc 5 mots),
- table d'émission : W_{13} à W_{18} (nombre de caractères $L = 12$, donc 6 mots).

2 ^{ème} caractère à recevoir	1 ^{er} caractère à recevoir	mot W_i	W_8
		mot W_{i+1}	W_9
Table de réception (mots internes W_i)			
		mot $W_{i+n/2-1}$	W_{12}
2 ^{ème} caractère à émettre	1 ^{er} caractère à émettre	mot $W_{i+n/2}$	W_{13}
		mot $W_{i+n/2+1}$	W_{14}
Table d'émission (mots internes W_i)			
		mot $W_{i+n/2+L/2-1}$	W_{18}

⁽¹⁾ uniquement si le bloc texte est défini en émission, sans réception.

6.2-2 Rangement des données dans les tables

Dans le cas de la réception du message "DROITE" par exemple, les caractères (les codes ASCII correspondants) seront rangés dans la table de réception comme l'indique le tableau ci-contre.

52	44
49	4F
45	54

Code ASCII en Hexadécimal
H'...'

rang	rang	rang	
F	7	0	
R	D		mot Wi
I	O		mot Wi+1
E	T		mot Wi+2

Pour émettre le message "SENS ?" par exemple, les caractères (les codes ASCII correspondants) doivent être rangés dans la table d'émission comme l'indique le tableau ci-contre. Sur le périphérique, les caractères seront affichés dans l'ordre.

45	53
53	4E
3F	20

Code ASCII en Hexadécimal
H'...'

rang	rang	rang	
F	7	0	
E	S		mot Wi+3
S	N		mot Wi+4
?	espace		mot Wi+5

6.3 Caractéristiques

Numéro de bloc Texte	TXTi	$0 < i < 7$
Type d'échange	CPL/TER/TXT	(voir tableau page 6/3)
Type de communication	LOCAL/BUS/RESAU	(voir tableau page 6/3)
Adresse et longueur table de réception	Wi[n]	Wi : adresse du premier mot de la table n : longueur de la table (nombre de caractères). Si n = 0, le bloc texte n'adressera qu'une table d'émission. Dans ce cas la table peut être définie par des mots internes Wi ou constants CWi.
Longueur de la table d'émission	TXTi,L	Mot définissant la longueur de la table, en nombre de caractères. Modifiable par programme utilisateur ou par terminal.
Mot STATUS	TXTi,S	<p>Contient le nombre de caractères reçus si le bit de sortie E (échange erroné) est à 0. Si le bit de sortie E est à 1, ce mot contient le code de l'erreur produite lors de l'échange.</p> <p>Ce mot ne peut être que lu par le programme.</p> <p>Code d'erreur status :</p> <ul style="list-style-type: none"> 1 = échange annulé par RESET, 2 = erreur longueur de table, 3 = message refusé (sur TSX 17-20 et TSX 47-10/20 V>3), 6 = adresse buffer incorrecte (sur TSX 17-20), 11 = type de bloc texte non géré (sur TSX 17-20 uniquement).
Entrée RESET	R	<p>La mise à l'état 1 de cette entrée empêche la prise en compte de l'entrée S.</p> <p>De la même manière, la mise à l'état 1 de cette entrée durant un échange interrompt l'échange et positionne les bits de sortie TXTi,D (échange terminé) et TXTi,E (échange erroné) à l'état 1. Le code de l'erreur est alors contenu dans le mot status TXTi,S.</p>
Entrée Départ	S	Un front montant sur l'entrée S provoque le lancement de l'échange défini par les entrées I (réception) et O (Emission) si le bit de sortie D est à l'état 1 et si l'entrée R est à l'état 0.

Caractéristiques (suite)

Entrée Emission	O	La mise à l'état 1 de cette entrée positionne le bloc en émission.
Entrée Réception	I	La mise à l'état 1 de cette entrée positionne le bloc en réception.
Bit de sortie "Echange terminé"	TXTi,D	Ce bit est mis à l'état 1 lorsque l'échange est terminé. Si le bit TXTi,E (échange erroné) est à l'état 0, TXTi,D indique que le message a été correctement émis et/ou reçu. Si le bit TXTi,E est à l'état 1, TXTi,D indique que l'échange est terminé mais erroné.
Bit de sortie "Echange erroné"	TXTi,E	Ce bit est mis à l'état 1 si l'échange ne s'est pas terminé correctement. Le mot status indique alors le type d'erreur qui s'est produite.
Dimension dans un réseau de contacts		2 colonnes de contacts sur 4 lignes

Paramètres spécifiques aux blocs textes CPL ou TXT⁽¹⁾

Adresse coupleur et numéro de voie	TXTi,M	Utilisé seulement par les blocs de type CPL. Ce mot doit être écrit par programme avant tout échange, il contient l'adresse du coupleur et le numéro de voie destinataire de l'échange.
Code requête	TXTi,C	Utilisé seulement par les blocs de type CPL. Ecrit par programme, ce mot contient sous forme de code le type de travail à effectuer (envoi de la configuration, écriture d'une table, ...). Pour la signification de ces codes, se reporter aux manuels spécifiques.
Compte-rendu d'échange	TXTi,R	Utilisé seulement par les blocs de type CPL. Après un échange, ce mot contient un code écrit par le coupleur ou le système indiquant la validité de l'échange (erroné ou correct). Ce mot ne peut être que lu.
Numéro du bloc	TXTi,T	Utilisé seulement par les blocs de type TXT. Ecrit par programme, ce mot contient le numéro du bloc texte destinataire de l'échange (voir manuels coupleur réseau).
Adresse station	TXTi,A	Utilisé seulement par les blocs de type TXT, lorsque le support de communication est le réseau (NET). Ecrit par programme utilisateur, ce mot contient le numéro de la station destinataire de l'échange

⁽¹⁾ L'utilisation et la gestion de ces mots sont détaillées dans les documents spécifiques aux coupleurs.

6.4 Fonctionnement

La mise à l'état 1 des entrées I ou O permet de positionner le bloc en lecture ou en écriture. La réception ou l'émission est alors déclenchée par un front montant sur l'entrée S (départ), si l'entrée R (Reset) est à l'état 0 et si la sortie D est à l'état 1 (aucun échange en cours). En fin d'écriture ou de lecture, la sortie D est mise à l'état 1. De même si l'échange n'est pas correct, la sortie E (échange erroné) est mise à 1.

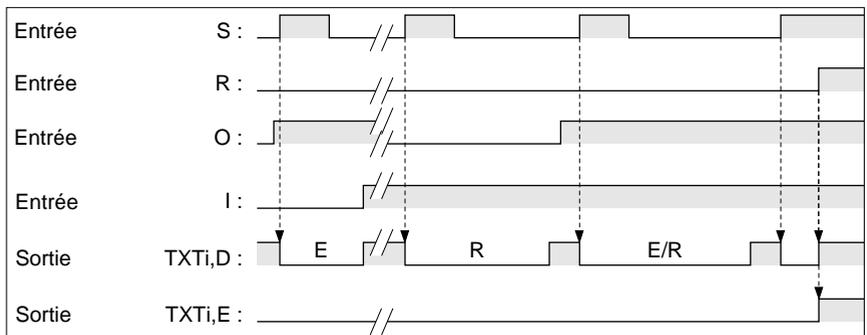
De la même manière la mise à l'état 1 simultanée des entrées I et O positionne le bloc texte en émission puis en réception. L'échange est déclenché par un front montant de l'entrée S (si R=0 et D=1). La sortie D est mise à l'état 1 lorsque l'émission et la réception sont terminées.

A tout instant, la mise à l'état 1 de l'entrée R (entrée prioritaire) permet d'interrompre l'échange en cours. Les bits de sortie D (échange terminé) et E (échange erroné) sont mis à l'état 1. Le mot status TXTi,S indique le type d'erreur du dernier échange en cours. Son contenu sera réinitialisé par un RESET du bloc texte ou réinitialisé lors du prochain échange.

R = 0	S = 1	I = 0	O = 1	Emission
R = 0	S = 1	I = 1	O = 0	Réception
R = 0	S = 1	I = 1	O = 1	Emission suivie de réception
R = 1	S = 0/1	I = 0/1	O = 0/1	Pas de possibilité d'échange ou interruption de l'échange en cours

	Echange en cours	Echange terminé	Echange erroné
TXTi,D	0	1	1
TXTi,E	0	0	1

Diagramme



E : émission; R : réception; E/R : émission/réception.



Sous-chapitre	Page
7.1 Présentation	7/2
7.2 Programmateur Temporel	7/2
7.2-1 Caractéristiques	7/3
7.2-2 Fonctionnement	7/5
7.3 Consignateur Temporel	7/6
7.3-1 Caractéristiques	7/6
7.3-2 Horodatation d'un événement : DT	7/7
7.3-3 Horodatation du dernier arrêt: DTP	7/7
7.4 Mesure de Durée	7/8
7.5 Exemple de mise à l'heure de l'horodateur	7/9
Ce chapitre se termine à la page	7/10

7.1 Présentation

Les automates TSX 17-20 équipés des cartouches micro-logiciel TSX P17 20FB ou 20FD possèdent une horloge, à partir de laquelle peuvent être élaborées trois fonctions :

- **Programmeur Temporel** qui permet de commander des actions à des horaires prédéfinis ou calculés,
- **Consignateur Temporel** qui permet l'horodatation d'événements,
- **Mesure de Durée** entre événements successifs.

La mise à l'heure de l'horodateur du TSX 17-20 s'effectue soit en mode REGLAGE, soit par programme. Son fonctionnement, lorsque l'automate est hors tension est assuré par la pile lithium de sauvegarde.

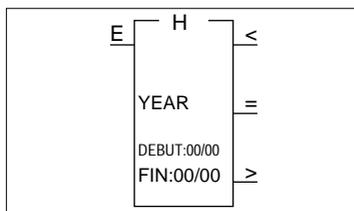
Le rafraîchissement des variables de l'horodateur est assuré tant que SY3=0.

7.2 Programmeur Temporel

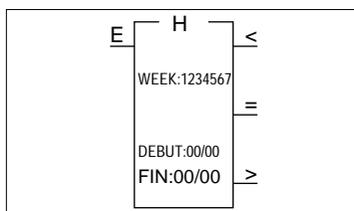
Le programmeur temporel permet de commander des actions à des horaires et des dates prédéfinies ou calculées.

La commande de ces actions peut être journalière, hebdomadaire et/ou annuelle, suivant les blocs horodateurs programmés :

- Annuel (YEAR), les paramètres associés sont le mois dans l'année et le jour dans le mois : MM/JJ.



- Hebdomadaire (WEEK), les paramètres associés sont les jours dans la semaine : N et les heures/minutes dans le jour : HH/MN.



Chaque bloc fonction horodateur (YEAR ou WEEK) est caractérisé par sa consigne de début de période active (DEBUT) et sa consigne de fin de période active (FIN).

Exemple de période temporelle : mise en route du climatiseur d'un bâtiment, du 15 juin au 15 septembre (période du bloc YEAR), du lundi au vendredi, de 9h à 19h (période du bloc WEEK).

7.2-1 Caractéristiques

Bloc Horodateur Nombre	H	Le nombre de blocs horodateur utilisés dans un programme utilisateur est illimité. Il dépend uniquement de la taille de la mémoire programme.
Type	YEAR	Bloc annuel. La période active est une tranche de date dans l'année.
	WEEK	Bloc hebdomadaire. La période active est une tranche horaire dans le jour. Le type de bloc doit être défini lors de la saisie.
Entrée "Validation"	E(ENABLE)	Sur état 1, valide le fonctionnement du bloc.
Sortie	<	"Avant". Sur état 1 indique que la date ou l'heure courante (suivant le type de bloc), est antérieure à la consigne de début de période active.
Sortie	=	"Pendant". Sur état 1 indique que la date ou l'heure courante (suivant le type de bloc), est comprise entre la consigne de début de période active et la consigne de fin de période active.
Sortie	>	"Après". Sur état 1 indique que la date ou l'heure courante (suivant le type de bloc), est postérieure à la consigne de fin de période active.
Dimension dans un réseau de contacts		2 colonnes de contacts sur 3 lignes

Paramètres du bloc horodateur "WEEK"

Sélection des jours "LMMJVSD" "1234567"	1234567	Jours de la semaine dans la période active. Peuvent être définis comme paramètres : <ul style="list-style-type: none"> • constants : CD (Constant Day), modifiables par Terminal, • variables : VD (Variable Day) : <ul style="list-style-type: none"> - Wi, calculé par programme, - CWi, défini par configuration. Exemple : W10 = 00000000 01001101 correspondant à : 1--45-7
Consigne de début	DEBUT : 00:00	Début de la période active: HH:MN Heures et minutes. Peut être défini comme paramètre : <ul style="list-style-type: none"> • constant : modifiable par Terminal, • variable : <ul style="list-style-type: none"> - Wi, calculé par programme, - CWi, défini par configuration. Exemple : W100 = H'1223' correspondant à 12h 23mn
Consigne de fin	FIN : 00:00	Fin de la période active: HH:MN Heures et minutes. Peut être défini comme paramètre : <ul style="list-style-type: none"> • constant : modifiable par Terminal, • variable : <ul style="list-style-type: none"> - Wi, calculé par programme, - CWi, défini par configuration. Exemple : CW100 = H'1859' correspondant à 18h 59mn La valeur de fin de période doit être exceptée : si la fin de période est 18h, il faut saisir "FIN:17:59".

Paramètres de blocs horodateur "YEAR"

Consigne de début	DEBUT : 00/00	Début de la période active: MM/JJ Mois et jour. Peut être défini comme paramètre : <ul style="list-style-type: none"> • constant : modifiable par Terminal, • variable : <ul style="list-style-type: none"> - Wi, calculé par programme, - CWi, défini par configuration. Exemple : W110 = H'0515' correspondant au 15 Mai
Consigne de fin 00/00	FIN :	Fin de la période active: MM/JJ Mois et jour. Peut être défini comme paramètre : <ul style="list-style-type: none"> • constant : modifiable par Terminal, • variable : <ul style="list-style-type: none"> - Wi, calculé par programme, - CWi, défini par configuration. Exemple: CW110 = H'0915' correspondant au 15 Septembre.

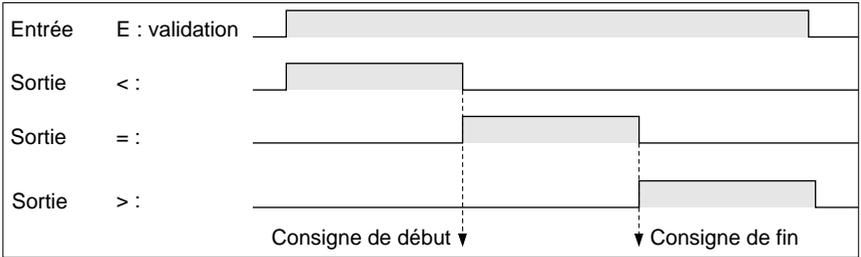
7.2-2 Fonctionnement

Les blocs fonction Horodateur (WEEK et YEAR) se programment en série et/ou en parallèle, suivant l'application.

Exemple précédent : Mise en route du climatiseur d'un bâtiment, du 15 juin au 15 septembre, du lundi au vendredi, de 9h à 19h.

PL7-2		Formulaire programmation schéma à contacts			
N°	ZONE TEST	ZONE ACTION		OBSERVATIONS	
". C . O . N . T . A . C . T . . C . L . I . M					
10,01	YEAR	WEEK	()	10,01 : Marche	
			00,01	YEAR=	
				Début : 06/15	
				Fin : 09/15	
				WEEK=12345	
				Début : 09 : 00	
				Fin : 18 : 59	

Diagramme



Nota : Dans un bloc horodateur, la valeur de fin de période active doit toujours être postérieure à la valeur de début de période active. Une période active ne peut pas être répartie sur deux journées ou deux années successives.

Les consignes variables Wi ou CWi doivent être définies en BCD (saisies en hexadécimal). Exemple: pour définir 17h 40 dans W20, il faut saisir : H'1740' → W20.

7.3 Consignateur Temporel

Le consignateur temporel permet d'horodater des événements grâce aux informations fournies par les mots SW50 à SW57.

7.3-1 Caractéristiques

Valeurs courantes date et heure	SW50 = H'SSAA'	mot contenant le siècle : SS et l'année : AA
	SW51 = H'MMJJ'	mot contenant le mois : MM et le jour : JJ
	SW52 = H'HHMN'	mot contenant l'heure : HH et les minutes : MN
	SW53 = H'SSXN'	mot contenant les secondes : SS et le numéro du jour dans la semaine (1, 2, 3, 4, 5, 6 ou 7 respectivement L, M, M, J, V, S ou D) : N La valeur de X est égale à 0.

Ces mots sont exprimés en hexadécimal : H'....'.

Ils peuvent être lus et écrits (mise à jour de l'horloge par la mise à jour de ces mots. Voir exemple page 7/9).

Valeur du dernier arrêt : • coupure secteur • STOP automate	SW54 = H'SSAA'	mot contenant le siècle : SS et l'année : AA du dernier arrêt
	SW55 = H'MMJJ'	mot contenant le mois : MM et le jour : JJ du dernier arrêt
	SW56 = H'HHMN'	mot contenant l'heure : HH et les minutes : MN du dernier arrêt
	SW57 = H'SSXP'	mot contenant les secondes : SS et le code du défaut : P. La valeur de X est égale à 0. P = 1 STOP AUTOMATE P = 4 coupure secteur

Ces mots sont exprimés en hexadécimal : H'....'.

Ils peuvent être uniquement lus.

Remarque : Les paramètres des blocs Horodateur sont définis en BCD (valeurs comprises entre 0 et 9), et saisis en hexadécimal (SW50=H'1989'). Le code hexadécimal étant une extension du code BCD.

7.3-2 Horodatation d'un événement : DT

L'apparition d'un événement quelconque interne (atteinte d'une valeur de présélection, mise à 1 d'un bit système, ...), ou externe (information provenant d'un capteur, ...), peut être datée et signalée sur un périphérique par émission d'un bloc texte.

Le message : SSAA-MM-JJ-HH:MN:SS-XP (22 caractères) de la fonction : DT "Date and Time" (contenu de SW50 à SW53 plus les caractères de séparation – et :) peut être transféré dans une table d'émission, puis envoyé sur un périphérique.

DT	Valeur courante	SS AA -MM- JJ -HH: MN :SS- XP (11 mots)
----	-----------------	---

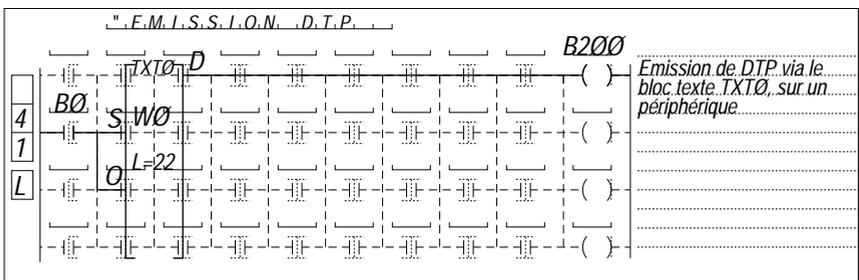
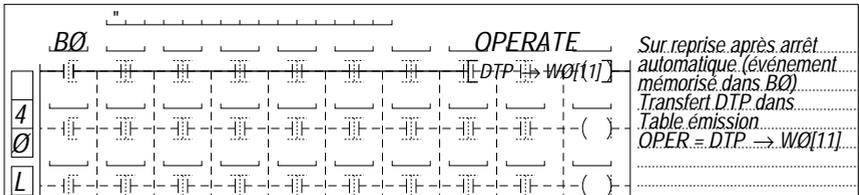
7.3-3 Horodatation du dernier arrêt : DTP

L'information d'arrêt du TSX 17-20 peut être, à chaque fois, affichée sur un périphérique (écran, imprimante, ...), par émission d'un bloc texte. Cette opération permet entre autres d'effectuer des statistiques sur les arrêts.

Le message : SSAA-MM-JJ-HH:MN:SS-XP (22 caractères) de la fonction : DTP "Date and Time Powerfail" (contenu de SW54 à SW57 plus les caractères de séparation – et :) peut être transféré dans une table d'émission, puis envoyé sur un périphérique.

DTP	Valeur dernier arrêt	SS AA -MM- JJ -HH: MN :SS- XP (11 mots)
-----	----------------------	---

Exemple de programmation :



Remarque : L'émission d'un message sur un périphérique nécessite au préalable une conversion Binaire vers ASCII. Dans le cas de la fonction DT ou DTP, cette conversion n'est pas nécessaire. Elle est assurée automatiquement par le système.

Un bloc texte pouvant émettre jusqu'à 30 caractères (maximum, à la fois), les messages des fonctions DT et DTP (22 caractères) peuvent être accompagnés chacun de 8 autres caractères (pour les commentaires par exemple).

7.4 Mesure de Durée

Cette fonction permet d'effectuer des mesures de durée entre deux événements quelconques, grâce aux informations fournies par les mots système SW58 et SW59.

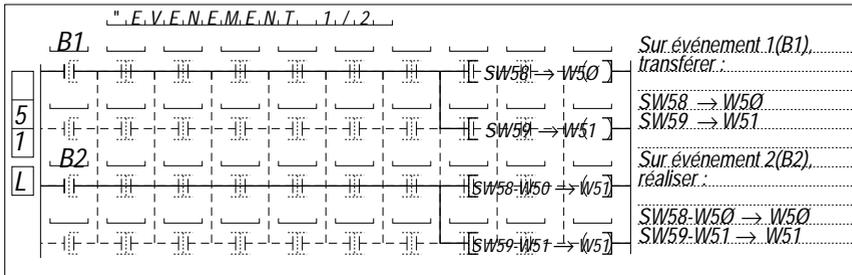
Lors de l'apparition de l'événement, les valeurs courantes contenues dans ces mots peuvent être prélevées (transférées dans des mots Wi) et utilisées dans des opérations arithmétiques.

Caractéristiques

Valeurs courantes	SW58 = D'xxxx'	heure courante en minutes, dans le jour courant : 0 ≤ xxxx: valeur décimale ≤ 1439
	SW59 = D'xxxx'	numéro du jour courant dans l'année courante : 0 ≤ xxxx: valeur décimale ≤ 366
Ces mots système peuvent être uniquement lus. Ils sont rafraîchis tant que SY3=0.		

Exemple de programmation :

Transférer dans les mots W50 et W51 le temps écoulé (nombre de minutes et de jours), entre deux événements 1 et 2.



7.5 Exemple de mise à l'heure de l'horodateur

La mise à l'heure de l'horodateur peut être effectuée :

- soit en mode REGLAGE, à partir du terminal,
- soit par programme utilisateur. Dans ce cas elle ne peut avoir lieu que si SY3=1 (voir ci-dessous).

Exemple de mise à l'heure, sur demande (I0,1), à partir des variables contenues dans la table W0[4] :

- W0 = H'SSAA' (ex: H'1989')
- W1 = H'MMJJ' (ex: H'0728')
- W2 = H'HHMN' (ex: H'1305')
- W3 = H'SSXN' (ex: H'2205')

PL7-2		Formulaire programmation schéma à contacts						
N°	ZONE TEST						ZONE ACTION	OBSERVATIONS
		" D.E.T.E.C. . . . F.R.O.N.T. . . . "						
	I0,1	B1					B0	Sur demande de I0,1 : détection de front pour l'exécution de la mise à l'heure. (sur 1 seul cycle).
6							()	
0							B1	
							()	
L							()	
		" G.E.L. . . . H.O.R.O.D.A.T.E.U.R. . . . "						
		B0					SY3	Gel de l'horodateur pour la mise à l'heure
6							()	
1							()	
							()	
L							()	
		" M.I.S.E. . . . A. . . . L. . . . H.E.U.R.E. . . . "						
		B0					OPERATE	W0 → SW50 W1 → SW51 W2 → SW52 W3 → SW53
6							[W0 → SW50]	
2							OPERATE	
							[W1 → SW51]	
L							OPERATE	
							[W2 → SW52]	
							OPERATE	
							[W3 → SW53]	





Chapitre	Page
1 <i>Le langage grafcet PL7-2</i>	1/1
1.1 Rappel des principes du grafcet	1/2
1.2 Les symboles graphiques spécifiques au langage grafcet	1/3
1.3 Les objets spécifiques au grafcet	1/4
1.4 Représentation du grafcet	1/5
1.5 Les étapes et actions associées	1/12
1.6 Les transitions et les réceptivités associées	1/15
2 <i>Organisation de la tâche maître</i>	2/1
2.1 Description tâche maître	2/2
2.2 Le traitement préliminaire	2/3
2.3 Utilisation des bits système dans le traitement préliminaire	2/3
2.4 Le traitement séquentiel	2/7
2.5 Le traitement postérieur	2/8



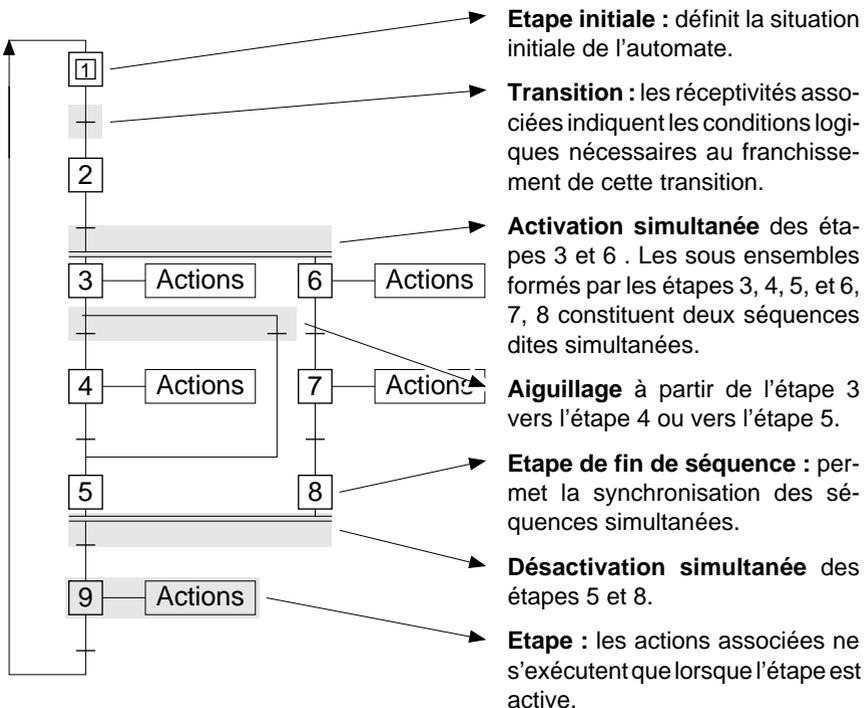
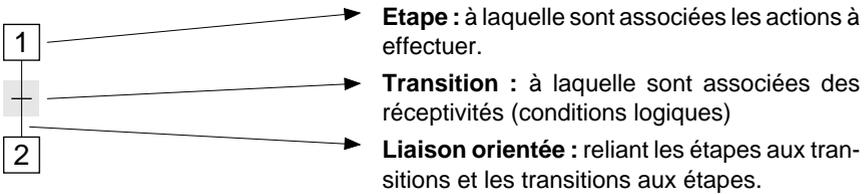


Sous chapitre	Page
1.1 Rappel des principes du grafcet	1/2
1.2 Les symboles graphiques spécifiques au langage grafcet	1/3
1.3 Les objets spécifiques au grafcet	1/4
1.4 Représentation du grafcet	1/5
1.4-1 La page grafcet	1/5
1.4-2 Règles d'écriture des graphes	1/6
1.4-3 Aiguillage et fin d'aiguillage	1/7
1.4-4 Activation et désactivation simultanée d'étapes	1/7
1.4-5 Utilisation des renvois	1/8
1.5 Les étapes et actions associées	1/12
1.5-1 Les étapes	1/12
1.5-2 Les actions associées aux étapes	1/12
1.5-3 Temps d'activité d'une étape	1/14
1.6 Les transitions et les réceptivités associées	1/15
1.6-1 Les transitions	1/15
1.6-2 Les réceptivités associées aux transitions	1/15
Ce chapitre se termine à la page	1/16

1.1 Rappel des principes du grafcet

Le grafcet permet de représenter graphiquement et de façon structurée le fonctionnement d'un automate séquentiel.

Cette description graphique du comportement séquentiel de l'automatisme, et des différentes situations qui en découlent, s'effectue à l'aide de symboles graphiques simples :



Les transitions et les liaisons orientées symbolisent les possibilités d'évolution des étapes actives.

Les actions associées aux étapes traduisent d'une façon générale "ce qui doit être fait" lorsque celles-ci sont actives. Elles décrivent en particulier les ordres qu'il faut transmettre à la partie opérative (processus à automatiser) ou à d'autres systèmes à automatiser. L'ensemble des étapes actives à un moment donné définit la situation du grafcet.

1.2 Les symboles graphiques spécifiques au langage grafcet

Désignation	Graphisme	Nb.max.	Commentaires
Étapes		96(*)	Symbolisent un état stable de l'automatisme. Bit étape associé : Xi Mot temps enveloppe : Xi,V Nombre maximum d'étapes actives simultanément : 16
Étapes		16	Symbolisent les étapes initiales actives en début de cycle après une initialisation ou une reprise à froid. Bit étape associé : Xi Mot temps enveloppe : Xi,V
Transitions		128	Séparent deux étapes et permettent le passage de l'une à l'autre. Nombre maximum de transitions validées simultanément : 24
Activations et désactivation simultanées d'étape			Permettent l'activation et la désactivation simultanée de 4 étapes maximum.
Aiguillage et fin d'éguillage			Nombre maximum de transitions en aval et en amont d'un aiguillage : 4
Renvois origine		42	Permettent d'assurer la continuité de la lecture du graphe.
Renvois destination		42	
Liaisons orientées			Permettent de relier une étape à une transition ou une transition à une étape.
Passage d'une étape à une autre (réceptivité)		128	Symbolise le passage d'une étape à une autre, apparaît automatiquement lors de la saisie d'une réceptivité.

(*) y compris les étapes initiales.

1.3 Les objets spécifiques au grafcet

L'utilisateur dispose d'objets bits associés aux étapes, de bits systèmes spécifiques au langage grafcet et d'objets mots indiquant le temps d'activité des étapes.

Désignation	Adresse	Description
Bits associés aux étapes	X_i	Etape i du grafcet principal $0 \leq i \leq 95$
Bits système associés au grafcet ⁽¹⁾	SY21	Provoque l'initialisation du grafcet
	SY22	Provoque la remise à zéro générale du grafcet
	SY23	Validation du prépositionnement et gel des graphes
Mots temps d'activité des étapes	X_i, V	Temps d'activité de l'étape i $0 \leq i \leq 95$ $0 \leq V \leq 9999$ secondes

⁽¹⁾ L'utilisation des bits systèmes est détaillée au sous chapitre 2.3 du présent intercalaire.

Bits associés aux étapes :

- Ils sont à 1 lorsque les étapes sont actives.
- Ces bits peuvent être testés dans toutes les tâches mais ne peuvent être écrits que dans le traitement préliminaire de la tâche maître (prépositionnement des graphes) Ces tests et actions sont programmés en langage à contacts.
- Les objets bits associés aux étapes ne sont pas indexables.

Mots temps d'activité :

- Ils sont incrémentés toutes les secondes et peuvent prendre la valeur 0 à 9999.
- Incrémentation du mot : pendant l'activité de l'étape associée.
- Comportement à l'activation et à la désactivation de l'étape :
 - à la désactivation : le contenu est remis à zéro,
 - à l'activation : incrémentation toutes les secondes.
- Ces objets mots ne sont pas indexables.

1.4 Représentation du grafcet

1.4-1 La page grafcet

La saisie du graphe dans la mémoire utilisateur s'effectue par pages repérées de P0 à P7. Chaque page est constituée de 8 colonnes (C0 à C7) et de 14 lignes (L0 à L13).

L'étude d'un automatisme par un grafcet peut être facilement effectué à l'aide d'un ensemble de formulaires conçus spécialement à cet effet. A chaque formulaire peut correspondre une page mémoire; l'étude grafcet faite sur les formulaires peut donc être directement saisie en mémoire.

8 pages
96 étapes maximum

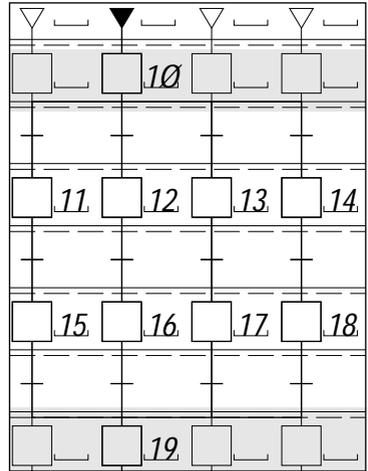


1.4-3 Aiguillage et fin d'aiguillage

Le nombre de transitions en aval d'un aiguillage ne doit pas dépasser 4.

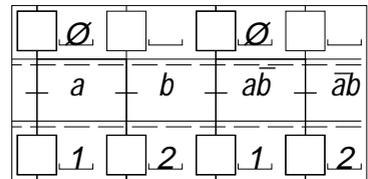
Le nombre de transitions en amont d'un aiguillage ne doit pas dépasser 4.

Un aiguillage doit généralement se terminer par une fin d'aiguillage.



Cas de transitions simultanément franchissables lors d'un aiguillage

Afin de ne pas franchir simultanément plusieurs transitions les réceptivités doivent être exclusives.

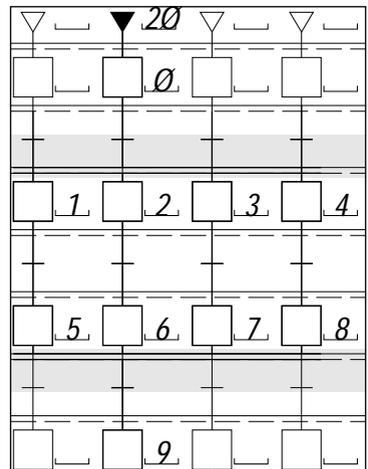


1.4-4 Activation et désactivation simultanées d'étapes

Le nombre d'étapes en aval d'une activation simultanée ne doit pas dépasser 4.

Le nombre d'étapes en amont d'une désactivation simultanée ne doit pas dépasser 4.

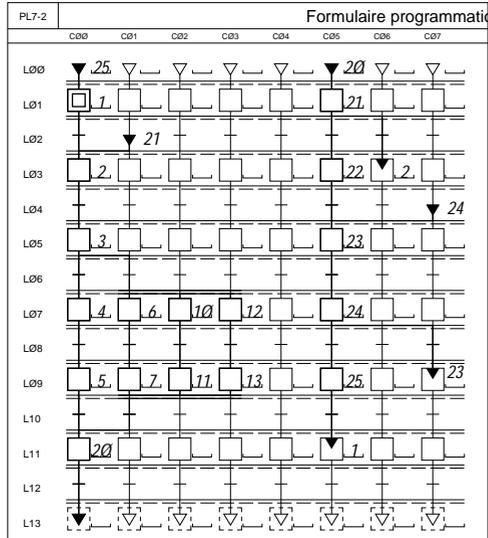
Une activation simultanée d'étapes doit généralement se terminer par une désactivation simultanée d'étapes.



1.4-5 Utilisation des renvois

Lorsque la liaison orientée reliant une transition à une étape ne peut être tracée de façon continue (par exemple dans le cas de tracé complexe ou de représentation sur plusieurs pages) on utilise des renvois pour assurer la continuité de la lecture du graphe.

Exemple: dans le grafcet ci-contre, le graphe situé à droite représente la continuité du graphe situé à gauche. Ces deux graphes peuvent être représentés indifféremment sur une même page ou sur deux pages différentes. Dans les deux cas, les renvois assurent la continuité de la lecture du grafcet.

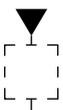


Types de renvois

Il existe deux types de renvoi :



- renvoi de destination : le numéro qui lui est associé indique l'étape de destination "où l'on va".



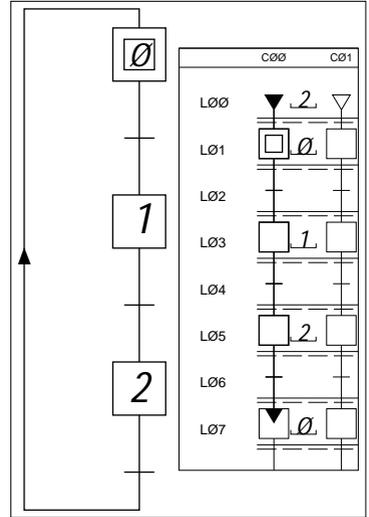
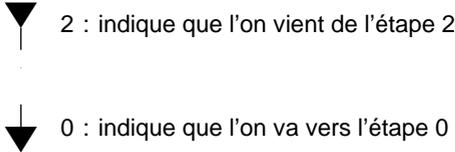
- renvoi d'origine : le numéro qui lui est associé indique l'étape d'origine "d'où l'on vient"

Toute liaison transition → étape peut être interrompue dans une page ou dans deux pages différentes. Le nombre maximum de renvois est de 42 renvois d'origine et de 42 renvois de destination.

Rebouclage d'un graphe

La saisie d'un rebouclage de graphe (liaison orientée vers le haut) s'effectue à l'aide de renvois.

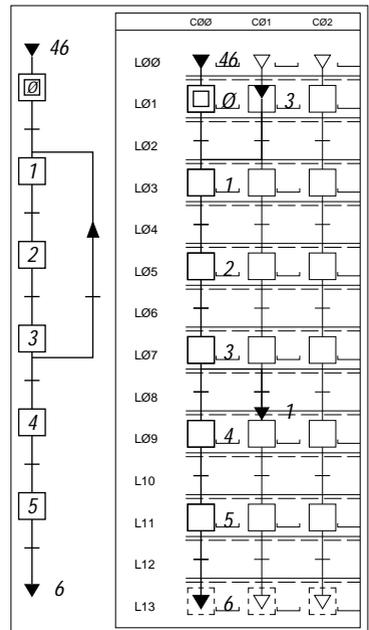
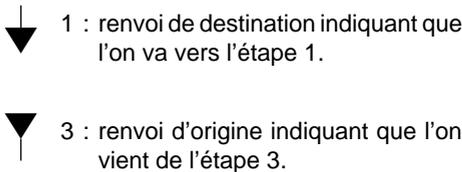
Dans l'exemple ci-contre le rebouclage de l'étape 2 vers l'étape 0 est symbolisé par les renvois suivants :



Renvois lors d'une reprise de séquence

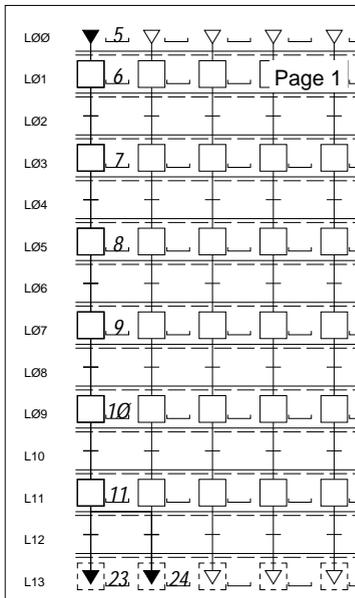
La saisie d'une reprise de séquence dans une page ou sur deux pages différentes s'effectue à l'aide de renvois.

L'exemple ci-contre montre une reprise de séquence de l'étape 3 vers l'étape 1.

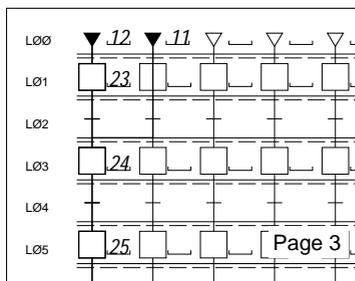


Renvois lors d'un aiguillage et d'une fin d'aiguillage

Les transitions et les renvois de destination (▼ 23 et ▼ 24) situés au début de l'aiguillage doivent être dans la même page.

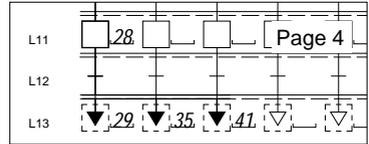


Les renvois d'origine : (▼ 12 et ▼ 11) doivent être dans la même page que les étapes de regroupement.

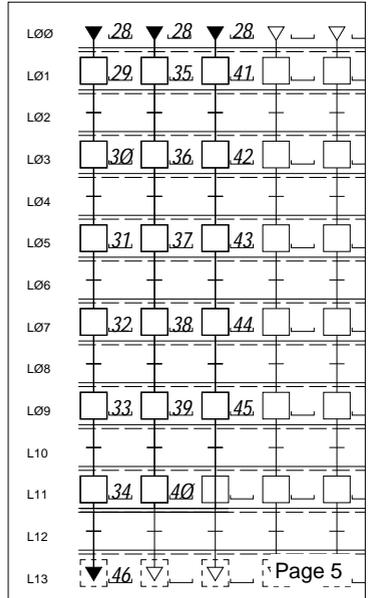


Renvois lors d'une activation et désactivation simultanées d'étapes

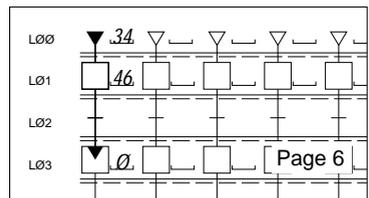
L'étape et la transition de divergence ainsi que les renvois de destination (↓ 29, ↓ 35, ↓ 41) doivent se trouver sur la même page.



Les étapes et la transition de convergence qui suit ces étapes doivent se trouver sur la même page que le renvoi de destination (↓ 46)



Lorsque plusieurs étapes convergent sur une seule transition, le renvoi d'origine porte le numéro de l'étape situé la plus à gauche (▽ 34)



1.5 Les étapes et actions associées

1.5-1 Les étapes

- Nombre maximum d'étapes (y compris les étapes initiales) : 96

A chaque étape est associé un bit étape X_i avec :

i : numéro d'étape (0 à 95)

$X_i = 0$ si l'étape est inactive

$X_i = 1$ si l'étape est active

- Deux étapes ne peuvent jamais avoir le même numéro.
- Nombre maximum d'étapes actives simultanément : 16

1.5-2 Les actions associées aux étapes

Les actions associées aux étapes peuvent se programmer de deux façons :

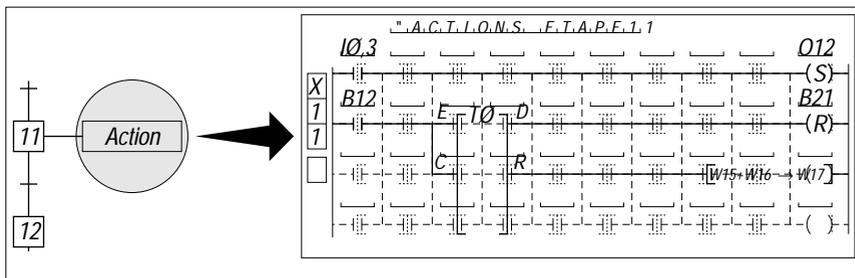
- soit par réseau de contacts associés à l'étape. Dans ce cas le réseau de contacts n'est scruté que si l'étape à laquelle il est associé est active.
- soit dans le traitement postérieur par utilisation du bit étape X_i . Dans ce cas la scrutation a lieu à chaque cycle. Pour des raisons de sécurité (scrutation du programme), il est préférable que les actions soient écrites dans le traitement postérieur.

Par réseau de contacts associé à l'étape

Ce réseau de contacts permet la programmation de 4 actions maximum qui peuvent être de type continu, conditionnel ou temporisé. IL est automatiquement étiqueté et porte le numéro de l'étape à laquelle il est associé.

Aucun saut de programme (JUMP) n'est possible dans ce type de réseau.

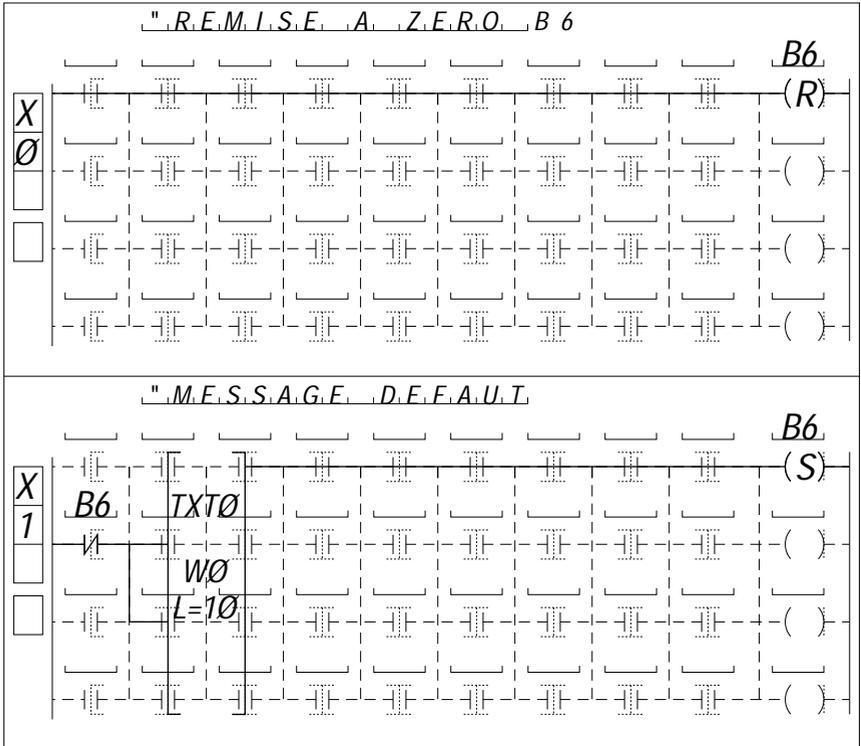
Un commentaire de 15 caractères maximum peut être associé à chaque réseau de contacts.



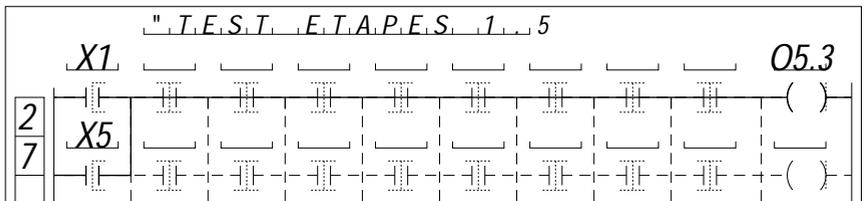
Les actions peuvent être asservies ou non à des conditions logiques et sont saisies sous forme de bobines à l'enclenchement (SET). Toute bobine (SET) doit obligatoirement être remise à zéro par une bobine (RESET). Ces bobines peuvent être sauvegardées ou non.

Les blocs fonction et blocs opération peuvent être programmés dans ce type de réseau. Toutefois le réseau de contacts n'étant scruté que si l'étape est active, il est impératif de créer un front montant pour piloter les blocs fonction possédant des entrées sur front (monostable, compteur,...)

L'exemple ci-contre montre la programmation du bloc texte TXT0. Le bit B6 étant remis à 0 lors de l'étape précédente X0. Le bloc texte envoie le message dès que l'étape X1 est active et met à 1 le bit B6.



Dans le traitement postérieur : par utilisation du bit étape Xi.



1.5-3 Temps d'activité d'une étape

A chaque étape i est associé un temps d'activité (mot X_i, V) permettant de connaître la durée d'activité d'une étape. La valeur de ce temps peut être testée dans une action, une transition, ou dans le traitement postérieur à l'aide d'un bloc opération "comparaison".

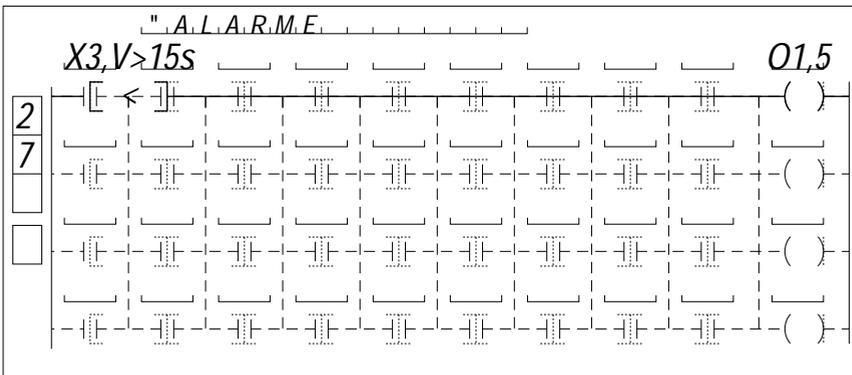
Ce temps d'activité X_i, V est mis à 0 sur :

- désactivation de l'étape,
- réinitialisation du grafcet (SY21 = 1),
- reprise à froid (SY0 = 1),
- remise à zéro du grafcet (SY22 = 1),
- prépositionnement (SY23 = 1).

Il est incrémenté toutes les secondes dès que l'étape est active et croît de 0 à 9999. Le temps d'activité d'une étape désactivée et activée simultanément continu à être incrémenté sans être remis à 0.

Exemple d'utilisation

Si le temps d'activité de l'étape 3 est supérieur à 15 secondes, activation de la sortie alarme O1,5.



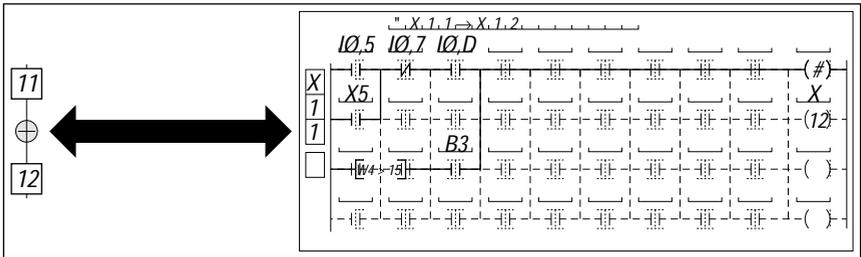
1.6 Les transitions et les réceptivités associées

1.6-1 Les transitions

- Nombre maximum de transitions : 128
- Nombre maximum de transitions validées simultanément : 24

1.6-2 Les réceptivités associées aux transitions

A chaque transition est associé un réseau de contacts permettant de saisir la réceptivité. Seul les opérations de test sont possibles (contact, bloc opération comparaison).



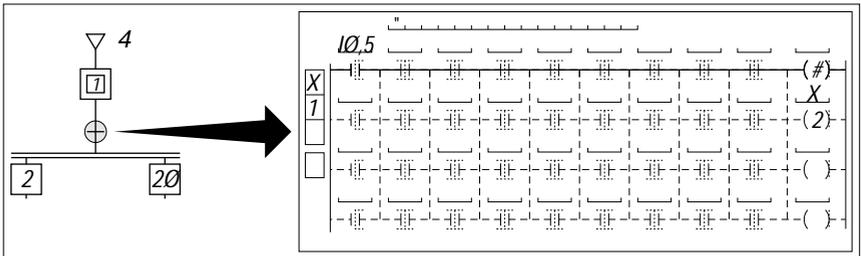
Le réseau de contacts est automatiquement étiqueté et porte le numéro de l'étape "d'où l'on vient" (ex. X11).

Le franchissement de la transition est symbolisé par une bobine portant le numéro de l'étape "où l'on va" (ex. X12). Cette bobine apparait systématiquement lors de la saisie.

La scrutation de ce réseau n'est effective que si l'étape située en amont est active.

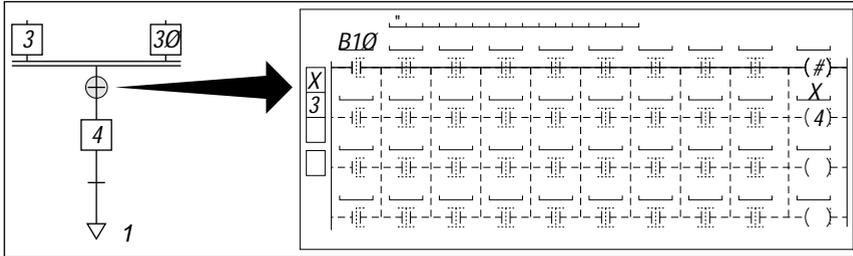
Cas d'une activation simultanée

Dans ce cas, la bobine $\overline{(\#)}$ porte le numéro de l'étape aval la plus à gauche.

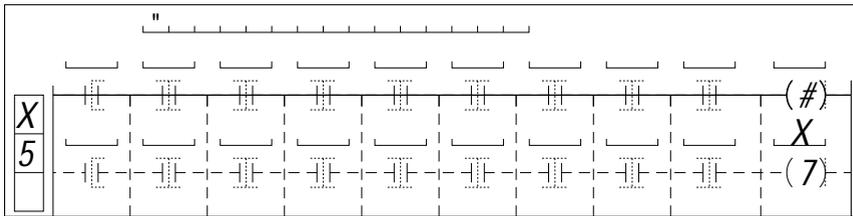


Cas d'une désactivation simultanée

Dans ce cas l'étiquette du réseau de contacts porte le numéro de l'étape amont la plus à gauche.



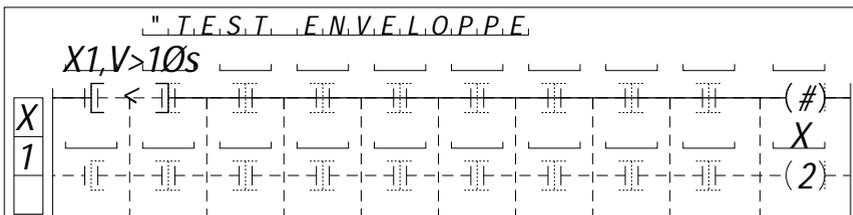
Cas d'une réceptivité toujours vraie



Cas d'une réceptivité non écrite

Le passage à l'étape suivante est impossible.

Utilisation du temps d'activité d'une étape dans une réceptivité





Sous chapitre	page
2.1 Description tâche maître	2/2
2.2 Le traitement préliminaire	2/3
2.3 Utilisation des bits système dans le traitement préliminaire	2/3
2.3-1 Initialisation du grafcet : SY21	2/4
2.3-2 Remise à zéro du grafcet : SY22	2/5
2.3-3 Prépositionnement du grafcet : SY23	2/6
2.3-4 Figeage d'une situation	2/7
2.4 Le traitement séquentiel	2/8
2.4-1 Caractéristiques	2/8
2.4-2 Principe d'évolution	2/8
2.5 Le traitement postérieur	2/9
Ce chapitre se termine à la page	2/10

2.1 Description tâche maître

Un programme écrit en langage grafcet comporte 3 traitements consécutifs : le traitement préliminaire, le traitement séquentiel et le traitement postérieur. Leur scrutation s'effectue selon le cycle de base suivant :

Gestion système

Le système assure implicitement :

- mise à jour des bits et mots système,
- surveillance de l'automate
- traitement des requêtes du terminal,
- routage de la messagerie.

Acquisition des entrées

Acquisition de l'état physique des entrées de l'automate (valeurs figées pendant le traitement).

Traitement préliminaire

Il permet de traiter :

- Les initialisations sur reprise secteur ou défaillance,
- le prépositionnement du graphe,
- les modes de marche de l'application,
- la logique d'entrée.

Traitement séquentiel

Il permet de traiter l'ossature séquentielle de l'application et donne accès au traitement des réceptivités et des actions directement associées aux étapes.

Traitement postérieur

Il permet de traiter :

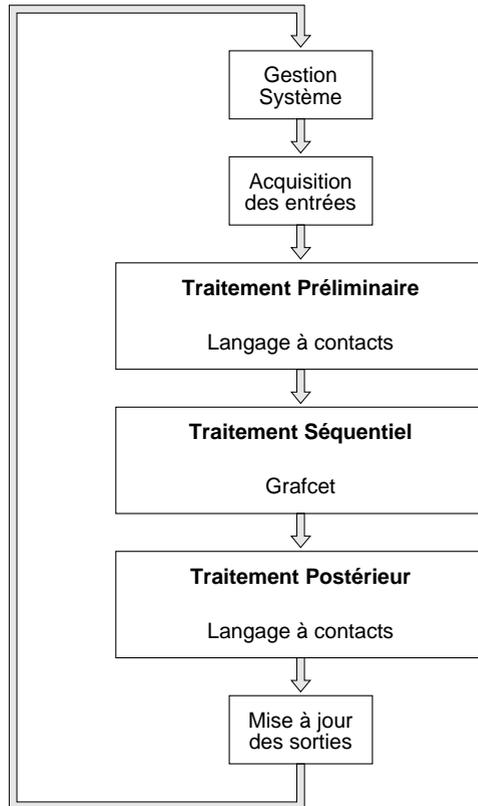
- la logique de sortie,
- les surveillances et sécurités indirectes spécifiques aux sorties.

Mise à jour des sorties

mise à jour de l'état physique des sorties (valeurs figées pendant le traitement).

Rappel : Les sauts de programme ne peuvent se faire qu'à l'intérieur d'un même traitement (préliminaire ou postérieur).

Un réseau de contacts du traitement préliminaire et un réseau de contacts du traitement postérieur peuvent avoir la même étiquette.



2.2 Le traitement préliminaire

Saisi en langage à contacts, le traitement préliminaire est scruté dans sa totalité du haut vers le bas.

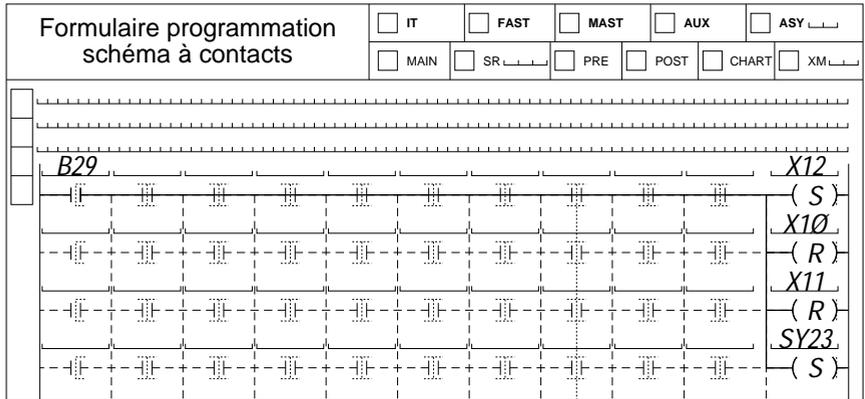
Exécuté avant les traitements séquentiel et postérieur, il permet de traiter tous les évènements ayant une influence sur ces derniers :

- gestion des reprises secteur et réinitialisation,
- demande de modes de marche
- remise à zéro ou prépositionnement des graphes.

C'est donc uniquement dans le traitement préliminaire que l'on agira sur les bits associés aux étapes (mise à 0 ou 1 des bits étapes Xi).

Exemple :

Ces actions réalisent un prépositionnement du grafcet.



2.3 Utilisation des bits système dans le traitement préliminaire

Les bits système associés au grafcet étant classés numériquement par ordre de priorité (SY21 à SY23), lorsque plusieurs d'entre eux sont simultanément mis à 1 dans le traitement préliminaire, ils sont traités un par un dans un ordre croissant (un seul est effectif par tour de scrutation). Ces bits sont effectifs au début du traitement séquentiel.

Important : les bits système SY21, SY22 et SY23 étant remis à 0 automatiquement par le système, l'écriture de ces bits ne devra être faite que sous la forme de bobine SET.

2.3-1 Initialisation du grafçet : SY21

En présence d'un évènement particulier (défaut de processus, traitement d'un arrêt d'urgence,...), il peut être nécessaire d'effectuer une initialisation du grafçet par la mise à l'état 1 dans le traitement préliminaire du bit système SY21.

Normalement à l'état 0, la mise à l'état 1 provoque :

- La désactivation des étapes actives,
- l'activation des étapes initiales.

Mis à l'état 1	Mis à l'état 0
<ul style="list-style-type: none"> • Par mise à l'état 1 de SY0 • Par le programme utilisateur • Par initialisation (INIT) du terminal 	<ul style="list-style-type: none"> • Par le système au début du traitement séquentiel

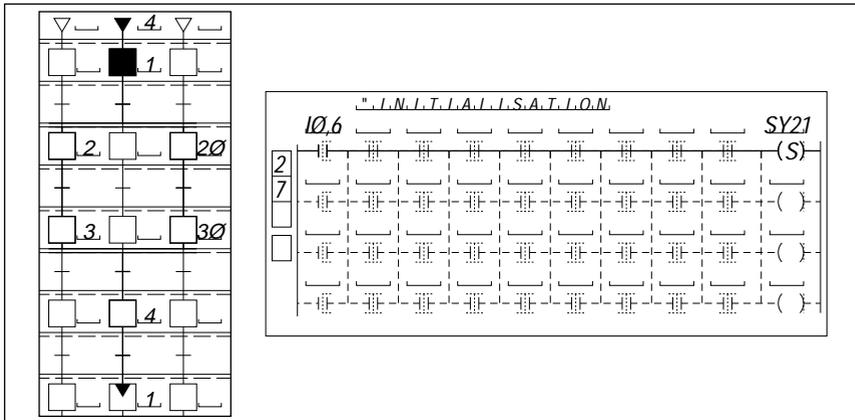
Utilisation

- Doit être écrit à 1 uniquement dans le traitement préliminaire
- Géré par le système, il est interdit de le remettre à 0 par le programme ou le terminal.

Exemple

Le programme ci-dessous montre l'initialisation du grafçet , provoqué par un évènement extérieur (I0,6).

Dans le grafçet cela se traduit par l'activation de l'étape initiale 1.



2.3-2 Remise à zéro du grafcet : SY22

Provoqué par la mise à l'état 1 du bit système SY22, la remise à zéro du grafcet entraine la non scrutation de celui-ci par l'automate; seuls les traitements préliminaire et postérieur sont exécutés.

Normalement à l'état 0, la mise à l'état 1 de SY22 provoque la désactivation des étapes actives de l'ensemble du traitement séquentiel.

Mis à l'état 1	Mis à l'état 0
<ul style="list-style-type: none"> Par le programme utilisateur 	<ul style="list-style-type: none"> Par le système au début du traitement séquentiel

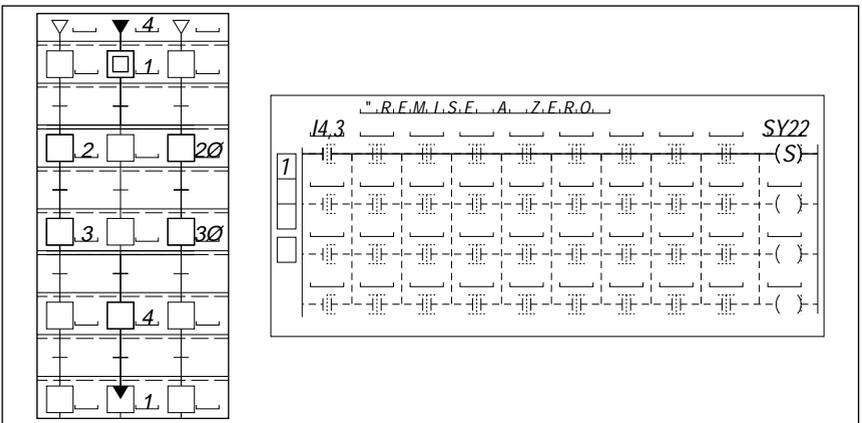
Utilisation

- Doit être écrit à 1 uniquement dans le traitement préliminaire
- Géré par le système, il est interdit de le remettre à 0 par le programme ou le terminal.

Pour redémarrer le traitement séquentiel dans une situation donnée, l'utilisateur devra prévoir selon son application une procédure d'initialisation ou de prépositionnement du grafcet.

Exemple

Le programme ci-dessous montre la remise à zéro générale du grafcet provoquée par un événement (I4,3). Dans le grafcet cela se traduit par la désactivation de la totalité des étapes.



2.3-3 Prépositionnement du grafcet : SY23

Il peut être nécessaire de prépositionner un grafcet dans une situation différente de la situation initiale lors du passage d'un fonctionnement marche normale en marche spécifique ou de l'apparition d'un incident (exemple : défaut provoquant une marche dégradée).

Normalement à l'état 0, la mise à l'état 1 de SY23 provoque la validation du prépositionnement du graphe.

Mis à l'état 1	Mis à l'état 0
<ul style="list-style-type: none"> Par le programme utilisateur 	<ul style="list-style-type: none"> Par le système au début du traitement séquentiel

Le prépositionnement intervenant sur le déroulement normal du cycle de l'application, il doit donc être effectué avec précaution.

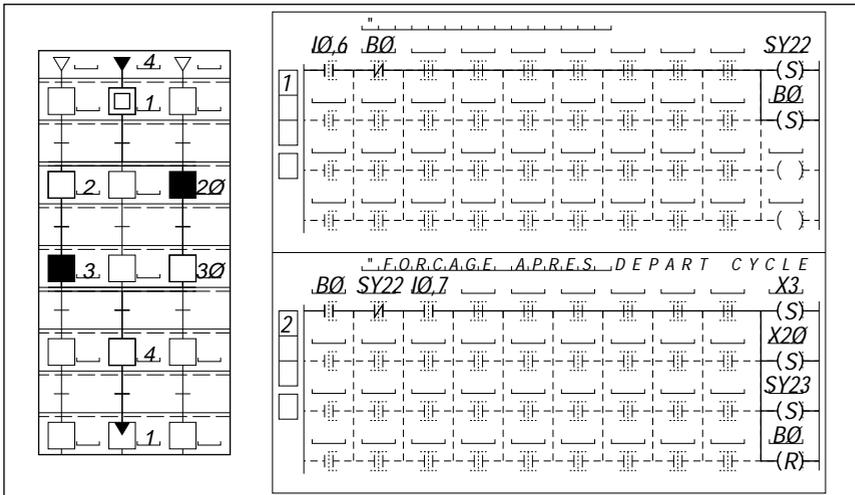
Il est alors nécessaire de procéder de la manière suivante :

- Remettre éventuellement à 0 la totalité du grafcet par le bit système SY22.
- Prépositionner les étapes à activer ou désactiver (ex. SET Xi ou RESET Xi).
- Valider impérativement le prépositionnement par le bit système SY23.

Exemple

Le programme ci-dessous montre la remise à 0 du graphe (SY22) sur un front montant de I0,6. Le bit système SY22 est remis à 0 par l'automate. Une action sur I0,7 provoque l'activation des étapes 3 et 20 et la validation du prépositionnement par le bit SY23. La validation doit être obligatoirement réalisée après les opérations de prépositionnement (SET de SY23); ce bit est remis systématiquement à 0 par le système.

Dans le grafcet cela se traduit par la désactivation de toutes les étapes et l'activation des étapes 3 et 20.



2.3-4 Figeage d'une situation

Figeage du graphe en situation initiale

Le maintien du bit SY21 à l'état 1 permet de figer le graphe en position initiale; dans ce cas aucune évolution du graphe n'est possible.

Figeage du graphe dans une situation vide

Le maintien du bit SY22 à l'état 1 permet de figer le graphe dans une situation vide; dans ce cas aucune étape n'est active

Figeage du graphe dans une situation donnée

Le maintien du bit SY23 à l'état 1 permet de figer le graphe dans une situation donnée; aucune évolution du graphe n'est alors possible.

2.4 Le traitement séquentiel

2.4-1 Caractéristiques

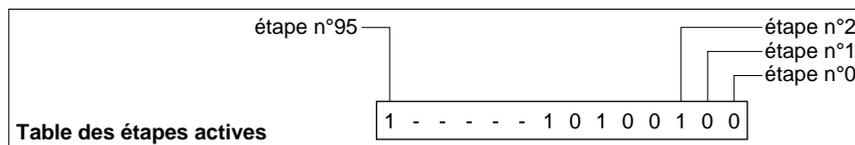
Ce traitement permet d'une part de définir la structure séquentielle de l'application et d'autre part son interprétation (définition des réceptivités associées aux transitions et des actions associées aux étapes).

Il comprend le graphe principal organisé en 8 pages avec 96 étapes et 128 transitions maximum.

Au sein du graphe principal, plusieurs graphes non connexes peuvent être programmés et se dérouler simultanément.

2.4-2 Principe d'évolution

L'automate tient à jour en permanence une table des étapes actives à un instant donné puis effectue les opérations suivantes dans l'ordre indiqué ci-dessous :



Détermination de toutes les transitions validées.



Test de l'ensemble des réceptivités associées aux transitions validées.



Si la condition de transition est vraie : activation de l'étape aval, désactivation de l'étape amont et mise à jour de la table des étapes actives.

Exécution des actions associées aux étapes actives dans l'ordre croissant de leur numéro.

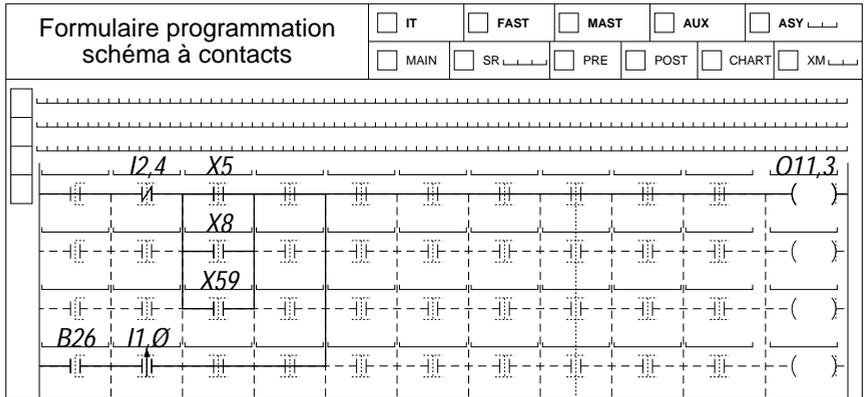
Remarques

- L'état des entrées est stable pendant toute la durée de l'interprétation.
- Le fonctionnement de la tâche rapide en association avec une tâche maître écrite en langage grafset est décrit dans "Généralités PL7-2 - intercalaire A - chapitre 5"

2.5 Le traitement postérieur

Saisi en langage à contacts, le traitement postérieur est scruté de haut en bas . Ce traitement est le dernier exécuté avant l'activation des sorties et permet de programmer la logique de sortie :

- Définition des actions associées ou non à des étapes,
- Gestion des sécurités inhérentes à ces actions (prise en compte d'un arrêt d'urgence, capteur de surcourse),
- Gestion des fonctions d'automatisme (temporisateur, compteur,...)



Actions associées au grafcet

L'activité d'une étape du grafcet peut être testée à l'aide d'un contact dans le traitement postérieur afin de piloter les actions associées. Ceci permet d'assurer l'unicité de la commande (une seule bobine commandée par plusieurs étapes). De même le traitement postérieur permet de compléter les consignes émises par le traitement séquentiel en intégrant à l'équation d'une sortie les modes de marche et d'arrêt et les sécurités indirectes à l'action.

Exemple

I2,4 : sécurité indirecte spécifique au pilotage de la sortie O11,3.

B26 : bit interne résultat de la logique d'entrée traitant des modes de marche et d'arrêts.

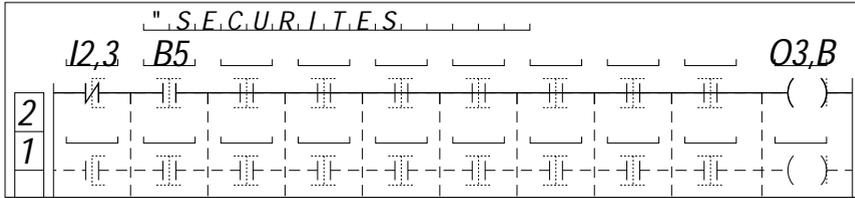
I1,0 : bouton poussoir

D'une manière générale il est recommandé de programmer les actions agissant directement sur le process dans le traitement postérieur.

Actions indépendantes du grafcet

Le traitement postérieur permet également de programmer les sorties indépendantes du traitement séquentiel.

L'exemple ci-contre montre la commande de la bobine O3,B si le bit d'entrée I2,3 est à 0 et si le bit interne B5 est à 1.



Remarque

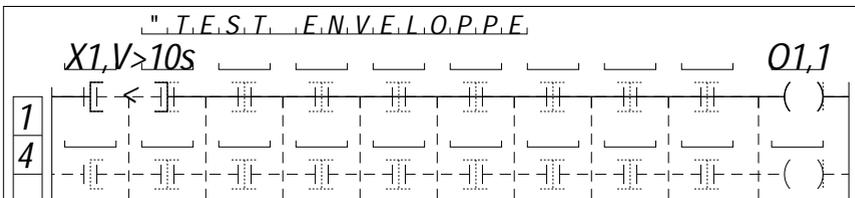
Les capteurs concernant la sécurité directe ne doivent en aucun cas être traités en direct par l'automate mais doivent agir directement sur les pré-actionneurs comme le fait un arrêt d'urgence qui coupe le contacteur principal. Il est toutefois nécessaire de transmettre ces informations à l'automate afin de les prendre en compte dans le traitement de l'automatisme.

Contrôle de l'exécution du grafcet

Dans certaines applications, il peut s'avérer nécessaire de contrôler le bon déroulement du grafcet en testant le temps d'activité de certaines étapes.

Le test de ce temps d'activité s'effectue par comparaison soit à une valeur minimum soit à une valeur maximum définie par l'utilisateur. L'exploitation du défaut est laissée au choix de l'utilisateur (signalisation, procédure particulière de fonctionnement, édition de messages).

L'exemple ci-contre montre la comparaison du temps d'activité de l'étape 1 à une valeur définie par l'utilisateur (10s) et la signalisation d'un défaut à l'aide d'un bit de sortie O1,1.



Règles pratiques

- Une sortie ne doit pas être commandée simultanément dans le traitement séquentiel et le traitement postérieur.
- Une sortie ne doit pas être commandée plusieurs fois dans le traitement postérieur.
- Il est impossible d'effectuer un saut de programme (JUMP) d'un traitement à un autre.



Chapitre	Page
1 Organisation des programmes application	1/1
1.1 Traitement analogique	1/2
1.2 Fonction communication	1/2
1.3 Tâche rapide	1/3
2 Traitement des modes de marche et d'arrêt	2/1
2.1 Le guide GEMMA	2/2
2.2 Structure des automates de la série TSX 7	2/4
2.3 Analyse des modes de marche et d'arrêt	2/7
2.4 Structuration du traitement préliminaire	2/7
2.5 Grafcet complet	2/12
3 Aide au diagnostic	3/1
3.1 Comportement des sorties en cas de disfonctionnement automate	3/2
3.2 Diagnostic des défauts	3/3
3.3 Traitement des défauts d'entrées/sorties	3/4
4 Fonctions spécifiques au TSX 17-20	4/1
4.1 Protection des programmes des cartouches EPROM et EEPROM	4/2
4.2 Configuration des entrées/sorties spécifiques	4/3
4.3 Modules de visualisation	4/4



Sous chapitre	Page
1.1 Traitement analogique	1/2
1.2 Fonction communication	1/2
1.3 Tâche rapide	1/3
Ce chapitre se termine à la page	1/4

Cahier des charges, analyse et structuration d'un automate sont les garants de la réussite de la mise en route et de l'exploitation d'une installation pilotée à partir d'automates programmables.

Chaque fonction doit être dissociée des autres, et sa spécificité va définir ses entrées/sorties éventuellement affectées.

1.1 Traitement analogique

Utilisé généralement dans les processus de régulation, ce type de traitement est caractérisé par sa "lenteur" et sa périodicité. Les coupleurs adaptés à ce type de traitement sont :

- les interfaces d'entrées/sorties analogiques TSX ADT./AST 200 pour TSX 47–J/10/20.
- les coupleurs analogiques TSX AEM./ASR., pour TSX 47-20,
- les modules d'entrées/sorties analogiques TSX AEG/ASG, pour TSX 17–20.

1.2 Fonction de communication

La communication automate ↔ périphérique s'effectuent :

- soit par l'intermédiaire de la prise terminal,
- soit par l'intermédiaire de liaisons séries.

Elle permet généralement :

- l'échange d'un nombre important de données,
- le dialogue avec un opérateur par un écran-clavier.

Dans le premier cas, l'échange dure quelque centaine de millisecondes (voir exemple ci-après). Dans le second type d'application la réponse d'un opérateur est de l'ordre de la seconde.

Exemple d'envoi de message

Soit un message de 200 caractères à émettre à la vitesse de 9600 bits par seconde dont les données ont le format suivant :

- 1 bit de start, 8 bits de données, 1 bit de parité et 1 bit de stop (total = 11 bits).
Le temps d'émission vaut : $200 \times 11 : 9600 \times 1000 = 230\text{ms}$.

Les coupleurs adaptés à ce type de fonction sont :

- les coupleurs TSX SCM (modules liaisons asynchrones), pour TSX 47-20,
- le coupleur TSX SCG 113 (module communication chaîne de caractères), pour TSX 17-20.

Ces coupleurs ou modules gèrent les échanges de manière autonome. Seuls leur configuration et la programmation des messages restent à la charge de l'utilisateur.

1.3 Tâche rapide

La création d'une tâche rapide est nécessaire si des évolutions rapides d'entrées tout ou rien doivent être surveillées. Des modules d'entrées/sorties vont être affectés à cette tâche. Le traitement dans cette tâche doit être le plus court possible. Par exemple, une série de conditions et de tests dont les éléments sont mis à jour ou calculés en tâche maître (donc à fréquence plus faible) ne doivent figurer que sous forme d'un résultat représenté par l'état d'un bit interne.

De la sorte, le temps d'exécution de la tâche rapide est beaucoup plus faible.

Conclusion

Les remarques mentionnées ci-dessus amènent ainsi à définir des ordres de grandeur pour les périodes de tâches :

- tâche rapide 1 à 2 ms,
- tâche maître 40 à 80 ms (déclenchement du "chien de garde" à 150 ms).

D



Sous chapitre	Page
2.1 Le Guide GEMMA	2/2
2.1-1 Introduction	2/2
2.1-2 Grille simplifiée	2/3
2.2 Structure des automates de la série TSX7	2/4
2.2-1 Structure multitâche	2/4
2.2-2 Structure de la tâche maître	2/5
2.2-3 Bits système	2/6
2.2-4 Bits et mots associés aux étapes GRAFCET	2/6
2.3 Analyse des modes de marches et d'arrêts	2/7
2.4 Structuration du traitement préliminaire	2/7
2.4-1 Traitement suite à une mise sous tension	2/8
2.4-2 Traitement suite à un changement de mode	2/10
2.4-3 Organigramme du traitement préliminaire	2/11
2.4-4 GRAFCET de Conduite	2/12
2.5 GRAFCET Complet	2/13
Ce chapitre se termine à la page	2/14

2.1 Le guide GEMMA

2.1-1 Introduction

Le GEMMA ("Guide d'Etude des Modes de Marches et d'Arrêts", proposé par l'ADEPA), est une méthode qui permet, dès l'élaboration du cahier des charges, de définir les différents modes de fonctionnement.

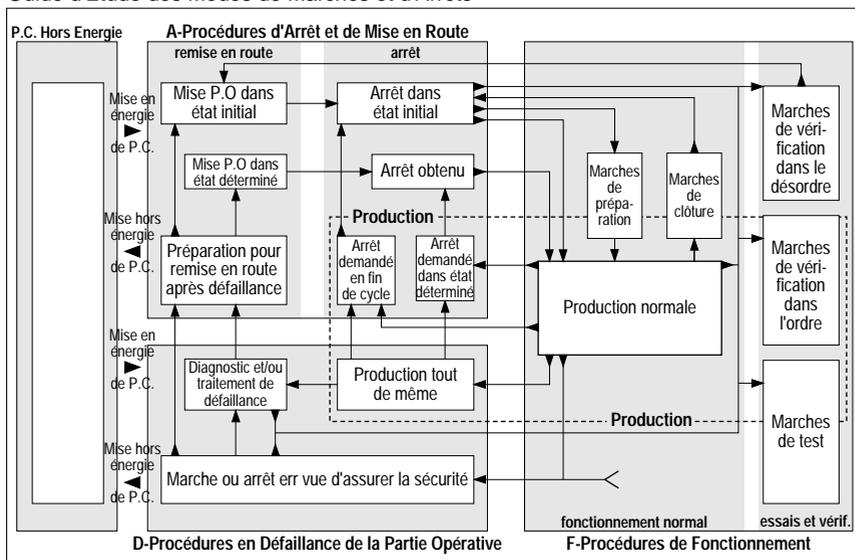
Cette méthode, encore peu utilisée, apporte de nouveaux concepts, définit un vocabulaire précis et propose une visualisation graphique.

Ce guide permet d'effectuer :

- le recensement des différents modes envisagés,
- la détermination des conditions provoquant le passage d'un mode à l'autre.

Il ne permet pas de passer à la phase de réalisation, c'est à dire transcription en langage automate.

GEMMA Guide d'Etude des Modes de Marches et d'Arrêts



Afin de répondre à ces besoins, et dans l'esprit des concepts de ce guide, Telemecanique propose, pour les automates de la série TSX7 :

- une grille simplifiée (voir page ci-contre),
- une structure d'accueil spécifique (voir chapitre 2.2-1),
- une démarche conduisant à la structure du programme traitant des modes de marches et d'arrêts (voir chapitres 2.2-2 à 2.2-4).

2.1-2 Grille simplifiée

Etablie à partir du guide GEMMA, la grille ci-dessous retient les principaux modes de marches et d'arrêts pris en compte par l'automate et se veut volontairement simplifiée par rapport aux rectangles des différents états pris par l'automatisme.

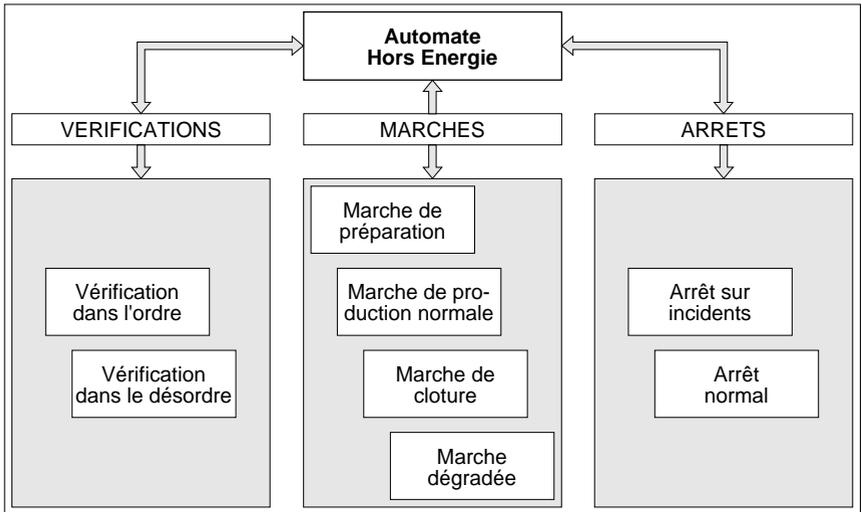
Trois grandes familles sont à distinguer :

- Vérifications,
- Marches,
- Arrêts.

Une fois les modes retenus, il faut alors détailler l'analyse jusqu'à définir les causes voulues ou subies provoquant le passage d'un mode à un autre. La mise sous tension de l'automate étant un cas particulier à examiner.

En fonction des causes, le mode en cours peut être abandonné pour passer dans un autre mode, il y a donc action sur le déroulement du traitement séquentiel.

Le vocabulaire utilisé est celui qui figure dans le guide GEMMA, la définition des principaux termes employés est rappelée en annexe.



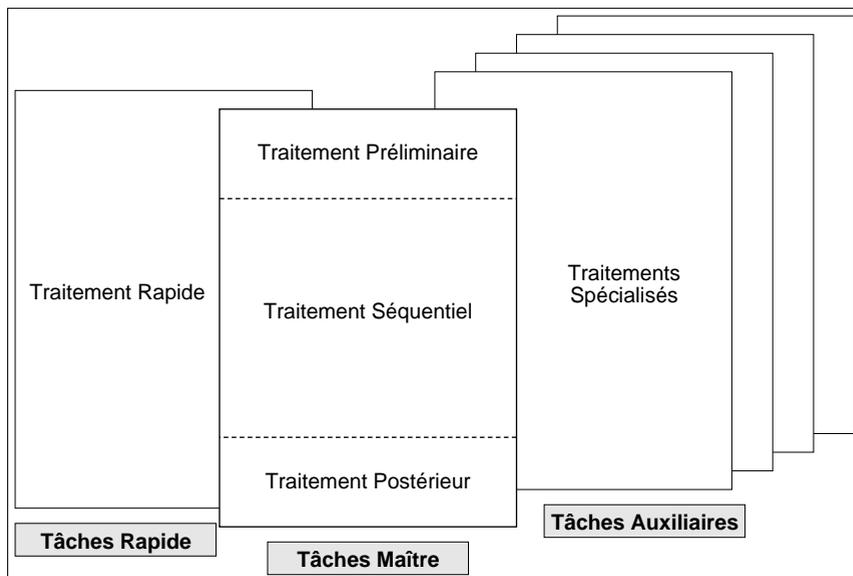
D

2.2 Structure des automates de la série TSX7

2.2-1 Structure multitâche

Afin de pouvoir traiter de façon optimale l'ensemble des besoins en automatismes (mesure, régulation, dialogue, communication, commande séquentielle), les automates TSX7 sont structurés en un système multitâche (bitâche dans le cas des TSX 17-20/27/47 : tâche maître et tâche rapide).

A la mise sous tension de l'automate, la seule tâche utilisateur active est la tâche maître. C'est cette tâche qui est dédiée à la gestion du traitement séquentiel et aux changements de modes de fonctionnement.



TSX7	Automates séquentiels Langages PL7-1 et PL7-2	Automates Multifonctions Langage PL7-3
	"Système bitâche"	"Système multitâche"
Tâche rapide	1	1
Tâche maître	1	1
Tâches auxiliaires	0	de 1 à 4

2.2-2 Structure de la tâche maître

Destinée principalement à la commande séquentielle, lors de l'utilisation du GRAFCET, la tâche maître se structure en trois parties, facilitant la gestion de l'automatisme et de ses modes de marches.

Le traitement Préliminaire

Effectué en début de la tâche maître, après la mise à jour de la mémoire image des entrées, le traitement préliminaire est exécuté à chaque tour de cycle et permet de traiter :

- les initialisations sur reprise secteur ou défaillance,
- la prise en compte de l'analyse d'une demande de changement de mode de marche (prédisposition ou action sur les graphes),
- la logique d'entrée (traitement combinatoire du programme application).

Le traitement Séquentiel

Permet la transcription graphique et la gestion des GRAFCET. Il donne accès au traitement des réceptivités associées aux transitions et des actions directement associées aux étapes.

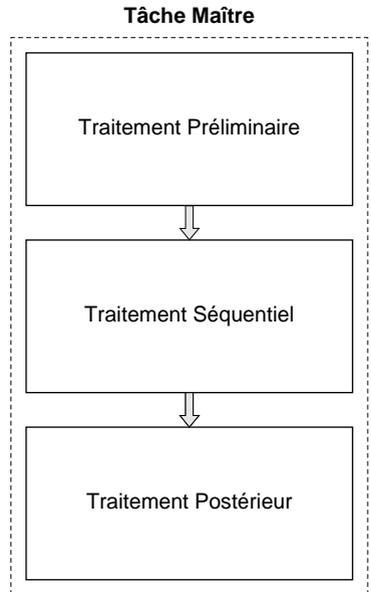
Seules les étapes actives et le traitement qui leur est associé sont scrutés.

Le traitement Postérieur

Scruté en fin de la tâche, avant la mise à jour de l'état physique des sorties, ce traitement est exécuté à chaque tour de cycle et permet de traiter la logique de sortie avec prise en compte :

- de l'ensemble des ordres émanant des deux traitements précédents,
- des sécurités indirectes spécifiques aux sorties.

A cette structure d'accueil, pour le traitement des modes de marches, il est important aussi d'y associer les moyens dont sont dotés les automates TSX 7 pour agir sur le déroulement du traitement séquentiel (bits et mots système accessibles par programme utilisateur).



2.2-3 Bits système

Afin de traiter les mises sous tension, les changements de modes, les actions sur le graphe, l'utilisateur dispose d'informations fournies par le système et définies dans des bits (SY0, SY1, SY21, SY22 et SY23). Ces bits doivent être exploités dans le traitement préliminaire.

L'écriture de ces bits doit être faite uniquement par "bobines : SET" (S). Ils sont remis à 0 automatiquement par le système.

2.2-4 Bits et mots associés aux étapes GRAFCET

A chaque étape GRAFCET programmée, est associé un bit Xi reflet de l'état de l'étape. Ce bit est mis à jour en même temps que les tables GRAFCET, pendant le traitement séquentiel. Puis il est pris en compte dans les équations pilotant les sorties dans le traitement postérieur, ou dans tout autre opération se déroulant dans le programme application.

Ces bits sont accessibles :

- **en lecture** (test), dans toute entité du programme application,
- **en écriture** (action), **uniquement dans le traitement préliminaire** afin d'assurer une mise à jour des tables dans le même tour de cycle.

A chaque étape GRAFCET est associé aussi un mot Xi,V contenant le temps d'activité de l'étape, en secondes.

2.3 Analyse des modes de marches et d'arrêts

Le logiciel PL7-2 permet la prise en compte des trois grandes familles de modes de marches et d'arrêts définies par le GEMMA (voir page 2/3) :

- Vérifications,
- Marches,
- Arrêts.

Ces différents modes peuvent être obtenus à partir du GRAFCET, par les actions suivantes :

- Initialisation du GRAFCET,
- Prépositionnement d'étapes,
- Maintien de situation,
- Gel de graphes.

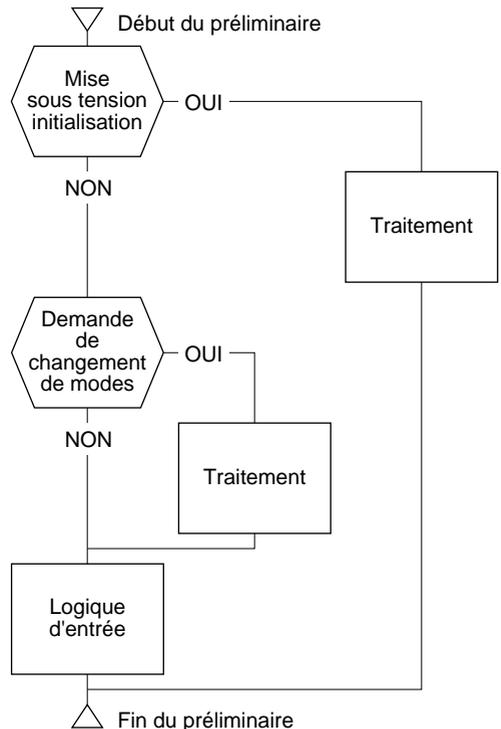
L'utilisation du traitement préliminaire et des bits système permet d'assurer la gestion de modes de marches et d'arrêts, sans compliquer ni alourdir le programme utilisateur.

2.4 Structuration du traitement préliminaire

Le synoptique ci-contre montre la structure à donner au préliminaire afin d'effectuer par ordre d'importance chaque traitement en cas de :

- Mise sous tension,
- Changement de modes de marches.

La logique d'entrée est le traitement combinatoire du programme application, effectué en préliminaire.



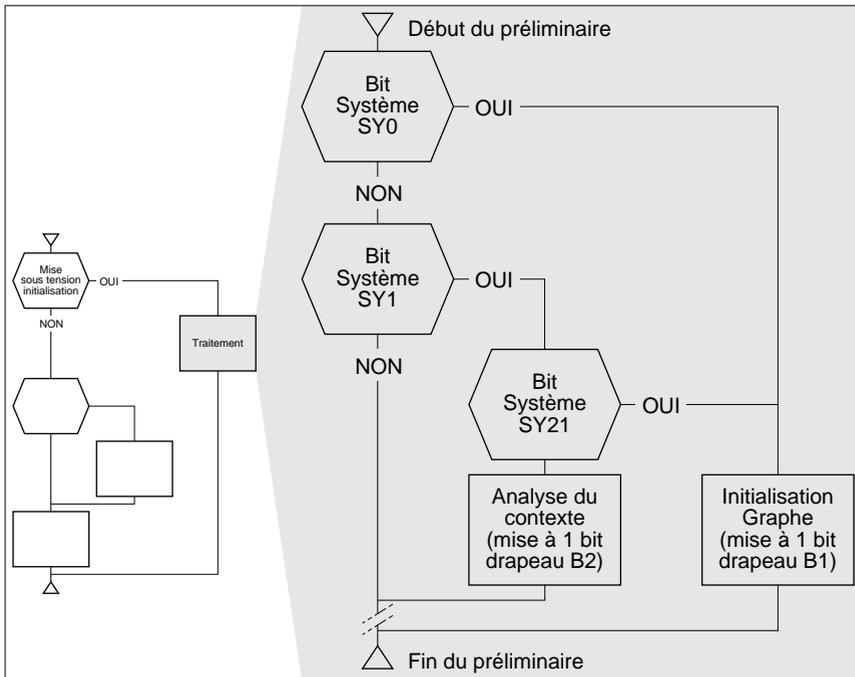
2.4-1 Traitement suite à une mise sous tension

Deux modes sont à analyser et traiter :

- Reprise à froid : redémarrage de l'automate avec perte du contenu de la mémoire de données. Le système met le bit SY0 à 1 et effectue automatiquement une initialisation du GRAFCET (SY21=1).
- Reprise à chaud : redémarrage de l'automate avec le contenu de la mémoire de données en état. Dans ce cas, l'installation peut nécessiter un redémarrage :
 - sur état initial,
 - sur l'état présent avant la coupure secteur,
 - sur d'autres états du graphe.

Le traitement doit contenir le test de l'état de SY21 (défini par le programme utilisateur) et ensuite prendre en compte le comportement désiré (initialisation, maintien de situation, prépositionnement, gel de graphes).

L'analyse des bits système conduit à l'élaboration de l'organigramme cidessous.



Le tableau ci-dessous décrit le comportement vis à vis du GRAFCET, suivant l'état du bit SY21 (défini par programme utilisateur).

Etat SY21 (défini par programme)	Comportement vis à vis du GRAFCET (Actions nécessaires)			
	Initialisation	Prépositionnement	Maintien de Situation	Gel de Graphe
SY21 = 0	-	<ul style="list-style-type: none"> • Mémorisation de la demande par bit drapeau Bi • SET SY22 • SET (Xi,SY23) 	-	<ul style="list-style-type: none"> • SET SY23
SY21 = 1	<ul style="list-style-type: none"> • Mémorisation de la demande par bit drapeau Bi 	-	-	-

2.4-2 Traitement suite à un changement de mode

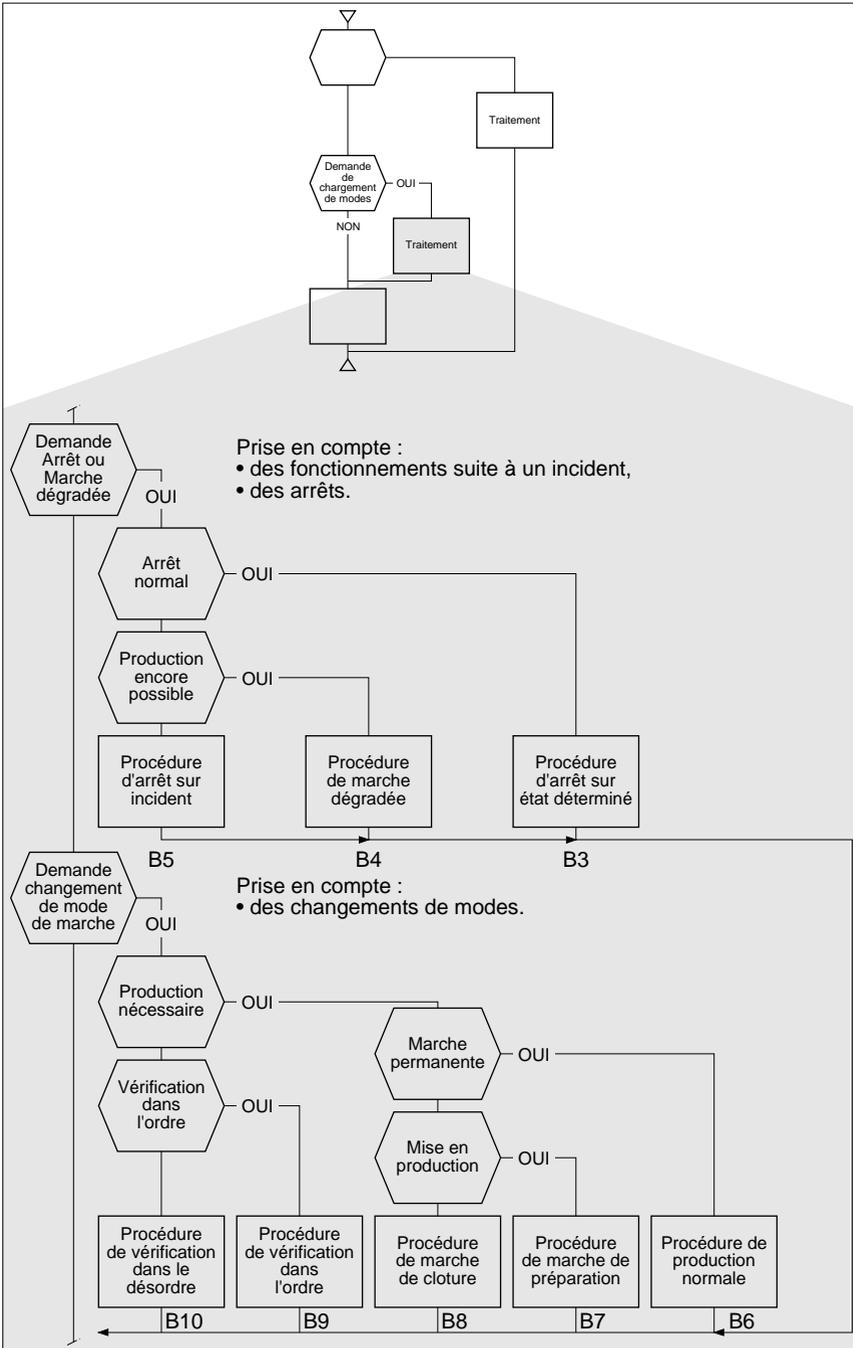
Les procédures sont spécifiques à chaque application, mais au niveau du traitement séquentiel, elles se résument comme suit :

- Remise à l'état initial,
- Remise à zéro du graphe principal,
- Prépositionnement d'étapes.

La démarche présentée pour la mise sous tension peut être reconduite pour les modes de marches ou d'arrêts (voir organigramme page suivante) :

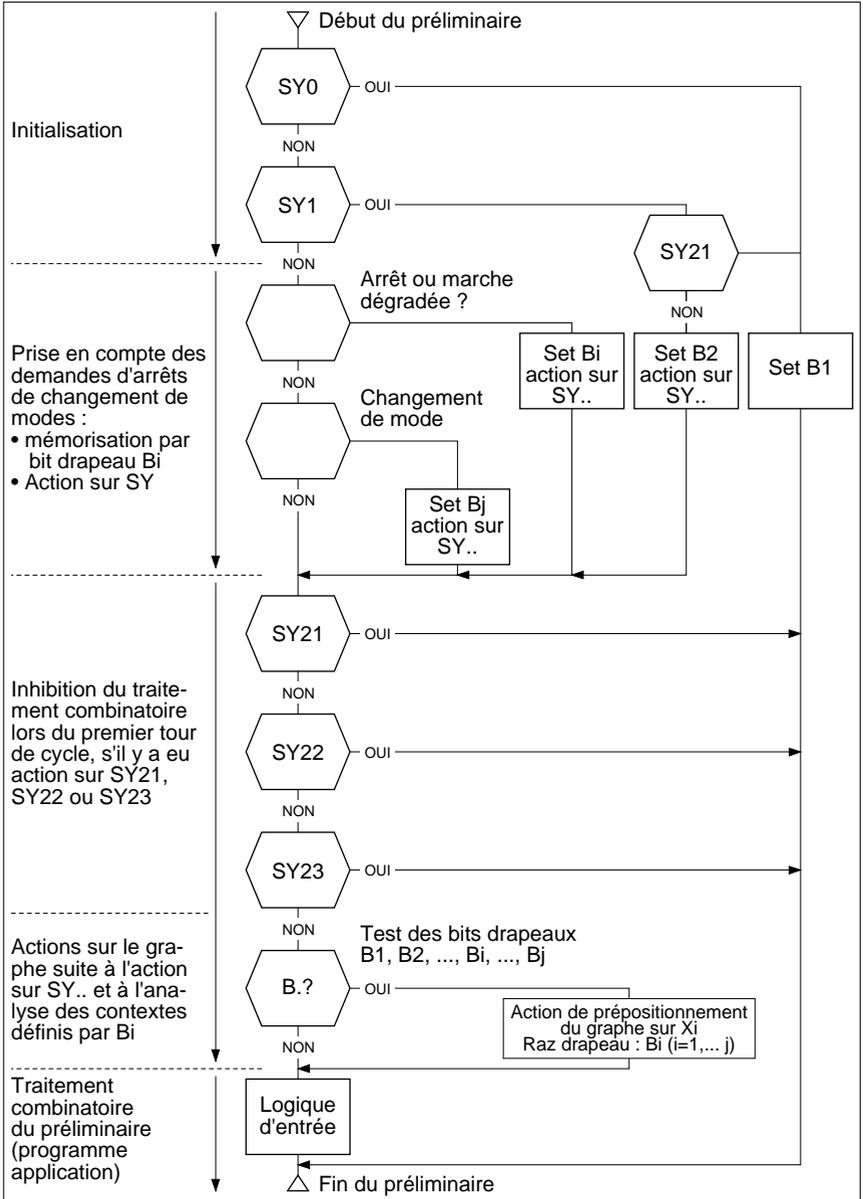
- Mémorisation par bits "drapeau" Bi,
- Action sur bits système,
- Action sur étapes.

Traitement suite à un changement de mode (suite)



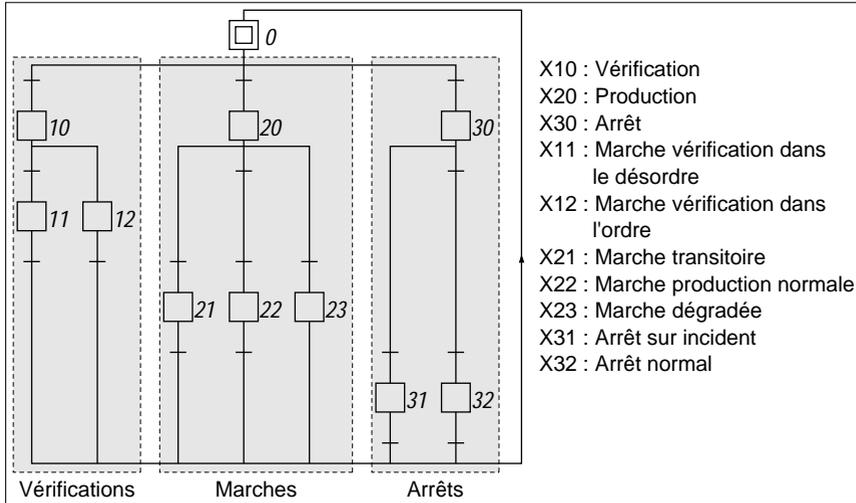
2.4-3 Organigramme du traitement préliminaire

La synthèse des traitements suite à une mise sous tension et/ou un changement de mode de marche conduit à l'organigramme ci-dessous, pour le traitement préliminaire.



2.4-4 GRAFCET de Conduite

L'analyse des modes de marches et d'arrêts, établie lors de l'élaboration du cahier des charges, peut être transcrite dans le programme application par un GRAFCET graphique appelé: GRAFCET de Conduite. Ce dernier vient s'ajouter au GRAFCET de l'application, appelé: GRAFCET de Production Normale.



Les marches de préparation et de clôture, dans le cas où l'ensemble des modes retenus sont utilisés, apparaissent regroupées sous la dénomination "Marche transitoire" (X21).

Le passage d'un mode à un autre entraîne la désactivation de l'étape en cours et l'activation de l'étape correspondant au nouveau mode (les réceptivités associées aux transitions sont définies par les bits "drapeaux" Bi).

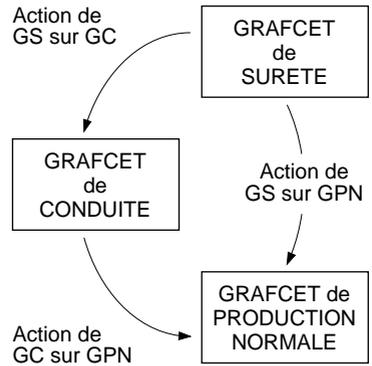
2.5 GRAFCET Complet

Répondant au cahier des charges, le GRAFCET Complet définit le comportement de l'application ou de l'automatisme.

Il comprend :

- Le GRAFCET de Sûreté (GS) : maître par rapport aux autres GRAFCET, il est consacré à la surveillance de la sûreté en signalant tout incident.
- Le GRAFCET de Conduite (GC) : intégrant les dispositions précisées par le GEMMA.
- Le GRAFCET de Production Normale (GPN): ou GRAFCET de l'application.

L'évolution de ces GRAFCET doit obéir à des règles de hiérarchie définies dans le préliminaire, lors de l'analyse et de l'élaboration des ordres de prépositionnement.



D

D



Sous chapitre	Page
3.1 Comportement des sorties en cas de dysfonctionnement automate	3/2
3.1-1 Définition	3/2
3.1-2 Comportement des sorties	3/2
3.2 Diagnostic des défauts	3/3
3.3 Traitement des défauts d'entrées/sorties	3/4
3.3-1 Défauts sur les modules TOR	3/4
3.3-2 Traitement	3/4
3.3-3 Cas d'un coupleur intelligent (pour TSX 47-20)	3/4
Ce chapitre se termine à la page	3/6

3.1 Comportement des sorties en cas de dysfonctionnement automate

3.1-1 Définition

Il y a disfonctionnement automate lorsque celui-ci n'est plus en mesure d'assurer l'exécution normale de son programme. L'origine de dysfonctionnement peut provenir du bac configuration (ou module) de base ou d'un bac (ou module) d'extension.

L'origine peut être :

- un défaut automate; défaut mémoire (MEM) ou défaut processeur (CPU),
- un débordement tâche : déclenchement du "chien de garde",
- un point d'arrêt effectif, demandé par le terminal en mode MISE AU POINT,
- une coupure secteur ou défaut d'alimentation sur la base ou une extension.

3.1-2 Comportement des sorties

En cas de disfonctionnement, le système réagit par sécurité sur les sorties, en fonction du positionnement du bit système "sécurité des sorties" : SY8, effectué par l'utilisateur.

Positionnement de SY8	Comportement des sorties sur disfonctionnement
SY8 = 1	Remise à zéro des sorties
SY8 = 0	Les sorties restent dans l'état

Sur disfonctionnement d'un module ou bac d'extension, seules les sorties du module ou bac concerné sont influencées. Le programme est toujours exécuté (RUN).

3.2 Diagnostic des défauts d'entrées/sorties

En cas de dysfonctionnement, des bits systèmes spécifiques permettent d'établir un diagnostic par programme ou par le terminal (en mode REGLAGE).

Ces bits sont :

- SY10 "défaut entrées/sorties", mis à l'état 0 sur défaut. Le voyant I/O de la base est également allumé.
- Bits "STATUS" : S, mis à 1 sur défaut du module d'entrées/sorties correspondants et **entraînant la mise à 0 de SY10**.

TSX 17-20

Il possède 8 bits de Status pour chaque module.

- Si,0=1 (ou Si) somme logique des défauts sur le module ou défaut de configuration,
- Si,1=1 défaut tension capteur (entrées/sorties T.O.R.), défaut d'alimentation (entrées analogiques), ou défaut d'alimentation extérieure (sorties analogiques),
- Si,2=1 défaut d'échange interne (entrées/sorties T.O.R.) ou problème de convertisseur (module analogique),
- Si,3=1 défaut de court-circuit sur les sorties statiques,
- Si,4=1 défaut d'alimentation interne des sorties relais (défaut 12V),
- Si,6=1 défaut d'accès module (entrées/sorties T.O.R. ou analogiques),
- Si,5 et Si,7 non utilisés.

Après avoir supprimé l'origine du défaut, si le voyant I/O reste allumé et si SY10 reste à 0, provoquer une initialisation par le terminal.

TSX 27

Uniquement pour sorties statiques

- Oy,S=1 court-circuit ou surcharge sur l'une des sorties du module de l'emplacement y.

TSX 47-J/47-10/47-20

Ils possèdent 1 bit status par module.

- Ixy,S=1 Défaut d'échange avec le processeur, module absent, configuration module erronée, bornier absent ou ouvert.
Cas du TSX DET 4 66 : contrôle supplémentaire de court-circuit.
Cas du TSX DET 32 12: contrôle supplémentaire de présence de l'alimentation des entrées.
- Oxy,S=1 Défaut d'échange avec le processeur, module absent, configuration module erronée, surintensité sur une sortie ou défaut fusible, bornier absent ou ouvert.
Cas du TSX DST 4 17 : contrôle supplémentaire d'alimentation externe.

Remarque : Ces bits sont remis à leur état initial par le système, dès la disparition du défaut.

3.3 Traitement des défauts d'entrées/sorties

3.3-1 Défauts sur les modules TOR

Un module d'entrées/sorties TOR est déclaré en défaut sur :

- défaut d'échange avec le processeur,
- configuration matérielle non conforme à la configuration logicielle ou absence module,
- bornier en cours de débrogage ou absent (sur TSX 47),
- surintensité
- module non alimenté (TSX DET 3212).

Dans l'un de ces cas, hors surintensité :

- le module est retiré de la table des échanges avec le processeur,
- pour un module d'entrées; mise à 0 des bits image dans la mémoire bits (TSX 27/47 uniquement),
- pour un module de sorties; forçage à 0 des sorties du module (si SY8=1), les bits image de la mémoire bits restant dans l'état,
- les bits status sont mis à 1, SY10 à 0 et le voyant I/O de la base allumé.

Sur surintensité, seul le dernier point cité ci-dessus est exécuté avec mise à 0 de la voie en défaut : une voie en défaut est sans effet sur les autres voies du même module.

3.3-2 Traitement

Lors de l'apparition d'un défaut sur un des modules d'entrées/-sorties, une procédure de traitement peut être effectuée afin de localiser rapidement le ou les modules responsable(s) de ce défaut. Ceci afin d'agir par programme pour :

- provoquer une marche dégradée, lancer un cycle de dégagement, etc,
- renseigner l'exploitant du processus sur le module en cause.

3.3-3 Cas d'un coupleur intelligent (pour TSX 47-20)

La procédure d'analyse est identique à celle d'un module d'entrées/sorties TOR. Si l'on désire analyser l'ensemble des défauts coupleurs il est nécessaire de tester en plus les bits Ixy,S, les bits IWxy,0,7 de chaque coupleur.

En effet, plusieurs catégories de défauts sont à distinguer (voir manuels de mise en oeuvre de chaque coupleur).

Exemples

Défaut "bloquant"

Ce défaut correspond à une configuration non conforme, à un défaut de l'unité de traitement du coupleur ou à un défaut d'échange.

Le coupleur se met en arrêt exécution . Le processeur le retire de la table des échanges, les bits et les mots correspondants sont mis à 0.

Défaut coupleur

Ce défaut correspond à un défaut des circuits d'acquisition ou d'adaptation ou bornier débroché.

Le processeur automate met le coupleur en STOP. Les échanges sont alors figés, les bits et les mots correspondants conservent la valeur qu'ils avaient au moment de la mise en STOP.

Défaut application

Ce défaut correspond au contrôle assuré par le coupleur vis-à-vis des capteurs ou préactionneurs qui lui sont directement connectés. Le coupleur reste en RUN.

D



Sous chapitre	Page
4.1 Protection des programmes des cartouches EPROM et EEPROM	4/2
4.2 Configuration des entrée/sortie spécifiques	4/3
4.2-1 Configuration de l'entrée de commande RUN/STOP	4/3
4.2-2 Configuration de la sortie SECURITE	4/3
4.3 Modules de visualisation	4/4
4.3-1 Principe	4/4
4.3-2 Module de visualisation standard	4/4
4.3-3 Module de visualisation numérique TSX 17 ACC2	4/5
4.3-4 Exemple de programmation	4/5
Ce chapitre se termine à la page	4/6

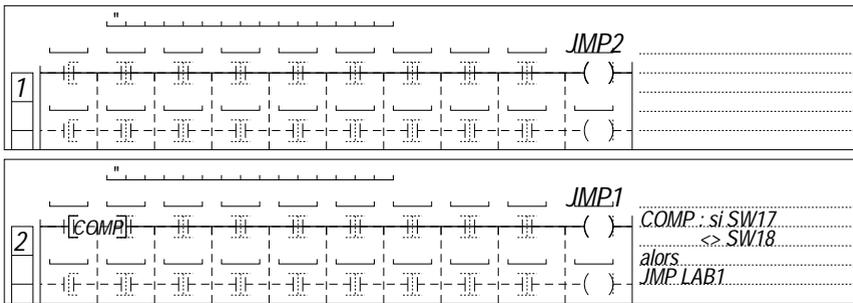
4.1 Protection des programmes des cartouches EPROM et EEPROM

Les programmes contenus dans les cartouches mémoire EPROM et EEPROM peuvent être protégés, durant leur sauvegarde, contre les relectures et les modifications. Pour cela, il suffit, lors du transfert RAM vers cartouche (à partir du terminal) de sélectionner l'option "Protection" (PROTECT).

Un programme protégé est **uniquement** exécutable à partir de la RAM. Il n'est plus accessible à la lecture ni à la modification. Ce programme reste cependant accessible aux multiples duplications sur d'autres RAM de TSX 17-20.

Le verrouillage contre les duplications consiste à rendre un programme exécutable, uniquement si la cartouche qui le contient est enfichée en face avant de l'automate, lorsque ce dernier est en RUN.

L'opération de verrouillage s'effectue par programme. Elle consiste à écrire au début du préliminaire le réseau de contact ci-dessous.



Si le contenu de SW17 est différent de celui de SW18, le processeur effectuera des sauts du label 2 vers le label 1, jusqu'à déclenchement du "chien de garde" (temps d'exécution programme supérieure à 150ms).

Un programme utilisateur protégé et verrouillé devient inviolable. Ainsi il est possible de disposer d'un "progiciel" par application.

4.2 Configuration des entrée/sortie spécifiques

L'entrée I0,0 et la sortie O0,0 du TSX 17-20 peuvent être chacune affectées à un rôle spécifique.

4.2-1 Configuration de l'entrée de commande RUN/STOP

I0,0 : entrée de commande RUN/STOP. Elle permet d'accéder directement au moyen d'un interrupteur raccordé, à la mise en RUN ou STOP de l'automate.

La configuration de cette entrée se fait à partir du terminal, dans le mode PROGRAMMATION :

I0=I entrée normale (état par défaut). Lorsque I0,0 n'est pas configurée, son fonctionnement est identique à tout autre entrée T.O.R. du module de base.

I0=R/S entrée RUN/STOP. Lorsque cette entrée est configurée, son fonctionnement devient le suivant :

- sur front montant: met en RUN l'automate,
- à l'état 0 : met en STOP l'automate.

La mise en STOP à partir de cette entrée est prioritaire sur celle à partir du terminal.

4.2-2 Configuration de la sortie SECURITE

O0,0 : sortie SECURITE. Elle permet l'asservissement de l'alimentation des sorties de l'automate. Cette sortie est à câbler dans les circuits de sécurité externes (voir manuel de mise en oeuvre).

La configuration de cette sortie se fait à partir du terminal, dans le mode PROGRAMMATION :

O0=O sortie normale (état par défaut). Lorsque O0,0 n'est pas configurée, son fonctionnement est identique à tout autre sortie T.O.R. du module de base.

O0=SECU sortie SECU. Lorsque cette SORTIE est configurée, son état est défini par le système comme suit :

- à l'état 1 : lorsque l'automate est en RUN et sans défaut,
- à l'état 0 : lorsque l'automate est en STOP ou en RUN avec défaut.

Nota : I0,0 et O0,0 sont configurées par défaut en entrée/sortie normales.

Elles peuvent être configurées indépendamment, à partir du terminal.

Une fois configurées, ces entrée/sortie ne doivent plus être utilisées dans le programme.

4.3 Modules de visualisation

4.3-1 Principe

L'automate de base TSX 17-20 dispose en face avant d'un module permettant selon l'état de SY14 :

- la visualisation des entrées/sorties si SY14=0,
- la visualisation de l'état des bits du mot système SW16 si SY14=1.

En lieu et place de ce module, peut être enfilé un autre module de visualisation Numérique (référéncé TSX 17 ACC2), qui complète la visualisation des entrées/sorties par un affichage sur 4 digits en BCD du contenu de SW16(si SY14=1).

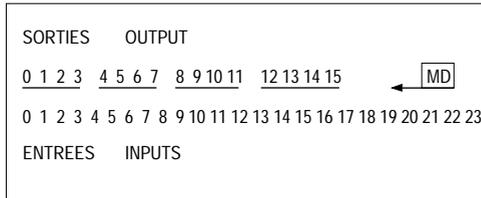
Le contenu de SW16 affiché, peut être une valeur de mot interne, de compteur, de temporisateur, un numéro d'étape GRAFCET, etc.

4.3-2 Module de visualisation standard

Eléments visualisés	Etat SY14	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15															
Sorties	0	00,00...				00,08...				00,08...				00,12...			
Mot SW16	1	SW16,F				SW16,3 ...			

L'état du bit SY14 est visualisé en face avant par le voyant MD (Memory Display).

- SY14=0 voyant MD éteint (état par défaut)
- SY14=1 voyant MD allumé

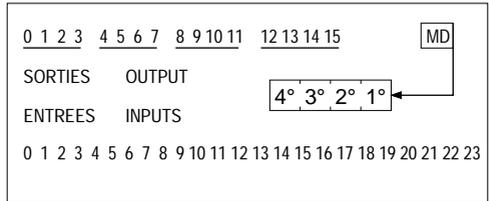


4.3-3 Module de visualisation Numérique TSX 17 ACC2

Eléments visualisés	Etat SY14	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Sorties	0	00,00...				00,08...				00,08...				00,12...			
Mot SW16	1	rang F (MSB)															
		4° quartet				3° quartet				2° quartet				1° quartet			

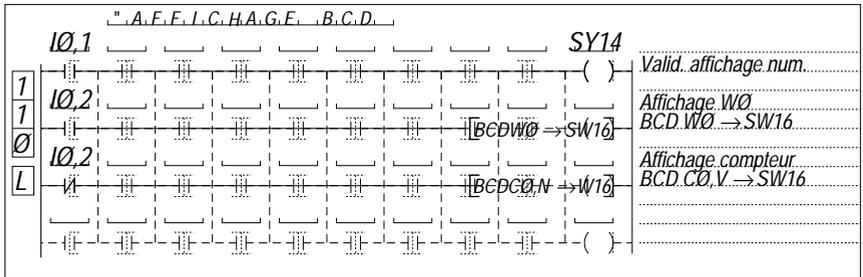
L'état du bit SY14 est visualisé en face avant par le voyant MD (Memory Display).

- SY14=0 voyant MD éteint (état par défaut)
- SY14=1 voyant MD allumé



4.3-4 Exemple de programmation

Il s'agit de visualiser alternativement, selon l'état de l'entrée I0,2, la valeur du mot W0 et la valeur courante C0,V du compteur C0.



D



Chapitre	Page
1 Généralités	1/1
1.1 Généralités sur la "liaison terminal"	1/2
1.2 Généralités sur les transmissions de données	1/3
2 Liaison terminal	2/1
2.1 Liaison terminal du TSX 17-20	2/2
2.2 Liaison terminal des TSX 27/47	2/5
2.3 Bloc texte du type terminal (rappel)	2/10



Sous chapitres	Page
1.1 Généralités sur la "liaison terminal"	1/1
1.1-1 Présentation	1/2
1.1-2 Caractéristiques de la liaison	1/2
1.1-3 Configuration de la liaison	1/2
1.2 Généralités sur les transmissions de données	1/3
1.2-1 Définition	1/3
1.2-2 Types de transmission	1/3
1.2-3 Transmission série asynchrone	1/4
1.2-4 Types de liaison série	1/5
Ce chapitre se termine à la page	1/6

1.1 Généralités sur la "liaison terminal"

1.1-1 Présentation

La prise terminal des automates de la série TSX7 permet :

- le raccordement des terminaux de la série TSX7 et des terminaux universels, pour la programmation et le réglage des TSX 17-20/27/47. Les échanges s'effectuent dans ce cas de manière "transparente" à l'utilisateur, à la vitesse optimale de 9600 bauds.
- le raccordement de terminaux d'exploitation type XBT. La gestion des échanges est rendue "transparente" à l'utilisateur grâce au protocole "Réglage", spécifique à la série TSX7, qui assure le dialogue entre l'automate et le terminal.
- le raccordement des terminaux universels ou des périphériques disposant d'une liaison série. La "liaison terminal" assure dans ce cas les échanges de chaînes de caractères ASCII entre la mémoire de l'automate et le périphérique. Les messages sont véhiculés vers la prise terminal par les blocs textes du type TER, et les échanges se font à une vitesse configurable (300 à 9600 bauds).

Le protocole de gestion des échanges est dans ce cas à la charge de l'utilisateur. Il doit être implanté dans le périphérique.

1.1-2 Caractéristiques de la liaison

La "Liaison Terminal" des automates de la série TSX7 est une liaison "point à point", servant de support à une transmission série asynchrone du type :

- Boucle de courant : cas des TSX 27/47,
- RS 485 : cas du TSX 17-20.

Le raccordement s'effectue sur l'unité centrale de l'automate à l'aide d'un connecteur "SUB D" (type CANNON), mâle :

- 9 broches, pour les TSX 27/47,
- 15 broches, pour le TSX 17-20.

1.1-3 Configuration de la liaison

En cas de raccordement d'un terminal TSX série 7, le dialogue s'effectue à la vitesse optimale de 9600 bauds.

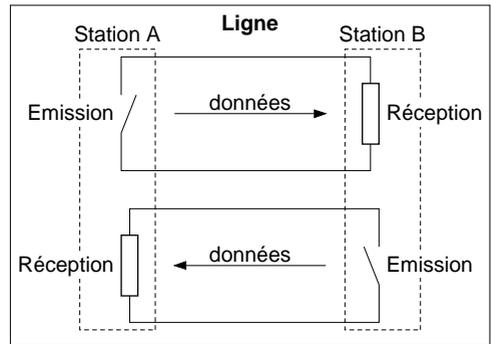
En cas de raccordement d'un périphérique, la vitesse est configurable (300 à 9600 bauds).

1.2 Généralités sur les transmissions de données

1.2-1 Définition

Une transmission de données est un échange d'informations entre deux équipements (stations A et B).

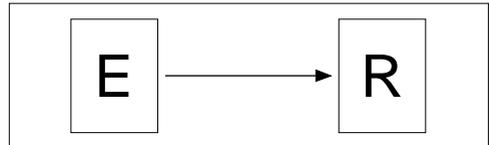
Cet échange, véhiculé sur la ligne de transmission, doit être défini dans un langage codé, compréhensible par l'interface émetteur et l'interface récepteur.



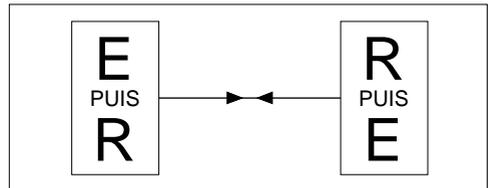
1.2-2 Types de transmission

Une transmission de données peut être :

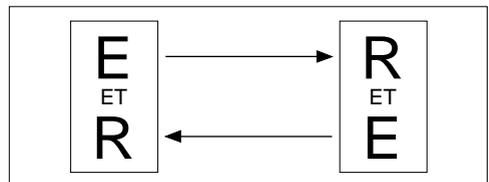
- du type **SIMPLEX** (ou unidirectionnelle). La station émettrice est toujours émettrice et la station réceptrice est toujours réceptrice. Exemple: transmission automate vers imprimante.



- du type **HALF-DUPLEX** (ou bidirectionnelle à l'alternat). La station émettrice peut être réceptrice, mais pas simultanément. C'est le cas de la prise terminal. Exemple : transmission entre automate et XBT.



- du type **FULL-DUPLEX** (ou bidirectionnelle simultanée). Les deux stations sont simultanément émettrices et réceptrices. Exemple: transmission entre mémoire automate et coupleur SCM..



1.2-3 Transmission série asynchrone

La transmission série asynchrone est utilisée dans le domaine industriel pour les échanges: automate-automate, ou terminal-automate.

La transmission série est une émission de caractères (chaînes de bits), les uns derrière les autres, sur une même voie, d'un émetteur vers un récepteur. Cette transmission est dite asynchrone car la synchronisation entre l'émetteur et le récepteur est au niveau du caractère : 1 bit "START" et 1 ou 2 bits "STOP" sont transmis entre l'envoi de 2 caractères. Ils signalent le début et la fin du code du caractère transmis.

Les caractéristiques de la transmission série sont (voir schéma ci-dessous) :

Vitesse de transmission

Variable selon les périphériques (300 à 9600 bauds),

Bit de START

Chaque caractère transmis est précédé d'un bit de start (synchronisation : bit à 0),

Format du caractère

7 ou 8 bits de données :

- Pour les échanges ASCII : format 7 bits (le huitième bit peut servir pour le contrôle de parité).
- Pour les échanges en binaire pur : format 8 bits.

Contrôle de parité

Afin de détecter les éventuelles erreurs de transmission, le bit de rang 8 du code ASCII du caractère peut être réservé au contrôle de parité :

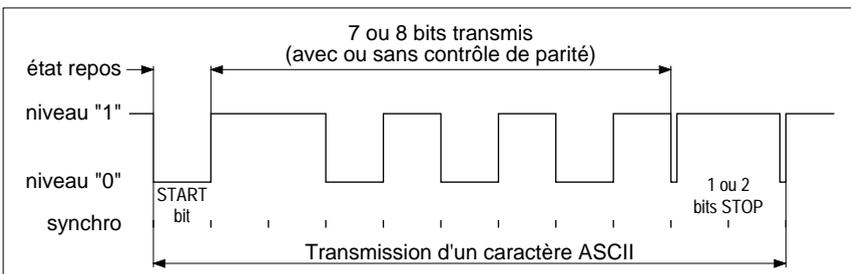
- contrôle de type PAIR : le bit de rang 8 est positionné à 0 ou à 1 afin que le nombre total de 1 dans le code du caractère soit pair.
- contrôle de type IMPAIR : le bit de rang 8 est positionné à 0 ou à 1 afin que le nombre total de 1 dans le code du caractère soit impair.

Ce bit est calculé et émis par l'émetteur à la suite du code du caractère. Il est de nouveau calculé par le récepteur et comparé au bit reçu.

Bits de STOP

Chaque caractère transmis peut être suivi d'un ou deux bits de stop (synchronisation : bits à 1). Ainsi, un bit de start peut de nouveau être identifié.

En l'absence de transmission (à l'état repos), la ligne est en général à l'état logique 1.



1.2-4 Types de liaison série

Les transmissions de données, via la liaison série asynchrone, sont normalisées. Les normes correspondantes, concernent les caractéristiques :

- électriques (tension, courant),
- fonctionnelles (circuits d'émission et de réception),
- de connexion (type de connecteur et brochage).

Ces normes sont définies par :

- le CCITT : Comité Consultatif International Télégraphique et Téléphonique,
- l'EIA : Electronic Industries Association,
- l'ISO : International Standard Organization.

Les principales normes sont :

- **RS 232C**, interface de tension, signal +12 / -12V:
 - 1 logique = -3 à -15V,
 - 0 logique = +3 à +15V.

Cet interface est utilisé pour des débits binaires inférieurs à 20 Kbits/s et entre des stations distantes de moins de 15m.

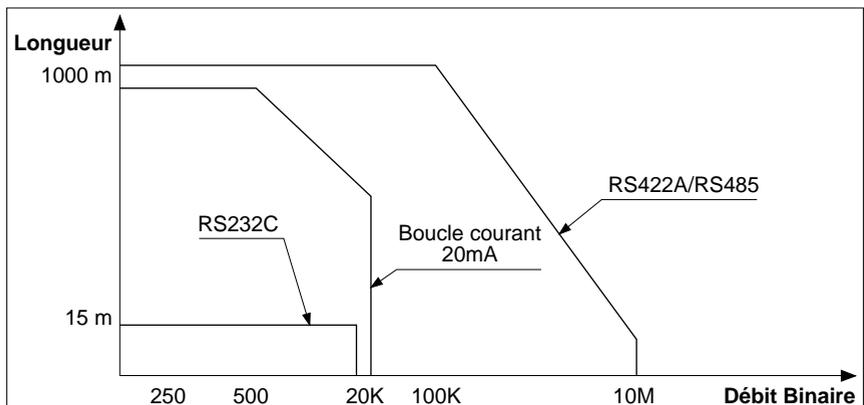
- **Boucle de courant**, interface de courant :
 - simple courant : 0 / 20 mA,
 - double courant : +20 / -20 mA.

L'interface est dite ACTIVE si elle fournit l'énergie pour transporter l'information. Elle est dite PASSIVE dans le cas contraire.

- **RS 422/485**, interface de tension, signal < 6V :
 - 1 logique = tension différentielle < -200 mV,
 - 0 logique = tension différentielle > 200 mV.

Ses caractéristiques électriques (mode différentiel, courant élevé) offrent de très bonnes qualités de transmission et d'immunités, supérieures à celles de la RS 232C.

Le diagramme ci-dessous illustre les domaines d'utilisation de ces trois normes, en fonction des distances entre stations et du débit binaire.



Nota : pour les liaisons asynchrones, le débit binaire maximum ne dépasse pas normalement 19200 bits/s.



Sous chapitre	Page
2.1 Liaison terminal du TSX 17-20	2/2
2.1-1 Définition	2/2
2.1-2 Utilisation	2/2
2.1-3 Caractéristique de la liaison	2/3
2.1-4 Raccordement de la prise terminal du TSX 17-20	2/4
2.2 Liaison terminal des TSX 27/47	2/5
2.2-1 Définition	2/5
2.2-2 Utilisation	2/5
2.2-3 Caractéristiques de la liaison	2/6
2.2-4 Raccordement de la prise terminal du TSX 27	2/7
2.2-5 Raccordement de la prise terminal du TSX 47	2/9
2.3 Bloc texte du type terminal (rappel)	2/11
2.3-1 Définition du bloc texte TER	2/11
2.3-2 Caractéristiques du bloc texte TER	2/11
Ce chapitre se termine à la page	2/12

2.1 Liaison terminal du TSX 17-20

2.1-1 Définition

La "liaison terminal" du TSX 17-20 est du type RS 485, non isolée.

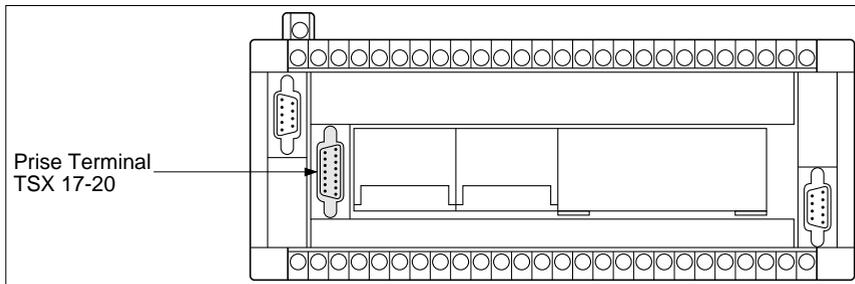
Elle assure :

- Les échanges avec les terminaux de programmation et de réglage TSX T317 (via adaptateur TSX 17 ACC7) FTX 417/507 et compatible PC (via adaptateur TSX 17 ACC PC = ACC8 + ACC11).
- Les échanges avec les périphériques type terminaux universels et écran/clavier⁽¹⁾. Les blocs textes du type TER sont utilisés pour l'émission et/ou la réception des chaînes de caractères.
Le protocole de gestion des échanges est à la charge de l'utilisateur; il doit être défini dans le périphérique.
- Les échanges avec les terminaux type XBT. Le protocole réglage, spécifique à la série TSX7, assure les échanges.
- La communication via le Bus multipoint UNI-TELWAY. Un interface (TSX 17 ACC5) assure la liaison entre le TSX 17-20 (par la prise terminal) et le bus.
Le TSX 17-20 se comporte comme une station ESCLAVE et répond aux requêtes de la station MAITRE (TSX 17-20 muni du coupleur TSX SCG 116, TSX 47/67/87 muni d'une carte coupleur TSX SCM 21.6).

2.1-2 Utilisation

La norme RS 485 est utilisée pour toutes les transmissions jusqu'à 1000 m, et pour un débit binaire maximum de 100 Kbits/s.

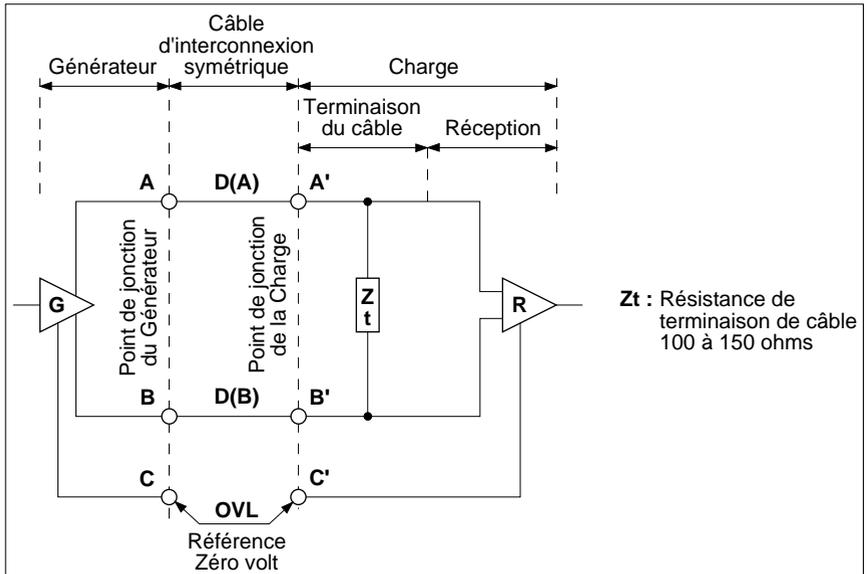
Ses caractéristiques électriques (mode différentiel, courant élevé) offrent de très bonnes qualités de transmission et d'immunité, supérieures à celles de la RS 232C.



⁽¹⁾ La liaison doit être isolée si la longueur de la ligne est supérieure à 5 m.

2.1-3 Caractéristiques de la liaison

Le schéma ci-dessous illustre le principe de fonctionnement de la transmission de données type RS 485.



La transmission des données s'effectue en mode différentiel sur une paire torsadée. Ces signaux de données, appelés D(A) et D(B) doivent avoir des caractéristiques électriques conformes à la norme EIA RS 485. Ces signaux correspondent respectivement aux points A/A' et B/B' de la norme. Le bon fonctionnement des émetteurs-récepteurs nécessite d'autre part l'interconnexion des "communs" de chaque station raccordée (points C/C').

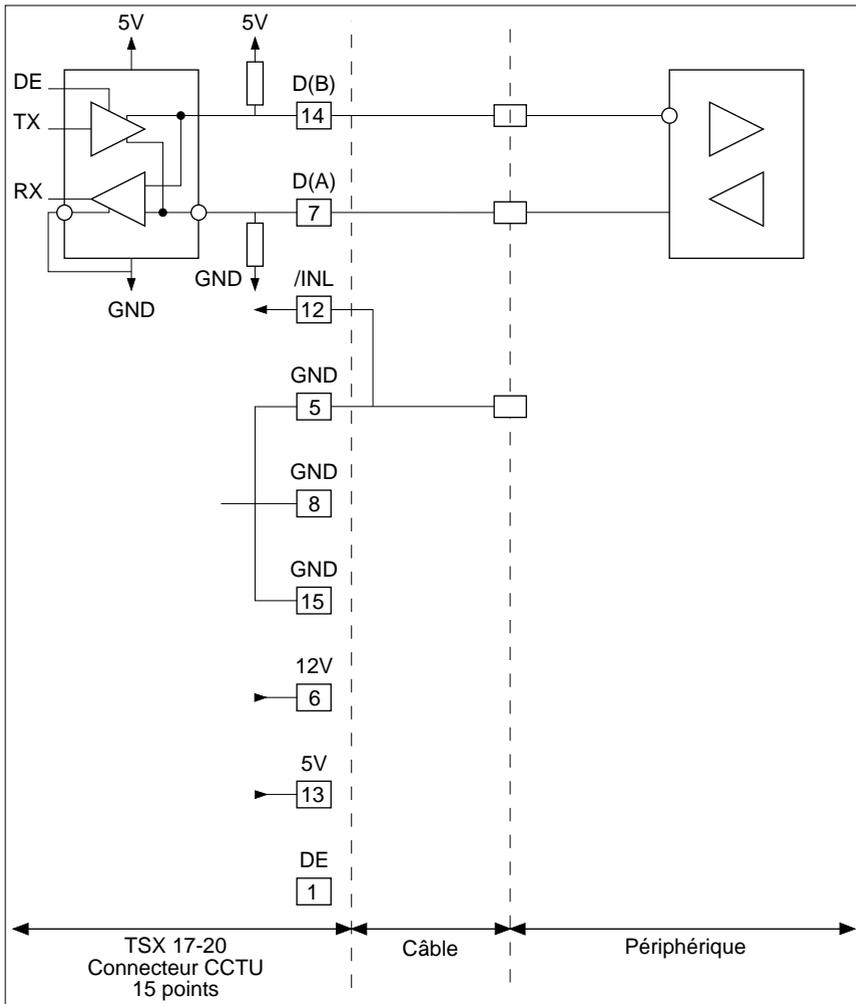
Cette référence de potentiel, appelée OVL, sera véhiculée par une deuxième paire torsadée. L'ensemble des deux paires torsadées sera impérativement blindé afin d'être protégé contre les perturbations électromagnétiques (signal appelé BLIND).

Tableau de fonctionnement, défini par convention

Potentiel D(A) - D(B)		Etat logique de la ligne
D(A) - D(B) < -200 mV	→	état binaire = 1 (ou MARK)
D(A) - D(B) > +200 mV	→	état binaire = 0 (ou SPACE)
-200 mV < D(A) - D(B) < +200mV	→	état indéterminé
Transmission : HALF-DUPLEX		
Impédance de charge minimale : 100 ohms		
longueur de câble maximale : 1000 m		

2.1-4 Raccordement de la prise terminal du TSX 17-20

Raccordement par connecteur SUB D femelle 15 points (voir schéma de raccordement et tableau de brochage ci-dessous).



Broche	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Référence	-DE*				GND	12V	D(A)	GND				-INL	VCC	D(B)	GND

* pour TSX ACC7 et TSX ACC5

2.2 Liaison terminal des TSX 27/47

2.2-1 Définition

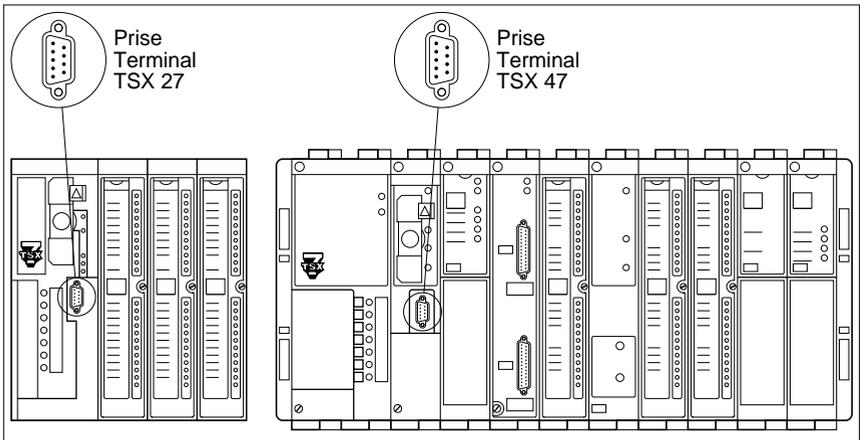
La "liaison terminal" des TSX 27/47 est du type Boucle de courant 20 mA.
Elle assure :

- Les échanges avec les terminaux de programmation et de réglage TSX T407, FTX 417/507 et compatible PC (via adaptateur TSX TAC 03).
- Les échanges avec les périphériques type terminaux universels et écran/clavier. Les blocs textes du type TER sont utilisés pour l'émission et/ou la réception des chaînes de caractères.
Le protocole de gestion des échanges est à la charge de l'utilisateur; il doit être défini dans le périphérique.
- Les échanges avec les terminaux type XBT. Le protocole réglage, spécifique à la série TSX7, assure les échanges.

2.2-2 Utilisation

La transmission de données du type Boucle de courant 0 / 20 mA est la plus courante. Elle n'est pas normalisée, mais est devenue une norme de fait.

Elle est utilisée pour des distances entre stations inférieures à 1000 m et pour des débits binaires de l'ordre 20 Kbits/s.



2.2-3 Caractéristiques de la liaison

La connexion est du type Boucle de courant 20 mA, courant passant au repos :

- non isolée pour le TSX 27,
- isolée pour le TSX 47.

Le connecteur fournit l'alimentation des terminaux TSX T107 (12V, 140mA), TSX T407 (12V, 900mA).

Le schéma ci-dessous illustre le principe de fonctionnement de la transmission de données type Boucle de courant.

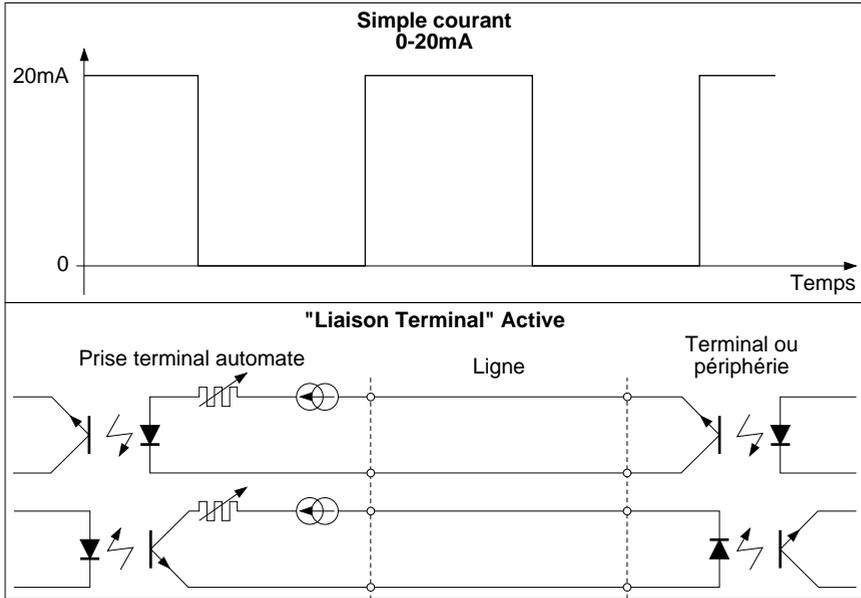
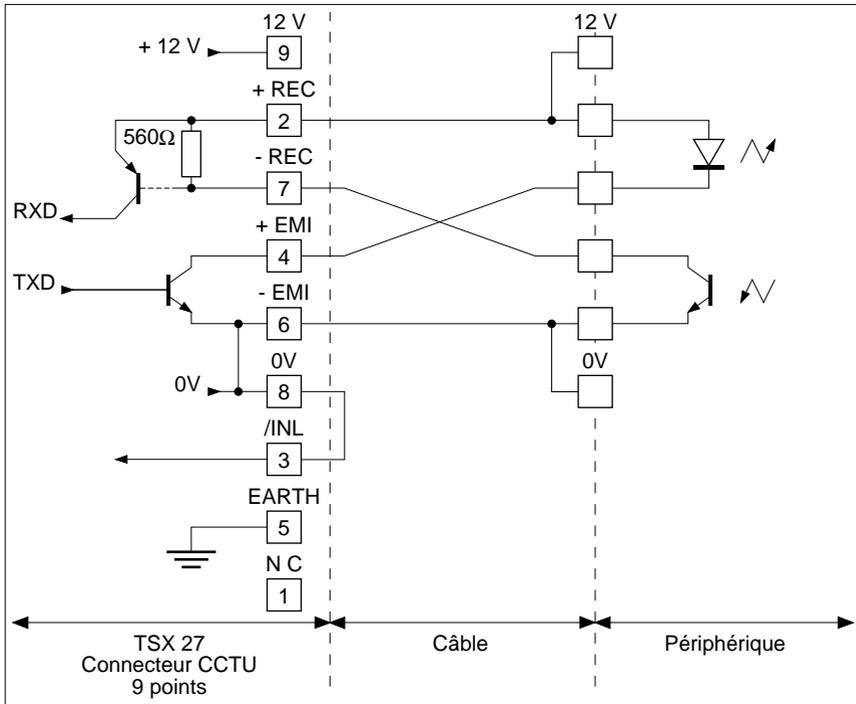


Tableau de fonctionnement, défini par convention

<p>Simple courant 0/20 mA</p> <p>20 mA → état binaire = 0</p> <p>0 mA → état binaire = 1</p>	<p>Transmission : HALF-DUPLEX</p> <p>Débit binaire maximum : 20 Kbits/s</p> <p>longueur de câble maximale : 1000 m</p>
---	--

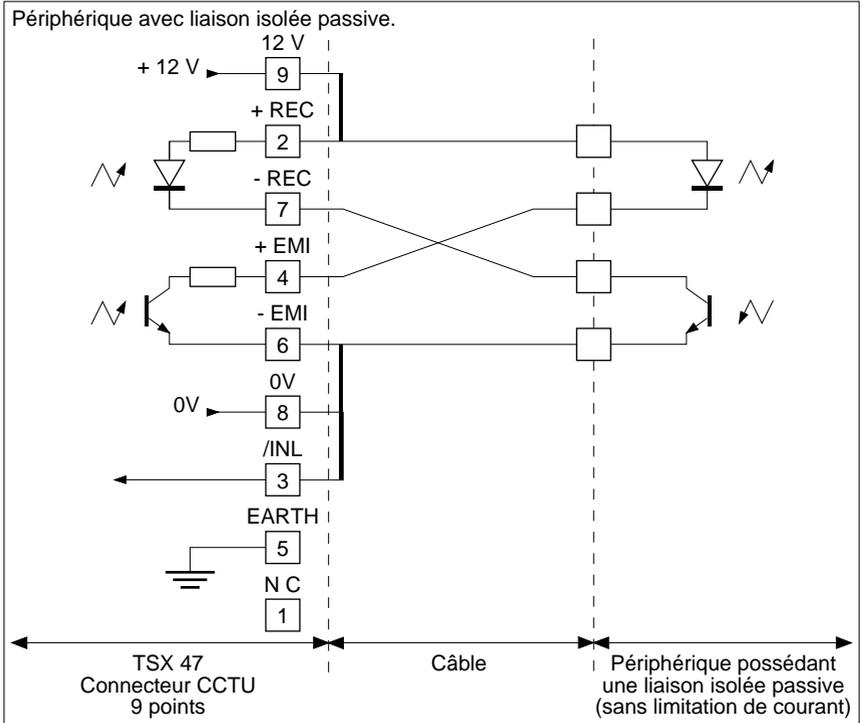
Périphérique avec liaison active (avec limitation de courant)



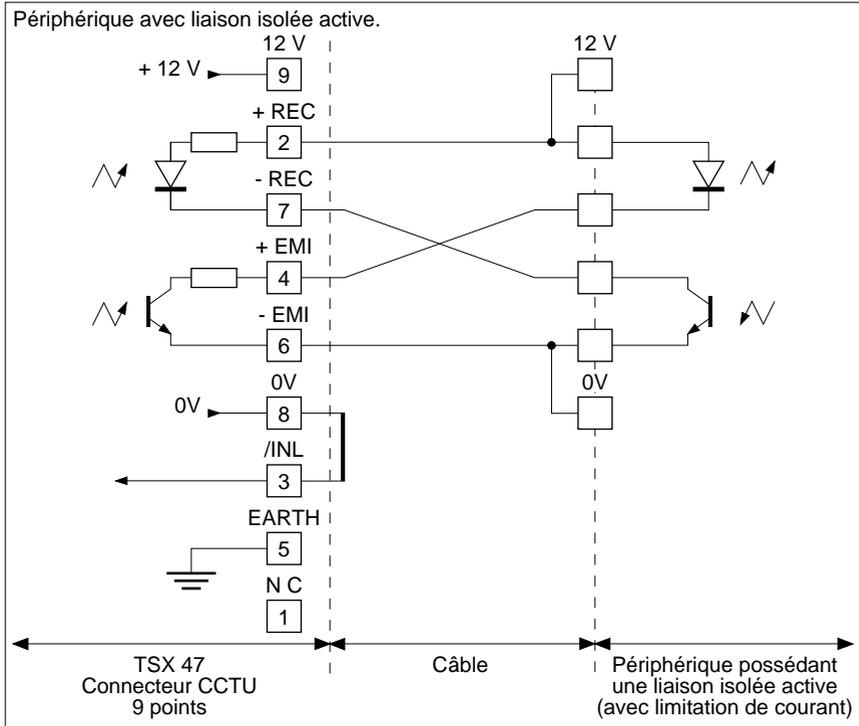
E

2.2-5 Raccordement de la prise terminal du TSX 47

Raccordement par connecteur SUB D femelle 9 points (voir schéma de raccordement ci-dessous).



Raccordement de la prise terminal du TSX 47 (suite)

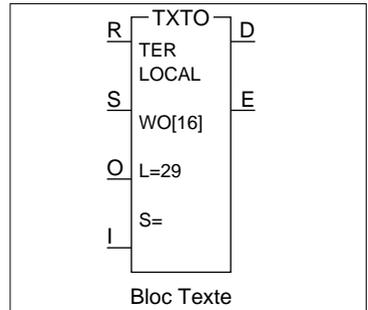


E

2.3 Bloc texte du type terminal (rappel)

2.3-1 Définition du bloc texte TER

Le bloc texte du type terminal (TER) est un bloc fonction qui véhicule des messages entre la mémoire RAM de l'automate et la prise terminal.



2.3-2 Caractéristiques du bloc texte TER

Les caractéristiques générales du bloc texte TER sont détaillées dans : "Langage à contacts", chapitre 6. Le tableau ci-dessous en donne un extrait.

<p>entrée R "Reset" : inhibition de l'échange</p> <p>entrée S "Start" : départ, sur front montant provoque le lancement de l'échange</p> <p>entrée O "Out" : émission</p> <p>entrée I "In" : réception</p> <p>sortie D "Done" : échange terminé bit associé: TXTi,D</p> <p>sortie E "Error" : erreur d'échange bit associé: TXTi,E</p>	
<p>TXTi,L : longueur table d'émission en nombre de caractères</p> <p>TXTi,S : nombre de caractères reçus ou code d'erreur</p> <p>↔ TER : bloc texte type terminal</p> <p>LOCAL : échange en local</p> <p>Wi[j] : Wi [j] : Wi adresse de début table de réception j : nombre de caractères à recevoir</p>	

Nota : Les fenêtres dans le tableau ci-dessus, illustrent la représentation des blocs fonction textes sur l'écran du terminal de programmation TSX T407.



Chapitre		Page
1	<i>Introduction</i>	1/1
	1.1 Préambule	1/2
	1.2 Aide à la programmation : actions sur front	1/3
2	<i>Exemples d'applications avec TSX 17-20</i>	2/1
	2.1 Traitement d'un positionnement par la tâche rapide	2/2
	2.2 Compteur/Temporisateur rapide en multiprésélection	2/6
3	<i>Exemples d'applications avec TSX 17-20/27/47</i>	3/1
	3.1 Description de la machine	3/2
	3.2 Grafset du cahier des charges (première partie)	3/3
	3.3 Modes de marche et d'arrêt (deuxième partie)	3/14
	3.4 Comptage des pièces (troisième partie)	3/20
	3.5 Communication par bloc texte (quatrième partie)	3/22
4	<i>Annexes</i>	4/1
	4.1 Temps d'exécution mémoire	4/2
	4.2 Définitions GEMMA	4/4
	4.3 Code ASCII	4/5



Sous chapitres	page
1.1 Préambule	1/2
1.2 Aide à la programmation : actions sur front	1/2
Ce chapitre se termine à la page	1/4

1.1 Préambule

Les chapitres 2 et 3 traitent quelques exemples d'applications étudiés en langage PL7-2.

Chaque exemple a été simplifié afin de paraître didactique et est décomposé en parties successives, analysées et traitées l'une après l'autre :

- Cycle de fonctionnement, défini par le cahier des charges.
- Modes de Marches et d'Arrêts.
- Fonctions complémentaires (affichage, communication, etc).

Tous les programmes sont donnés sur des formulaires de programmation ("schéma à contacts" ou "GRAFSET"). Ils peuvent être saisis sur n'importe quel terminal de programmation en langage PL7-2.

La machine automatique de perçage (chapitre 3) est un exemple extrait de l'ouvrage : "Le GRAFCET, sa pratique et ses applications", des éditions EDUCALIVRE. Cet exemple est traité avec une affectation d'entrées/sorties pour TSX 17-20. Moyennant un adressage adéquat (voir intercalaire A, chapitre 2, sous-chapitre 2.1), Il pourra être testé sur un automate TSX 27 ou TSX 47.

1.2 Aide à la programmation : Actions sur front

Afin de tenir compte de contraintes d'application ou de modes de marches, certaines actions ne doivent être exécutées que sur un cycle programme (dès que et non pas tant que la condition est vraie).

Ces actions sont alors dites exécutées sur fronts :

- montant, lorsque le bit associé à la condition passe de l'état 0 à 1,
- descendant, lorsque le bit associé à la condition passe de l'état 1 à 0.

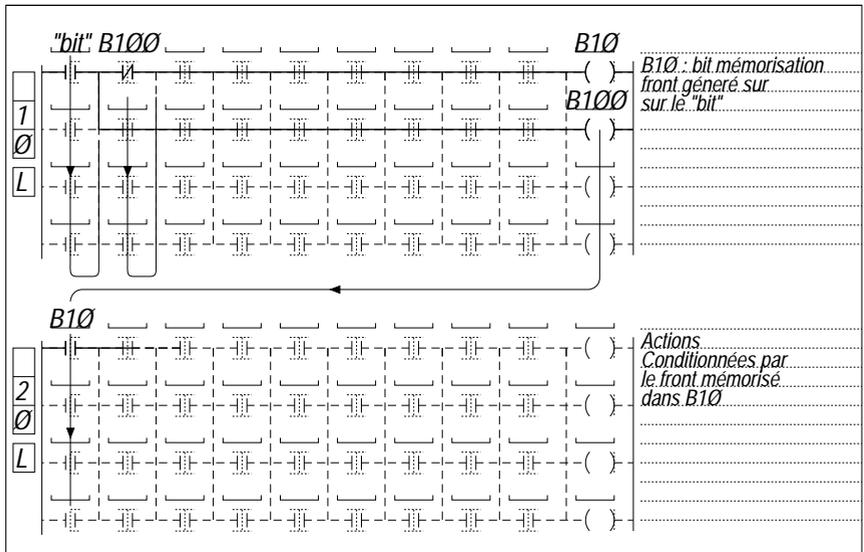
Principe de programmation d'une action sur fronts

Dès que la condition est vraie (B100 étant encore à 0), le bit B10 est positionné à 1. Le bit B100 est à son tour positionné à 1.

Au label 20, B10 étant à 1, les actions associées sont alors effectuées.

Au tour de cycle suivant, B100 étant à 1, B10 repasse à 0 et les actions associées ne seront pas exécutées.

Ce traitement est renouvelé chaque fois que le bit associée à la condition passe de l'état 0 à l'état 1.



Remarque : Ce traitement prend obligatoirement deux réseaux de contacts, car le bit mémorisation de front n'est reconnu à 1 qu'à partir du réseau qui suit celui dans lequel est géré le bit "condition".



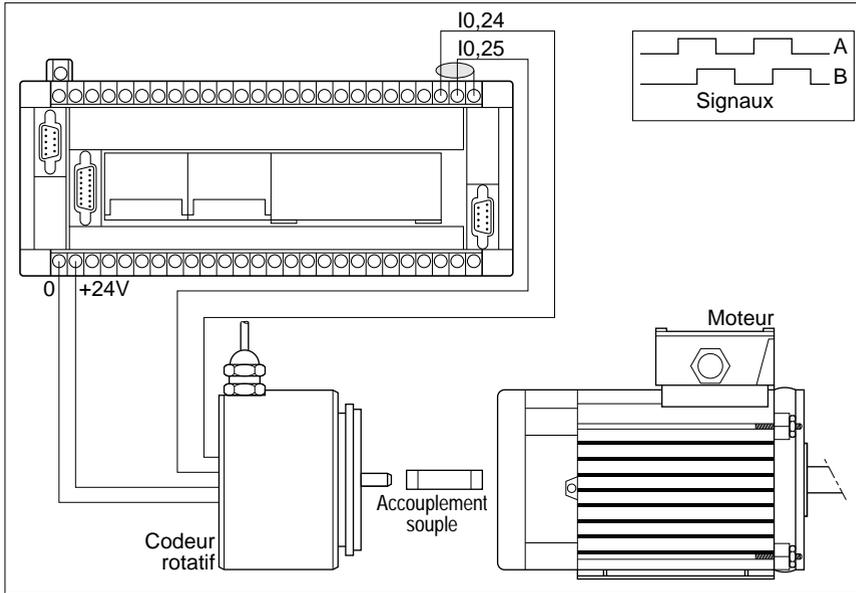
Sous chapitre	Page
2.1 Traitement d'un positionnement par la tâche rapide	2/2
2.2 Compteur/temporisateur rapide en multiprésélection	2/6
2.2-1 Description de l'application	2/6
2.2-2 Données de programmation	2/9
2.2-3 Programme	2/10
 Ce chapitre se termine à la page	 2/16

2.1 Traitement d'un positionnement par la tâche rapide

Les entrées événementielles I0,24 et I0,25 sont raccordées à un codeur rotatif incrémental, avec discriminateur de sens de marche (voir schéma ci-dessous).

Réaliser le programme permettant de calculer la position, en fonction du sens de marche.

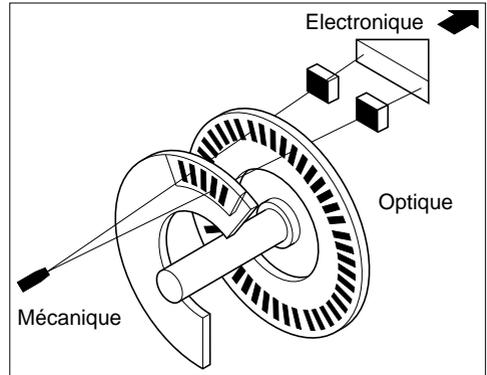
Un bouton-poussoir raccordé à l'entrée I0,01 permet d'initialiser le calcul de la position (position de référence : 0).



Principe de fonctionnement d'un codeur rotatif incrémental

Un codeur est un dispositif électromécanique, dont la sortie électrique représente sous forme numérique la position angulaire de l'axe d'entrée.

Le codeur incrémental est constitué d'un disque qui comporte deux pistes divisées en "n" intervalles d'angles égaux, alternativement opaques et transparents. "n" est appelé résolution ou nombre de périodes.

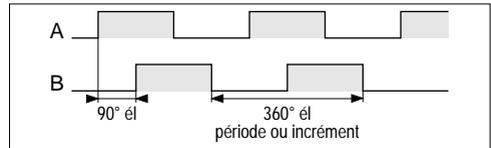


Deux photo-diodes décalées envoient deux faisceaux qui traversent les fentes du disque, délivrant en sortie, deux signaux "carré" (A et B) en quadrature de phase.

Pour un tour complet de l'axe du codeur, les faisceaux sont interrompus "n" fois et délivrent "n" signaux consécutifs.

Le déphasage (90° électrique) des signaux A et B permet de déterminer le sens de marche :

- dans un sens pendant le front montant du signal A, le signal B est à 1 (sens positif par exemple),
- dans l'autre sens pendant le front montant du signal A, le signal B est à 0 (sens négatif par exemple).



Corrigé de l'exercice

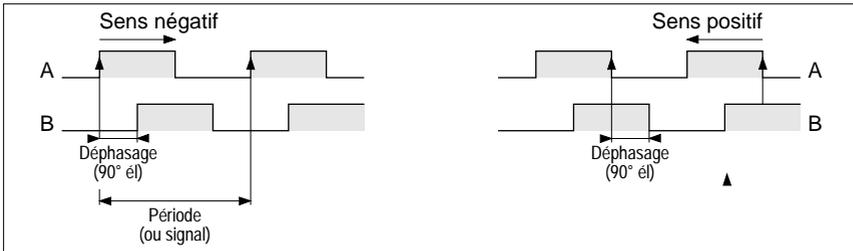
Compte-tenu de la fréquence élevée⁽¹⁾ des signaux A et B en sortie (recueillis respectivement sur I0,24 et I0,25), le traitement du positionnement sera écrit en tâche rapide.

La tâche maître sera ralentie, compte tenu des appels fréquents de la tâche rapide. En effet, cette dernière est lancée chaque fois que I0,24 change d'état physique (sur front montant ou sur front descendant).

Le mot W0, mis à jour chaque fois qu'un front montant sur I0,24 est détecté, est :

- incrémenté d'une unité pour signaler un pas dans le sens positif,
- décrémenté d'une unité pour signaler un pas dans le sens négatif.

Configurer I0,24 comme entrée événementielle. **I0,25 peut être testée en tâche rapide, bien qu'elle demeure configurée comme entrée normale.**



⁽¹⁾ : Dans le cas d'une application ayant un programme conséquent en tâche maître (réseaux avec plusieurs blocs opération et comparaison), la fréquence des signaux du codeur doit être inférieure à 300 Hz, afin de garantir leur prise en compte.

PL7-2	Formulaire programmation schéma à contacts								OBSERVATIONS	
N°	ZONE TEST					ZONE ACTION				
<u>" . P . O . S . I . T . . . I . N . I . T . I . A . L . E .</u>										
1	L	SY19								Tâche Maître
		(R)								Validation
		()								Tâche rapide
		IØ.1								Initialisation position
		IØ → WØ								
		()								
		()								
<u>" . C . A . L . C . U . L . . P . O . S . I . T . I . O . N</u>										
1	F	IØ.24 IØ.25								Tâche Rapide
		IØ.25								Calcul position
		WØ+1 → WØ								en fonction du
		WØ → WØ								sens de marche
		()								
		()								
<u>"</u>										
		()								
		()								
		()								
		()								
<u>"</u>										
		()								
		()								
		()								
		()								
L Tâche maître langage à contacts		F Tâche rapide		P Traitement préliminaire		P Traitement postérieur				
	Mise à jour	Par	Date	Etude :	Dessins :	Date :				
A.				Positionnement			Folio			
B.										
C.										

F

2.2 Compteur/temporisateur rapide en multiprélélection

Cet exercice est composé de trois parties successives qui doivent être traitées dans l'ordre :

- 1^{ère} partie : dosage par multiprélélection,
- 2^{ème} partie : affichage volume,
- 3^{ème} partie : lecture à la volée du volume instantané.

2.2-1 Description de l'application

Première partie

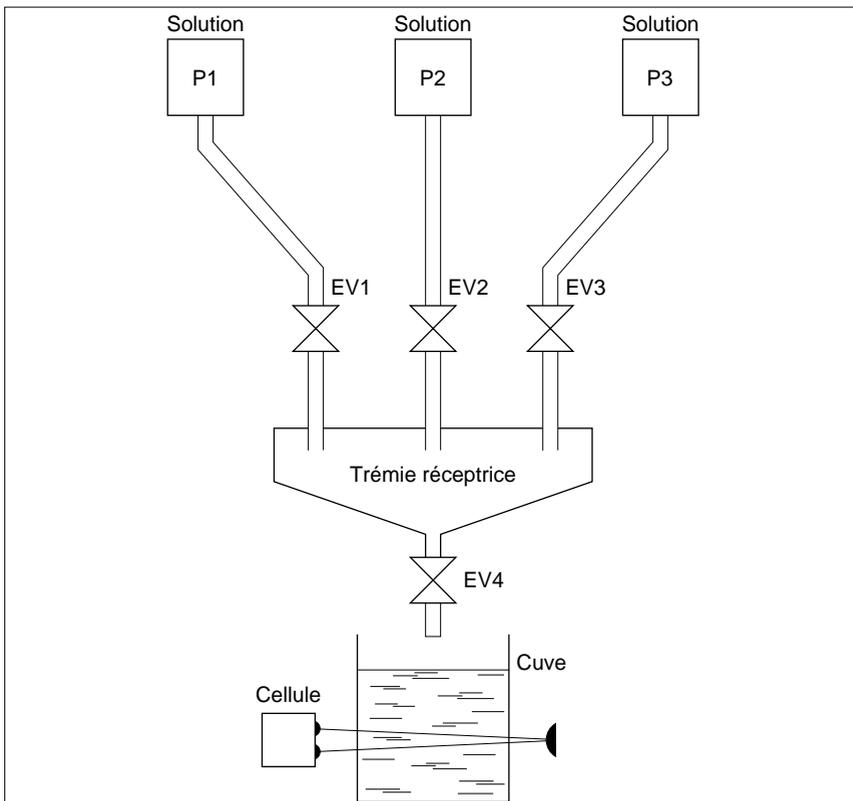
• Enoncé

Le processus de remplissage d'un produit chimique nécessite le dosage précis et successif de trois sous-produits P1, P2 et P3.

L'écoulement de ces sous-produits dans la trémie réceptrice est contrôlé respectivement par trois électrovannes EV1, EV2 et EV3. L'écoulement de la trémie réceptrice vers la cuve est contrôlé par l'électrovanne EV4.

L'ouverture des électrovannes est provoqué par la présence de tension à leurs bornes; la fermeture par l'absence de tension (électrovannes monostables).

La présence de la cuve sous la trémie est contrôlée par une cellule photoélectrique.



• Modes de marches

M1 : Les trois sous-produits P1, P2 et P3 sont dosés dans cet ordre et immédiatement l'un après l'autre. C'est la présence de la cuve qui provoque immédiatement le dosage du premier sous-produit et l'ouverture de EV4.

M2 : L'électrovanne EV4 se ferme :

- à la fin du dosage du 3^{ème} sous-produit,
- en cas de retrait de la cuve pendant la phase de remplissage,
- en cas de coupure secteur.

M3 : Une reprise secteur ou le retrait de la cuve avant la fin du cycle provoquent la réinitialisation complète du processus (pour la cuve vide suivante).

M4 : Le volume d'une cuve n'autorise que la délivrance d'une seule dose de chacun des sous-produits.

• Données de l'exercice

Les sous-produits P1, P2 et P3 sont envoyés par des doseurs ayant un débit de 1 Litre/seconde :

- P1 est dosé à 5200L,
- P2 est dosé à 3125L,
- P3 est dosé à 1649L.

Le dosage est géré par le compteur rapide. Il peut être réalisé de deux manières différentes :

- soit par l'entrée comptage rapide, où à chaque impulsion externe correspond 1L dosé,
- soit par le temporisateur rapide, où à chaque impulsion interne de 0,555ms correspond 1L dosé.

Pour les besoins de la simulation, on retiendra la deuxième solution.

La présélection du sous-produit P1 peut être définie de deux manières différentes :

- soit saisie lors de la programmation du bloc FC (FC,P = 9999 par défaut) : solution non retenue,
- soit définie dans le programme application, lors de l'initialisation (M3). Dans ce cas et seulement si un "Reset" préalable du Timer est effectué, la présélection P1 est retenue pour le premier cycle de comptage.

• Adressage entrées/sorties

EV1 = 00,01

EV2 = 00,02

EV3 = 00,03

EV4 = 00,04

Cellule présence cuve = I0,01

Deuxième partie

- **Affichage volume en BCD sur TSX 17 ACC2**

En remplaçant le module de visualisation standard du TSX 17-20 par le module numérique (réf: TSX 17 ACC2), afficher les trois volumes intermédiaires de la cuve (Vc), correspondants aux moments de passage d'un sous-produit au suivant :

$$\begin{aligned}\text{volume cuve } Vc &= P1 \\ &= P1 + P2 \\ &= P1 + P2 + P3\end{aligned}$$

En début de cycle, $Vc = 0$

$$P1 = 5200 \quad P2 + P2 = 8325 \quad P1 + P2 + P3 = 9974$$

Données complémentaires : I0,02 = valide l'affichage du contenu de SW16.

Troisième partie

- **Lecture à la volée du volume instantané**

En actionnant un bouton-poussoir (Bp) raccordé à l'entrée 3 (I0,03), l'opérateur peut visualiser sur l'afficheur numérique le volume instantané de la cuve (Vi). Cette valeur reste affichée et figée tant que le bouton est actionné.

Il peut être effectué durant un cycle de remplissage de la cuve autant de lecture à la volée que d'actions sur le bouton-poussoir.

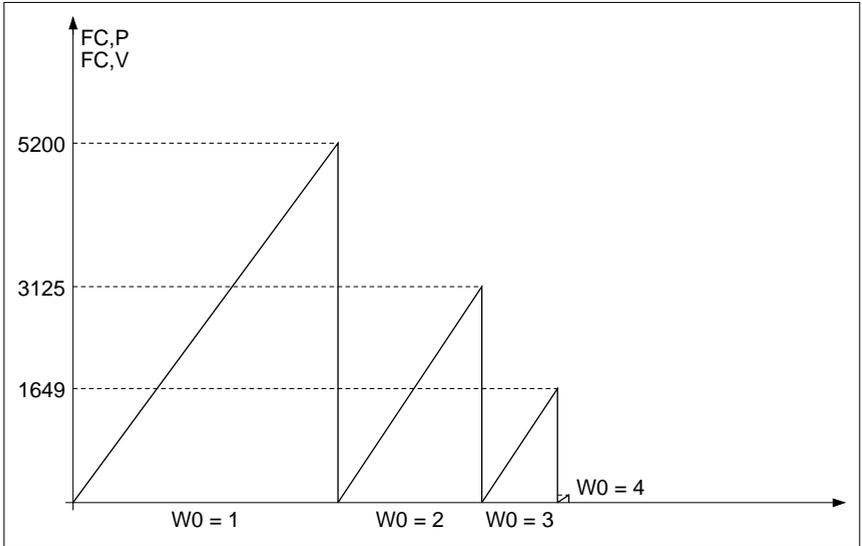
La lecture à la volée ne doit pas empêcher l'évolution du dosage et l'affichage de Vc (lorsque Bp n'est pas actionné).

Données complémentaires : I0,03 = bouton-poussoir Bp.

2.2-2 Données de programmation.

Pour la première partie

- Configurer FC en Timer rapide
- Configurer l'octet de sortie O0,00 à O0,07 en tâche rapide.
- Le mot W0 sera incrémenté en tâche rapide afin que chacune des valeurs prises représente une phase de dosage d'un sous-produit.



Nota : Pour visualiser le fonctionnement du timer rapide en mode REGLAGE, l'entrée V du bloc fonction FC (C31 en ADJUST) doit être à 1.

Pour la deuxième partie

Le mot W16 sera utilisé comme mot de "chargement" des volumes intermédiaires de la cuve. Il doit être converti en code BCD avant d'être transféré dans SW16.

Pour la troisième partie

Le mot W17 sera utilisé comme mot de "chargement" du volume instantané de la cuve. Il doit être converti en code BCD avant d'être transféré dans SW16.

- I0,03 = 0 affichage de Vc,
- I0,03 = 1 affichage de Vi.

PL7-2		Formulaire programmation schéma à contacts									
N°	ZONE TEST									ZONE ACTION	OBSERVATIONS
" M.O.D.E.S. D.E. M.A.R.C.H.E "											
1	SY0									B0	bit de mémorisation.....
0	SY1									(S)	
L	J0,1									(S)	Modes de Marches M2
										B0	
										(R)	
										SY9	
										(R)	
" J.N.L.T.I.A.L.L.S.A.T.I.O.N "											
2	SY0										Présélection P1.....
0	SY1									[5200] → FC,P]	
L	J0,1									[0] → W0]	Modes de Marches M3
										SY19	
										(R)	Validation Tâche Rapide
										O0,4	
										(R)	
" P.R.E.S. C.U.V.E. "											
3	J0,1										Sur front élané un seul cycle de remplissage
0	B4									B5	
L										()	Modes de Marches M4
										()	
										()	
										()	
" J.A.N.C.F.M.F.N.T. T.A.C.H.E "											
4	B5										
0										[1] → W0]	
L										O0,4	Modes de Marches M1
										(S)	
											Présélection P2
										[3125] → FC,P]	
										C0,01	dosage P1
										()	
L Tâche maître langage à contacts F Tâche rapide P Traitement préliminaire P Traitement postérieur											
		Mise à jour _____ A. _____ B. _____ C. _____	Par _____	Date _____	Etude : _____	Dessins : _____	Date : _____	Telemecanique		Folio 1 / 1	
Langage à contacts 1^{ère} partie : dosage par multiprésentation											

F

PL7-2	Formulaire programmation schéma à contacts						
N°	ZONE TEST	←----->	ZONE ACTION	OBSERVATIONS			
" P.R.E.S.E.L.E.C. . . . P.3 . . . "							
5 Ø L	W0=2	FC	P	1649	FC	P	Présélection P3
	Ø0,2						dosage P2
	Ø0,3						dosage P3
	W0=3						(présélection P1 dans label 20)
" T.I.M.E.R. R.A.P.I.D.E. . . . "							
6 Ø L	B0	FC					
	B5	R					(B0,B5) : Modes de
	B10	P-D					Marches M2 et M3
	I0,1	V					
	W0=0	C					validation dosage
" "							
Ø Ø Ø							
" "							
Ø Ø Ø							
<div style="display: flex; justify-content: space-between;"> L Tâche maître langage à contacts F Tâche rapide P Traitement préliminaire P Traitement postérieur </div>							
	Mise à jour	Par	Date	Etude :	Dessins :	Date :	
	A. _____	_____	_____	Langage à contacts			
	B. _____	_____	_____	1 ^{ère} partie : dosage par multiprésélection			
C. _____	_____	_____				Folio 2 / 1	

F

PL7-2	Formulaire programmation schéma à contacts						
N°	ZONE TEST	←	→	ZONE ACTION	OBSERVATIONS		
	" . A . F . F . I . C . H . A . G . E . _____						
	SY15				Tâche RAPIDE		
5	W0-1	W0-2	W0-3	[5200] → W16	insertion réseau en début de programme tâche rapide		
F	W0-2	W0-3	[8325] → W16				
F	W0-3	W0-4	[9974] → W16				
	_____ () _____						
	" . P . R . E . S C . U . V . E . _____						
	I0.1 B4				B5	Langage à contacts modification label 30	
3	W0-1	W0-2	W0-3	()	Initialisation affichage		
0	W0-2	W0-3	[0] → W16				
L	W0-3	W0-4	()				
	_____ () _____				B4		
	_____ () _____						
	" . P . R . E . S . E . L . E . C P . 3 . . . _____						
	I0.2				SY14	Langage à contacts modification label 50	
5	W0-2	W0-3	W0-4	[1649] F[P]	validation affichage SW16		
0	W0-3	W0-4	()	00.02			
L	W0-4	W0-5	()	00.03			
	_____ () _____						
	_____ () _____						
	" . C . O . N . V . E . R B . C . D _____						
	I0.2				SY14	conversion BCD en affichage	
7	W0-1	W0-2	W0-3	[BCD W16] → SW16	Langage à contacts label à saisir à la suite		
0	W0-2	W0-3	()				
L	W0-3	W0-4	()				
	_____ () _____						
	_____ () _____						
L Tâche maître langage à contacts		F Tâche rapide		P Traitement préliminaire		P Traitement postérieur	
	Mise à jour	Par	Date	Etude :	Dessins :	Date :	
A. _____	_____	_____	_____	Multiprésélection pour dosage 2 ^{ème} partie : Affichage volume			Folio
B. _____	_____	_____	_____				1
C. _____	_____	_____	_____				/ 2

F

F



Sous chapitre	page
3.1 Description de la machine	3/2
3.2 Grafcet du cahier des charges (première partie)	3/3
3.2-1 Description des actionneurs et des préactionneurs	3/4
3.2-2 Conditions initiales	3/5
3.2-3 Tableau d'adressage des entrées/sorties	3/6
3.2-4 Programme	3/7
3.3 Modes de marche et d'arrêt (deuxième partie)	3/14
3.3-1 Enoncé	3/14
3.3-2 Complément d'adressage des entrées/sorties	3/15
3.3-3 Programme	3/16
3.4 Comptage des pièces (troisième partie)	3/22
3.4-1 Programme	3/23
3.5 Communication par bloc texte (quatrième partie)	3/24
3.5-1 Affichage défauts sur imprimante	3/24
3.5-2 Datation des défauts (avec TSX 17-20 muni de l'option horodateur)	3/26
Ce chapitre se termine à la page	3/26

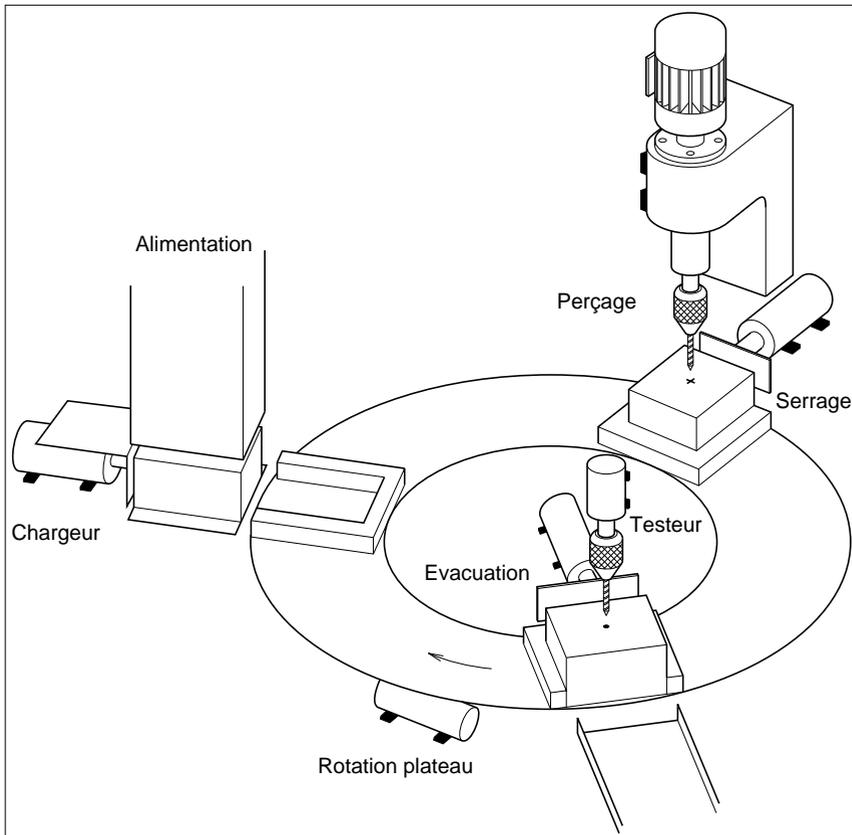
3.1 Description de la machine

Il s'agit d'une machine automatique de perçage comportant un plateau tournant qui dessert 3 postes de travail :

- Poste 1 : chargement des pièces sur palettes.
- Poste 2 : bridage des palettes et perçage des pièces.
- Poste 3 : contrôle des pièces percées, puis évacuation. Le contrôle de perçage s'effectue par un testeur qui doit descendre en position basse si le trou est correctement percé. Si cela n'est pas réalisé, tout le système se bloque, testeur en position haute, de façon à ce que l'opérateur puisse enlever la pièce défectueuse avant de réarmer manuellement le système.

Un vérin permet la rotation de 120° du plateau extérieur supportant les pièces à usiner. Il permet aussi son indexation, c'est à dire son blocage précis après chaque rotation.

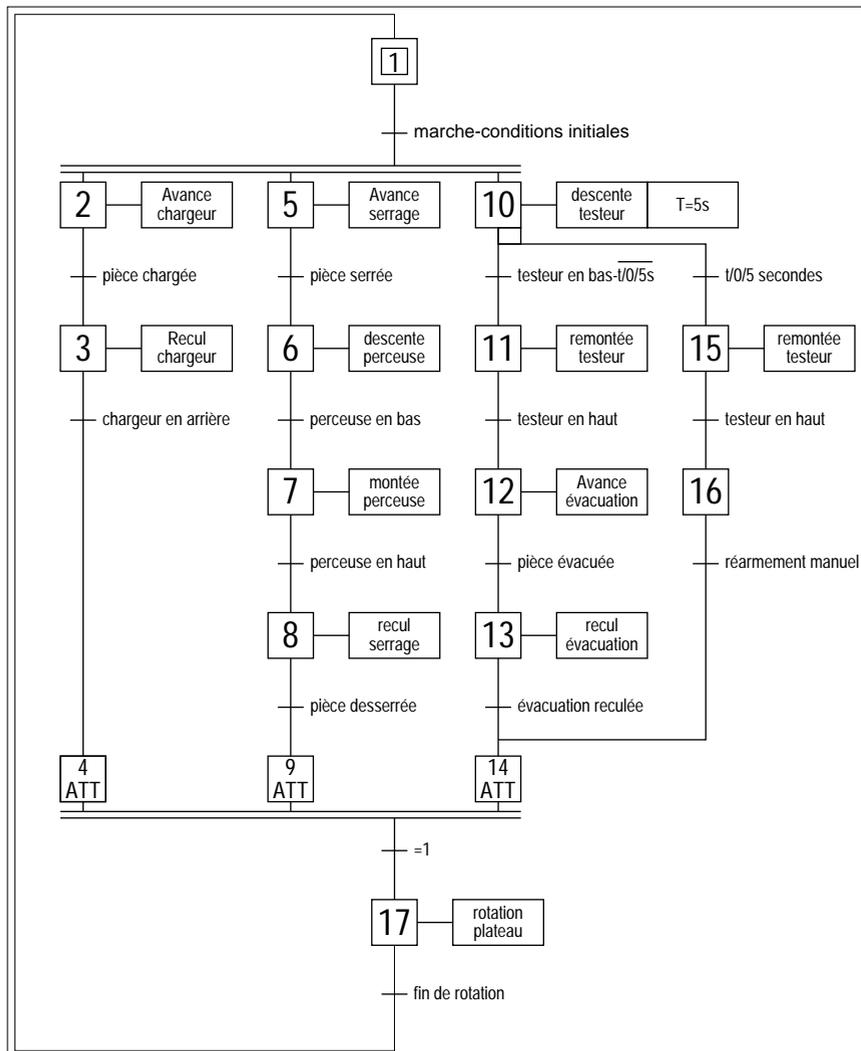
On supposera que le moteur de la perceuse est actionnée grâce à un système mécanique, commandé lors du mouvement de translation vertical.



3.2 GRAFCET du cahier des charges (Première partie)

Le GRAFCET décrivant le fonctionnement de cette installation est représenté ci-dessous. Lorsque l'ordre "marche" apparaît et à condition que la partie opérative soit correctement positionnée, le franchissement de la transition conduit à l'activation simultanée des étapes 2, 5 et 10. A partir de cette situation, les trois séquences (2-3-4), (5 à 9) et (10 à 14) évolueront indépendamment les unes des autres mais elles devront être toutes achevées pour aboutir à une évolution commune à l'étape 17 :

- La séquence 2-3-4 évoluera jusqu'à l'étape 4 où elle s'arrêtera.
- La séquence 5 à 9 évoluera jusqu'à l'étape 9 où elle s'arrêtera.
- La séquence 10 à 14 évoluera jusqu'à l'étape 14 où elle s'arrêtera.



Lorsque les étapes 4, 9 et 14 correspondantes aux attentes des trois séquences citées seront actives, l'évolution pourra s'effectuer à l'étape 17 puisque la réceptivité notée "=1" est toujours vraie. Ces trois étapes seront appelées "étapes d'attente" et n'ont, du fait de leur rôle de synchronisation, aucune action à effectuer.

La transition de l'étape 10 à l'étape 11 tient compte de la position basse du testeur et de la temporisation de 5 secondes non achevée, afin d'assurer l'exclusion avec la réceptivité de la transition de l'étape 10 à l'étape 15.

3.2-1 Description des actionneurs et des préactionneurs

Les mouvements commandés sur les trois postes (voir page ci-contre), sont régis par des vérins pneumatiques, commandés par des interfaces modulaires électropneumatiques :

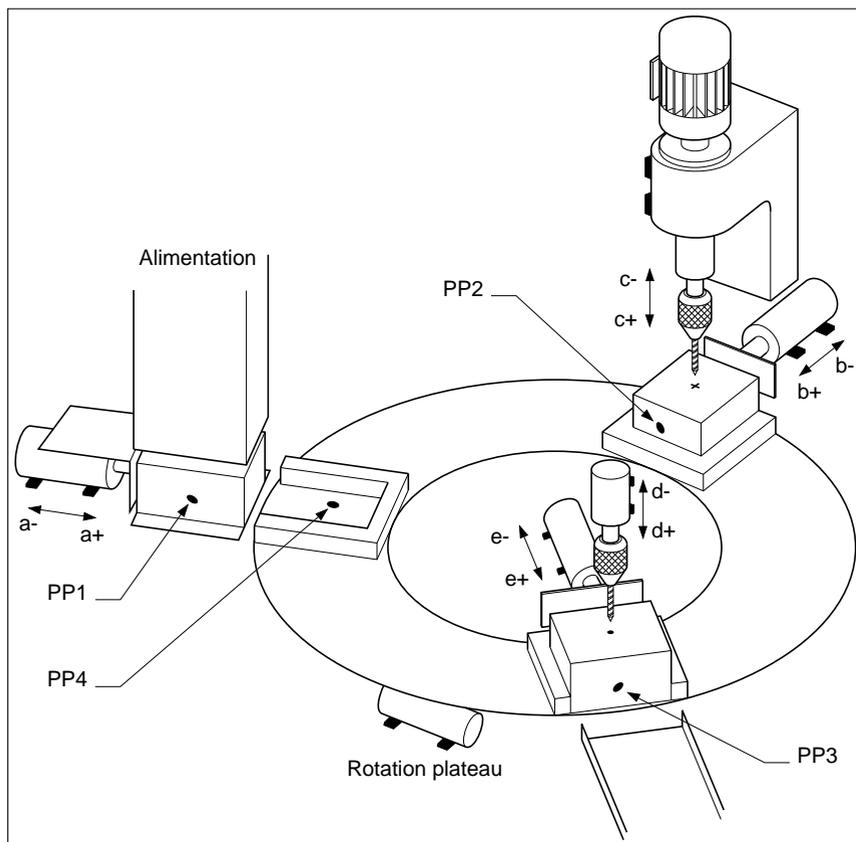
- vérins double effet : A, B, C, D et E.
- vérin simple effet : F.

Des détecteurs de fin de course sont intégrés aux vérins pour détecter les positions extrêmes.

3.2-2 Conditions initiales

A l'état initial, la machine doit répondre aux conditions suivantes :

- tige vérin A de chargement rentrée (fdca-),
- tige vérin B de serrage rentrée (fdcb-),
- tige vérin C de translation perceuse rentrée (fdcc-),
- tige vérin D de translation testeur rentrée (fdcd-),
- tige vérin E d'évacuation rentrée (fdce-),
- une pièce au moins présente devant un poste (pp1 OU pp2 OU pp3 : détecteurs disposés sous le plateau tournant).



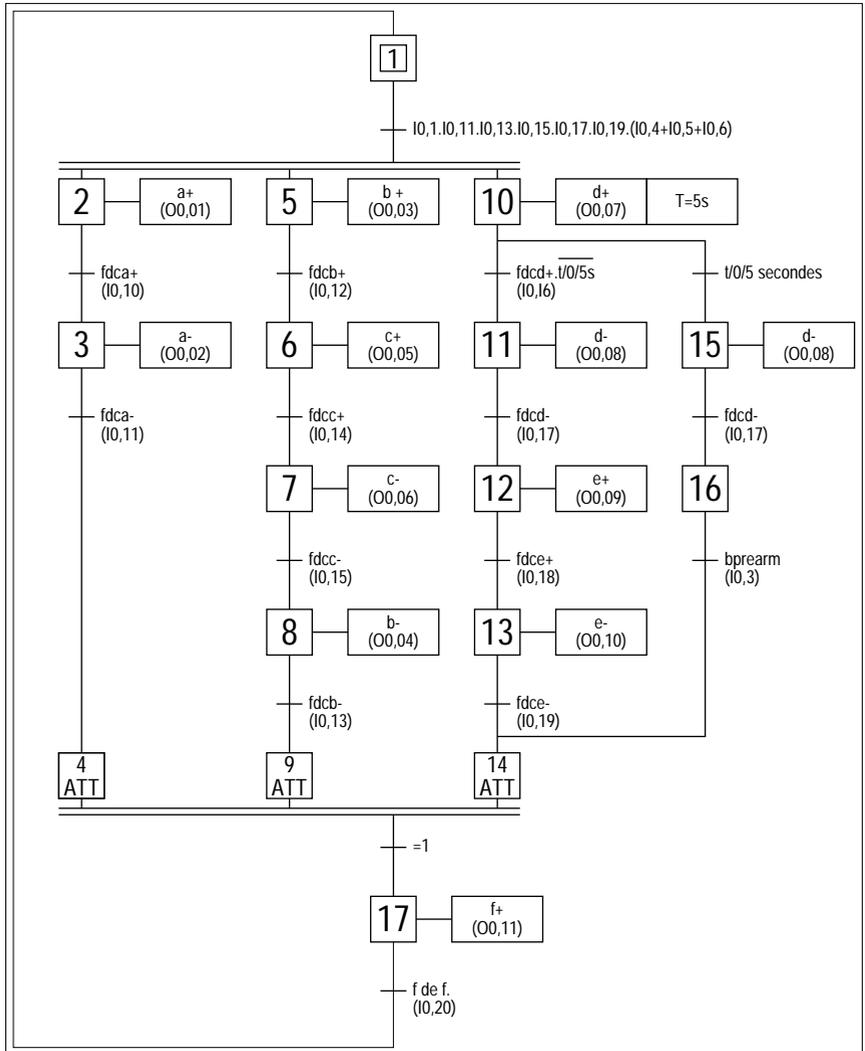
3.2-3 Tableau d'adressage des entrées/sorties

10,1	bpm	bouton-poussoir "Marche"
10,3	bprearm	bouton-poussoir "Réarmement manuel"
10,4	pp1	détecteur présence pièce poste de chargement
10,5	pp2	détecteur présence pièce poste de perçage
10,6	pp3	détecteur présence pièce poste de contrôle
10,10	fdca+	fin de course travail vérin A tige sortie
10,11	fdca-	fin de course repos vérin A tige rentrée
10,12	fdcb+	fin de course travail vérin B tige sortie
10,13	fdcb-	fin de course repos vérin B tige rentrée
10,14	fdcc+	fin de course travail vérin C tige sortie
10,15	fdcc-	fin de course repos vérin C tige rentrée
10,16	fdcd+	fin de course travail vérin D tige sortie
10,17	fdcd-	fin de course repos vérin D tige rentrée
10,18	fdce+	fin de course travail vérin E tige sortie
10,19	fdce-	fin de course repos vérin E tige rentrée
10,20	fdcf+	fin de course travail vérin F tige sortie
O0,01	a+	avance vérin A
O0,02	a-	recul vérin A
O0,03	b+	avance vérin B
O0,04	b-	recul vérin B
O0,05	c+	avance vérin C
O0,06	c-	recul vérin C
O0,07	d+	avance vérin D
O0,08	d-	recul vérin D
O0,09	e+	avance vérin E
O0,10	e-	recul vérin E
O0,11	f+	avance vérin F

Nota : Cet adressage correspond à l'utilisation d'un TSX 17-20.

3.2-4 Programme

- Saisir le GRAFCET et les transitions associées dans la zone de traitement séquentiel.
- Saisir les actions associées aux étapes dans la zone de traitement postérieur.





Formulaire programmation Grafcet

PL7-2	Formulaire programmation Grafcet													COMMENTAIRES / OBSERVATIONS			
C00	C01	C02	C03	C04	C05	C06	C07						Date :				
L00	▼ 1Z	▽	▽	▽	▽	▽	▽	▽	▽	▽	▼ 04						
L01	□ 01										□ 1Z						
L02																	
L03	□ 02	□ 05	□ 10								▶ 01						
L04																	
L05	□ 03	□ 06	□ 11	□ 15													
L06																	
L07	□	□ 07	□ 12	□ 16													
L08																	
L09	□	□ 08	□ 13														
L10																	
L11	□ 04	□ 09	□ 14														
L12																	
L13	▼ 1Z	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽	▽
<p>01 : étape initiale 1Z : Rotation plateau</p> <p>marche et conditions initiales fin de rotation</p> <p>A+ : B+ d+ 1/0/1/5s 1/0/1/5s u.</p> <p>A- C+ D- D-</p> <p>a- C+ d- d-</p> <p>C- E+ E-</p> <p>c- e+ bprearm</p> <p>B- E- E-</p> <p>b- e- e-</p> <p>étapes d'attente</p> <p>reception toujours vraie (=1)</p>																	
<p>N° Page Grafcet</p> <p>Mise à jour</p> <p>A. _____</p> <p>B. _____</p> <p>C. _____</p>																	
<p>Etude : _____</p> <p>Date : _____</p> <p>Par : _____</p>																	
<p>Machine de perçage : GRAFCET de fonctionnement</p> <p>Traitement séquentiel (SE0)</p>																	
<p>TSX</p>																	
<p>Thermocentrique</p> <p>Folio</p> <p style="font-size: 2em;">1 / 6</p>																	

Saisie des réceptivités associées aux transitions

PL7-2	Formulaire programmation schéma à contacts									
N°	ZONE TEST						ZONE ACTION	OBSERVATIONS		
<input checked="" type="checkbox"/> X <input type="checkbox"/> 0 <input type="checkbox"/> 1 <input type="checkbox"/>	IO,4 IO,1 IO,11 IO,13 IO,15 IO,17 IO,19 _____ X02									
	_____ (#) X01 → X02									
	IO,5 _____									
	_____ ()									
	IO,6 _____									
<input checked="" type="checkbox"/> X <input type="checkbox"/> 0 <input type="checkbox"/> 2 <input type="checkbox"/>	IO,10 _____ X03									
	_____ (#) X02 → X03									
	_____ ()									
	_____ ()									
	_____ ()									
<input checked="" type="checkbox"/> X <input type="checkbox"/> 0 <input type="checkbox"/> 3 <input type="checkbox"/>	IO,11 _____ X04									
	_____ (#) X03 → X04									
	_____ ()									
	_____ ()									
	_____ ()									
<input checked="" type="checkbox"/> X <input type="checkbox"/> 0 <input type="checkbox"/> 4 <input type="checkbox"/>	IO,12 _____ X17									
	_____ (#) X04 → X17									
	_____ ()									
	_____ ()									
	_____ ()									
[L] Tâche maître langage à contacts [F] Tâche rapide [P] Traitement préliminaire [P] Traitement postérieur										
	Mise à jour	Par	Date	Etude :	Dessins :	Date :				
	A. _____	_____	_____	Machine de perçage : Réceptivités Traitement Séquentiel (SEQ)				Folio 2 / 6		
	B. _____	_____	_____							
C. _____										

F

PL7-2

Formulaire programmation schéma à contacts

N°	ZONE TEST	ZONE ACTION	OBSERVATIONS
<input checked="" type="checkbox"/> X <input type="checkbox"/> 0 <input type="checkbox"/> 5 <input type="checkbox"/>	10,12	X06	X05 → X06
	_____ (#)	_____	_____
	_____ ()	_____	_____
	_____ ()	_____	_____
	_____ ()	_____	_____
<input checked="" type="checkbox"/> X <input type="checkbox"/> 0 <input type="checkbox"/> 6 <input type="checkbox"/>	10,14	X07	X06 → X07
	_____ (#)	_____	_____
	_____ ()	_____	_____
	_____ ()	_____	_____
	_____ ()	_____	_____
<input checked="" type="checkbox"/> X <input type="checkbox"/> 0 <input type="checkbox"/> 7 <input type="checkbox"/>	10,15	X09	X07 → X08
	_____ (#)	_____	_____
	_____ ()	_____	_____
	_____ ()	_____	_____
	_____ ()	_____	_____
<input checked="" type="checkbox"/> X <input type="checkbox"/> 0 <input type="checkbox"/> 8 <input type="checkbox"/>	10,13	X09	X08 → X09
	_____ (#)	_____	_____
	_____ ()	_____	_____
	_____ ()	_____	_____
	_____ ()	_____	_____

<input type="checkbox"/> Tâche maître langage à contacts	<input type="checkbox"/> Tâche rapide	<input type="checkbox"/> Traitement préliminaire	<input type="checkbox"/> Traitement postérieur
--	---------------------------------------	--	--

	Mise à jour	Par	Date	Etude :	Dessins :	Date :		
	A	_____	_____	_____	_____	_____		Modes de Marches et d'Arrêts : Traitement Séquentiel (transitions)
	B	_____	_____	_____	_____	_____		
C	_____	_____	_____	_____	_____	_____	Folio 3 / 6	

F

PL7-2	Formulaire programmation schéma à contacts							
N°	ZONE TEST					ZONE ACTION	OBSERVATIONS	
<input checked="" type="checkbox"/> X <input checked="" type="checkbox"/> 1 <input type="checkbox"/> 0	I0,16	I0,D					X11	(#) X10 → X11 () () () ()
<input checked="" type="checkbox"/> X <input checked="" type="checkbox"/> 1 <input checked="" type="checkbox"/> 1 <input type="checkbox"/>	I0,17						X12	(#) X11 → X12 () () () ()
<input checked="" type="checkbox"/> X <input checked="" type="checkbox"/> 1 <input checked="" type="checkbox"/> 2 <input type="checkbox"/>	I0,18						X13	(#) X12 → X13 () () () ()
<input checked="" type="checkbox"/> X <input checked="" type="checkbox"/> 1 <input checked="" type="checkbox"/> 3 <input type="checkbox"/>	I0,19						X14	(#) X13 → X14 () () () ()

L Tâche maître langage à contacts
 F Tâche rapide
 P Traitement préliminaire
 P Traitement postérieur

	Mise à jour	Par	Date	Etude :	Dessins :	Date :	
	A. _____	_____	_____	_____	_____	_____	
	B. _____	_____	_____	_____	_____	_____	
C. _____	_____	_____	_____	Modes de Marches et d'Arrêts : Traitement Séquentiel (transitions)		Folio 4 / 6	

F

PL7-2

Formulaire programmation schéma à contacts

N°	ZONE TEST	←----->	ZONE ACTION	OBSERVATIONS
<input checked="" type="checkbox"/> X <input type="checkbox"/> 1 <input type="checkbox"/> 0 <input type="checkbox"/>	$\overline{10,D}$	$X15$	(#)	$X10 \rightarrow X15$
	_____	_____	_____	_____
	_____	_____	_____	_____
	_____	_____	_____	_____
	_____	_____	_____	_____
<input checked="" type="checkbox"/> X <input type="checkbox"/> 1 <input checked="" type="checkbox"/> 5 <input type="checkbox"/>	$\overline{10,17}$	$X16$	(#)	$X15 \rightarrow X16$
	_____	_____	_____	_____
	_____	_____	_____	_____
	_____	_____	_____	_____
	_____	_____	_____	_____
<input checked="" type="checkbox"/> X <input type="checkbox"/> 1 <input checked="" type="checkbox"/> 6 <input type="checkbox"/>	$\overline{10,3}$	$X14$	(#)	$X16 \rightarrow X14$
	_____	_____	_____	_____
	_____	_____	_____	_____
	_____	_____	_____	_____
	_____	_____	_____	_____
<input checked="" type="checkbox"/> X <input type="checkbox"/> 1 <input checked="" type="checkbox"/> 7 <input type="checkbox"/>	$\overline{10,20}$	$X01$	(#)	$X17 \rightarrow X01$
	_____	_____	_____	_____
	_____	_____	_____	_____
	_____	_____	_____	_____
	_____	_____	_____	_____

<input type="checkbox"/> Tâche maître langage à contacts	<input type="checkbox"/> Tâche rapide	<input type="checkbox"/> Traitement préliminaire	<input type="checkbox"/> Traitement postérieur
--	---------------------------------------	--	--

	Mise à jour _____	Par _____	Date _____	Etude : _____	Dessins : _____	Date : _____	Telemecanique
Modes de Marches et d'Arrêts : Traitement Séquentiel (transitions)							Folio 5 / 6

F

PL7-2		Formulaire programmation schéma à contacts										OBSERVATIONS	
N°	ZONE TEST										ZONE ACTION	OBSERVATIONS	
" . A . C . T . I . O . N . S .													
2 Ø P	X2											ØØ,1	Actions associées
												()	aux étapes :
	X3											ØØ,2	(2,3,5 et 6)
												()	
	X5											ØØ,3	
											()		
	X6										ØØ,5		
											()		
" . A . C . T . I . O . N . S .													
3 Ø P	X7											ØØ,6	Actions associées
												()	aux étapes :
	X8											ØØ,4	(7,8,11 et 15)
												()	
	X11											ØØ,8	
											()		
	X15										()		
											()		
" . A . C . T . I . O . N . S . X . 1 . Ø .													
5 Ø P	X1Ø											ØØ,7	Actions associées
												()	à l'étape 1Ø :
												()	• avance vérin D
												()	• lancement tempo
												()	
											()		
" . A . C . T . I . O . N . S .													
7 Ø P	X12											ØØ,9	Actions associées
												()	aux étapes :
	X13											ØØ,1Ø	(12,13 et 17)
												()	
	X17											ØØ,11	
											()		
											()		

<input type="checkbox"/> L	Tâche maître langage à contacts	<input type="checkbox"/> F	Tâche rapide	<input type="checkbox"/> P	Traitement préliminaire	<input type="checkbox"/> P	Traitement postérieur
	Mise à jour	Par	Date	Etude :	Dessins :	Date :	
	A.			Machine automatique de perçage Traitement Postérieur (PCS)			Folio
	B.						6 / 6
C.							

F

3.3 Modes de Marche et d'Arrêts (deuxième partie)

3.3-1 Enoncé

La machine décrite dans la première partie, possède 3 modes de fonctionnement : AUTO / MANU / Cycle par cycle.

Marche AUTO : Lorsque le commutateur est sur position "AUTO", l'action sur l'interrupteur "Départ auto" lance l'évolution du GRAFCET décrit précédemment.

Si les conditions initiales ne sont pas vérifiées, une Marche de préparation doit se lancer automatiquement, afin de les satisfaire et de permettre le démarrage.

Marche MANU : Lorsque ce même commutateur est sur position "MANU", le GRAFCET doit être inhibé (mise à 0 de toutes les étapes), et chaque actionneur doit pouvoir être commandé manuellement par un bouton-poussoir.

Marche Cycle/Cycle : Lorsque ce même commutateur est sur position "Cycle/Cycle", une impulsion sur le bouton-poussoir "Départ cycle" lance l'exécution d'un seul cycle machine (cycle GRAFCET).

Arrêt d'Urgence : En cas d'Arrêt d'Urgence par action sur un bouton poussoir coup de poing, les sorties sont toutes mises à 0. La reprise, à partir de l'état initial, ne peut s'effectuer que par réarmement manuel (bouton-poussoir défini dans la première partie : I0,3 Bprearm).

Reprise secteur : Reprise à froid ou reprise à chaud, elle doit s'effectuer à partir de l'état initial.

Réglage et mise au point machine : Afin de pouvoir intervenir sur la machine, un commutateur à clé à deux positions permet (en position travail), de figer le GRAFCET dans l'état. La position repos autorise la reprise.

Conditions initiales : La machine doit pouvoir fonctionner tant qu'il y a une pièce présente devant un poste (détecteurs de présence pièce "pp."). De même, elle doit pouvoir fonctionner s'il y a deux pièces au poste de chargement :

- une pièce sur la palette (détecteur de présence pp4),
- une pièce prête à être chargée (détecteur de présence pp1).

Un voyant s'allume lorsque les conditions initiales sont satisfaites.

3.3-2 Complément d'adressage des entrées/sorties

I0,1	bpm	bouton-poussoir "Marche"
I0,2	bpcycl	bouton-poussoir "Départ cycle"
I0,3	bprearm	bouton-poussoir "Réarmement manuel"
I0,4	pp1	détecteur présence pièce poste de chargement
I0,5	pp2	détecteur présence pièce poste de perçage
I0,6	pp3	détecteur présence pièce poste de contrôle
I0,7	pp4	détecteur présence pièce poste sur palette
I0,8	AUTO	commutateur en position AUTO
I0,9	MANU	commutateur en position MANU
I0,10	fdca+	fin de course travail vérin A tige sortie
I0,11	fdca-	fin de course repos vérin A tige rentrée
I0,12	fdcb+	fin de course travail vérin B tige sortie
I0,13	fdcb-	fin de course repos vérin B tige rentrée
I0,14	fdcc+	fin de course travail vérin C tige sortie
I0,15	fdcc-	fin de course repos vérin C tige rentrée
I0,16	fdcd+	fin de course travail vérin D tige sortie
I0,17	fdcd-	fin de course repos vérin D tige rentrée
I0,18	fdce+	fin de course travail vérin E tige sortie
I0,19	fdce-	fin de course repos vérin E tige rentrée
I0,20	fdcf+	fin de course travail vérin F tige sortie
I0,21	AU	bouton coup de poing Arrêt d'Urgence
I1,0	bpiniC0	bouton-poussoir init. comp. C0 (3 ^e partie)
I1,1	ATregl	commutateur position figeage pour réglage
I1,10	manua+	marche manuelle avance vérin A
I1,11	manua-	marche manuelle recul vérin A
I1,12	manub+	marche manuelle avance vérin B
I1,13	manub-	marche manuelle recul vérin B
I1,14	manuc+	marche manuelle avance vérin C
I1,15	manuc-	marche manuelle recul vérin C
I1,16	manud+	marche manuelle avance vérin D
I1,17	manud-	marche manuelle recul vérin D
I1,18	manue+	marche manuelle avance vérin E
I1,19	manue-	marche manuelle recul vérin E
I1,20	manuf+	marche manuelle avance vérin F
O0,01	a+	avance vérin A
O0,02	a-	recul vérin A
O0,03	b+	avance vérin B
O0,04	b-	recul vérin B
O0,05	c+	avance vérin C
O0,06	c-	recul vérin C
O0,07	d+	avance vérin D
O0,08	d-	recul vérin D
O0,09	e+	avance vérin E
O0,10	e-	recul vérin E
O0,11	f+	avance vérin F
O1,0	VCI	voyant cond. initiales vérifiées

Le programme de la deuxième partie est donné dans les pages ci-après. Il doit être saisi à la suite du précédent.

3.3-3 Programme

PL7-2		Formulaire programmation schéma à contacts										
N°	ZONE TEST										ZONE ACTION	OBSERVATIONS
" . R . E . P . R . I . S . E . . S . E . C . T . E . U . R												
1 Ø p	SYØ										BØ	Sur reprise à froid.....
	SY1										(S)	ou reprise à chaud.....
	BØ										(S)	ou Arrêt d'urgence.....
	B21	IØ,3									()	: initialisation.GRAF CET.....
	BØ											()
" . A . U . T . O . - . M . A . N . U . - . C . Y . / . C . Y												
2 Ø p	IØ,8										SY21	Marche Automatique.....
	IØ,9										B8	ou fin marche manuelle.....
	IØ,9										()	
	IØ,2										SY22	Marche manuelle.....
	IØ,2										B2	Départ cycle/cycle.....
" . A . R . R . E . T . . D . ' . U . R . G . E . N . C . E												
3 Ø p	IØ,21										B21	Arrêt d'urgence.....
	IØ,21										(S)	
	IØ,21										SY9	
	IØ,21										(S)	
	IØ,21										()	
" . D . E . P . A . R . T . - . R . E . A . R . M . E . .												
4 Ø p	IØ,1	IØ,8	IØ,9								BØ	Départ auto.....
	B2	IØ,8									B1	ou cycle/cycle.....
	B21	IØ,3	IØ,21								()	
	B21	IØ,3	IØ,21								SY9	Réarmement manuel.....
	B21	IØ,3	IØ,21								(R)	

Tâche maître langage à contacts
 Tâche rapide
 Traitement préliminaire
 Traitement postérieur

	Mise à jour _____ A _____ B _____ C _____	Par _____ _____ _____	Date _____ _____ _____	Etude : _____ Dessins : _____ Date : _____		Folio 1 / 6
--	--	-----------------------------	------------------------------	--	--	----------------

*Machine automatique de perçage
Traitement préliminaire (PRE)*

PL7-2	Formulaire programmation schéma à contacts								OBSERVATIONS
N°	ZONE TEST						ZONE ACTION		
" . C . O . N . D . . . I . N . I . T . I . A . L . E . S									
	10,4	10,11	10,13	10,15	10,17	10,19	O1Ø		
5	10,5						()	Si conditions initiales vérifiées : voyant allumé	
Ø							B4		
p	10,6						(R)		
							()		
							()		
" . C . . P . R . E . P . A . . R . E . G . L . A . G . E									
	X1	B1	O1,Ø				B4		
6							(S)	lancement cycle de préparation.	
Ø	11,1						SY23		
p							(S)		
							()		
							()		
"									
Ø							()		
Ø							()		
Ø							()		
Ø							()		
"									
Ø							()		
Ø							()		
Ø							()		
Ø							()		
L Tâche maître langage à contacts		F Tâche rapide		p Traitement préliminaire		P Traitement postérieur			

Mise à jour

A. _____

B. _____

C. _____

Par

Date

Etude :

Dessins :

Date :

Telmeccanique

Folio

2 / 6

Machine automatique de perçage:

Traitement préliminaire (PRE)

F

PL7-2		Formulaire programmation schéma à contacts					
N°	ZONE TEST			ZONE ACTION		OBSERVATIONS	
<input checked="" type="checkbox"/> X <input type="checkbox"/> Ø <input type="checkbox"/> 1	01,Ø B1	_____	_____	_____	_____	XØ,2	
	(#)						Modification
	()						transition : XØ,1 → XØ,2
	()						voyant C.I. et
	()						départ
<input checked="" type="checkbox"/> X <input type="checkbox"/> Ø <input type="checkbox"/> 2	10,10	_____	_____	_____	_____	XØ,3	
	(#)						Modification
	()						transition: XØ,2 → XØ,3
	()						pièce chargée ou
	()						déjà présente (P4)
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	_____	_____	_____	_____	_____	()	
	()						
	()						
	()						
	()						
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	_____	_____	_____	_____	_____	()	
	()						
	()						
	()						
	()						

Tâche maître langage à contacts
 Tâche rapide
 Traitement préliminaire
 Traitement postérieur

	Mise à jour	Par	Date	Etude :	Dessins :	Date :	Ternemantique
	A. _____	_____	_____	Modes de Marches et d'Arrêts: Traitement Séquentiel (transitions)			
	B. _____	_____	_____				
C. _____	_____	_____	Folio	3	/	6	

PL7-2		Formulaire programmation schéma à contacts											
N°		ZONE TEST										ZONE ACTION	OBSERVATIONS
1 Ø P		" _____ "											
		B2										B2	(R) Départ cycle/cycle
		B4										100	(J) Saut vers programme cycle de préparation
		10,8	10,9									120	(J) Saut vers programme marche manuelle
													()
2 Ø P		" A.C.T.I.O.N.S. _____ "											
		X2	10,7									00,1	() modification action
		X3										00,2	() associée à l'étape
		X5										00,3	() X2
		X6										00,5	()
		" _____ "											
													() Réseaux de contacts
													() 30 et 50
													() inchangés
													()
7 Ø P		" _____ "											
		X12										00,9	()
		X13										00,10	()
		X17										00,11	()
												150	(J) Saut vers fin postérieur

L Tâche maître langage à contacts
F Tâche rapide
P Traitement préliminaire
P Traitement postérieur

Mise à jour _____
 Par _____
 Date _____

Etude : _____
 Dessins : _____
 Date : _____

Telemecanique

Machine automatique de perçage
Traitement Postérieur (POS)

Folio
4 / 6

F

PL7-2		Formulaire programmation schéma à contacts									
N°	ZONE TEST								ZONE ACTION	OBSERVATIONS	
" M.A.R. . . . P.R.F.P.A.R.A.T.L.O.N											
	B4	I0,11							O0,2	()	Marche de préparation
1		I0,15							O0,6	()	pour satisfaire
0		I0,13							O0,4	()	les conditions
P										()	initiales
										()	(a. c. b.)
" M.A.R. . . . P.R.F.P.A.R.A.T.L.O.N											
	B4	I0,17							O0,8	()	d.
1		I0,19							O0,10	()	e.
1									150		
0										(J)	saut vers fin postérieur
P										()	
" M.A.R.C.H.E. M.A.N.U. A. - B											
	I0,9	I0,8	I0,10	I1,10					O0,1	()	C ¹⁰⁰ manuelle sortie vérin A
1			I0,11	I1,11					O0,2	()	C ¹⁰⁰ manuelle recul vérin A
2			I0,12	I1,12					O0,3	()	C ¹⁰⁰ manuelle sortie vérin B
0			I0,13	I1,13					O0,4	()	C ¹⁰⁰ manuelle recul vérin B
P										()	
" M.A.R.C.H.E. M.A.N.U. C. - D											
	I0,9	I0,8	I0,14	I1,14					O0,5	()	C ¹⁰⁰ manuelle sortie vérin C
1			I0,15	I1,15					O0,6	()	C ¹⁰⁰ manuelle recul vérin C
3			I0,16	I1,16					O0,7	()	C ¹⁰⁰ manuelle sortie vérin D
0			I0,17	I1,17					O0,8	()	C ¹⁰⁰ manuelle recul vérin D
P										()	
<input type="checkbox"/> Tâche maître langage à contacts		<input type="checkbox"/> Tâche rapide		<input type="checkbox"/> Traitement préliminaire		<input type="checkbox"/> Traitement postérieur					
		Mise à jour	Par	Date	Etude :	Dessins :	Date :				
A. _____ B. _____ C. _____					Machine automatique de perçage Traitement Postérieur (POS)			Folio 5 / 6			

F

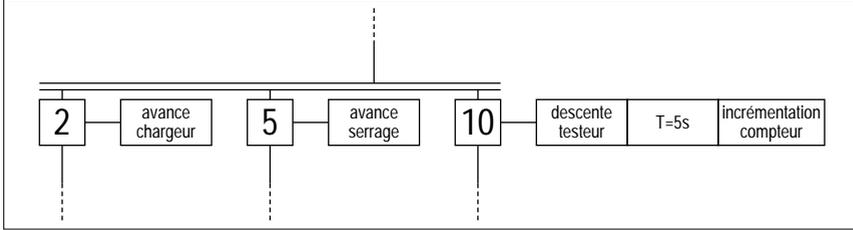
PL7-2	Formulaire programmation schéma à contacts										
N°	ZONE TEST							ZONE ACTION	OBSERVATIONS		
". M.A.R.C.H.E. M.A.N.U. .E. - F											
	10,9	10,8	10,18	11,18					00,9	()	C. manuelle sortie vérin E.
1											
4			10,19	11,19					00,10	()	C. manuelle recul vérin E.
Ø											
P			10,20	11,20					00,11	()	C. manuelle sortie vérin F.
". F.I.N. .P.O.S.T.E.R.I.E.U.R.											
									B255	()	Label fin du
1											
5											traitement postérieur
Ø											
P											
"											
"											
<input type="checkbox"/> Tâche maître langage à contacts			<input type="checkbox"/> Tâche rapide			<input type="checkbox"/> Traitement préliminaire			<input type="checkbox"/> Traitement postérieur		
	Mise à jour	Par	Date	Etude :	Dessins :	Date :					
	A. _____	_____	_____	Machine automatique de perçage Traitement Postérieur (POS)							Folio
	B. _____	_____	_____								6 /
C. _____	_____	_____	6								

F

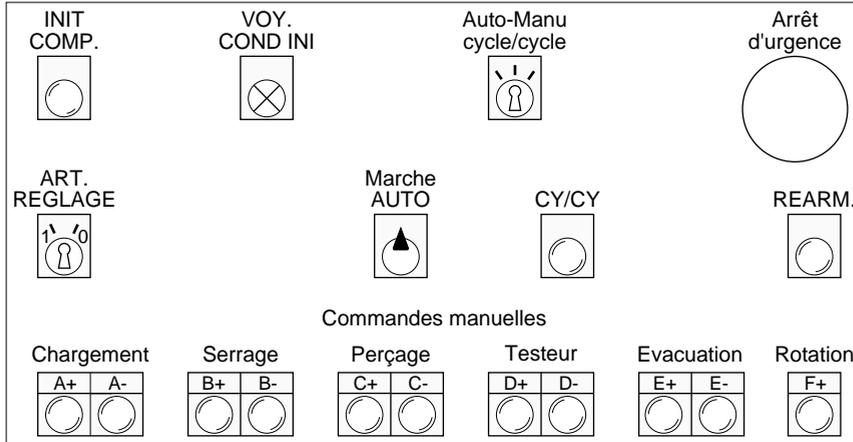
3.4 Comptage des pièces (troisième partie)

Un compteur permet de compter le nombre de pièces contrôlées. La valeur courante du compteur (C0) peut être initialisée par action sur un bouton-poussoir "bpiniC0" (I1,0).

Modification du GRAFCET



Pupitre de commande



F

3.4-1 Programme

Les entrées R et U du compteur C0 fonctionnent sur front :

- un front montant sur l'entrée I1,0 réinitialise C0,V
- l'activation de l'étape 10 incrémente le compteur.

PL7-2		Formulaire programmation schéma à contacts													
N°		ZONE TEST										ZONE ACTION		OBSERVATIONS	
".C.O.M.P.T.A.G.E. .P.I.E.C.E.S															
		I1,0 C0 B254													
4		R E										()		Insertion label 40	
0												-		dans traitement	
P		X10										-		Postérieur (POS)	
		B40										-		entre label 30 et	
		U										-		label 50	
".M.A.R. .P.R.E.P.A.R.A.T.I.O.N															
		B4. I0,17										O0,8			
1												()		Modification	
1		I0,19										O0,10		label 110	
0												-		Traitement postérieur	
P		I1,0										B40			
												-		mémoire de	
												-		l'information I1,0	
												-		dans B40	
												-		(J)	

3.5 Communication par bloc texte (quatrième partie)

3.5-1 Affichage défauts sur imprimante.

Les caractéristiques des pièces déclarées mauvaises au contrôle sont affichées sur une imprimante, raccordée à la prise terminal de l'automate.

Message émis : "**PIECE NUMERO :CRLF**" (émission de 20 caractères).

CR : caractère de contrôle "retour en début de ligne" (Carriage Return), code ASCII: H'0D'.

LF : caractère de contrôle "saut de ligne" (Line Feed), code ASCII : H'0A'.

La partie constante du message doit être mémorisée dans une table de mots constants :

Tableau des codes ASCII des caractères : H'..' à saisir en mode CONFIGURATION.

CW0	I	P	49	50
CW1	C	E	43	45
CW2		E	20	45
CW3	U	N	55	4E
CW4	E	M	45	4D
CW5	O	R	4F	52
CW6		:	20	3A

Les valeurs numériques⁽¹⁾ codées en binaire sur 16 bits doivent être obligatoirement transcodées en ASCII (opération: BTA), avant d'être émise par un bloc texte.

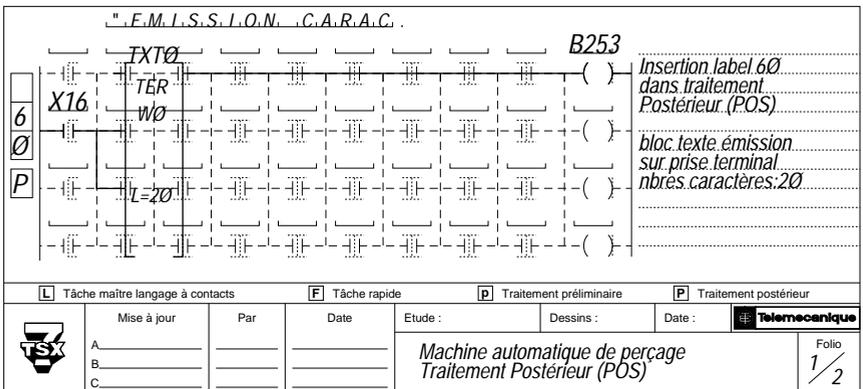
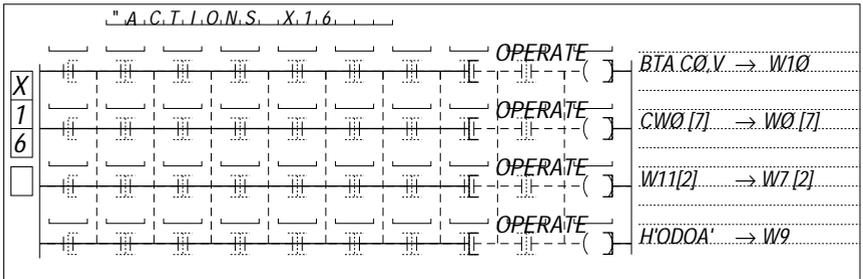
Le contenu de C0,V devra donc être transcodé en ASCII. Le résultat sera placé dans trois mots successifs.

Exemple : BTA C0,V → W10 si C0,V contient la valeur 978, après transcodage, les mots W10, W11 et W12 contiendront le résultat en hexadécimal.

W10	W11	W12	Les chiffres sigificatifs sont donc placés dans W11 et W12. En effet, C0,V ≤ 9999
00 00	00 39	37 38	

Programme

Ce réseau de contacts est associé à l'étape 16. Il est saisi dans la zone de traitement séquentiel.



⁽¹⁾ caractères alphanumériques en général.

3.5-2 Datation des défauts (avec TSX 17-20 muni de l'option horodateur)

La fonction "DT" (voir intercalaire B : Langages à contacts, chapitre 7 : Bloc fonction Horodateur), sera utilisée pour horodater les messages concernant les pièces mauvaises.

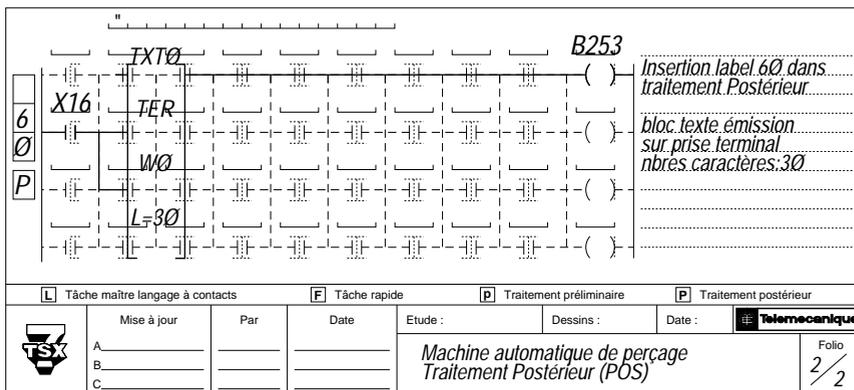
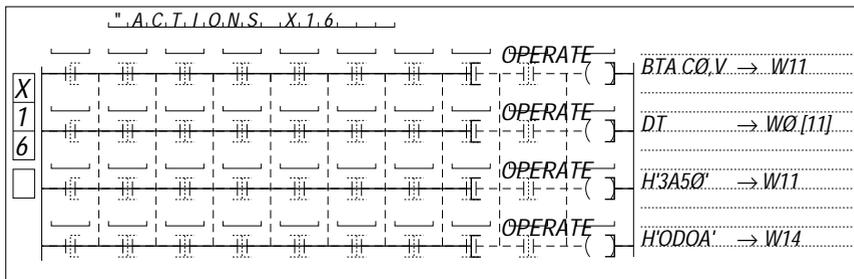
Le message à afficher devient le suivant :

"SSAA-MM-JJ-HH:MN:SS-DNP:....CRLF" (30 caractères au total)
 < DT : 22 caractères > _____ voir page 3/22
 _____ valeur de C0,V

Programme

Le programme ci-dessous doit se substituer à celui de la page précédente.

Ce réseau de contacts est associé à l'étape 16. Il est saisi dans la zone de traitement séquentiel.



F



Sous chapitre	Page
4.1 Temps d'exécution mémoire	4/2
4.2 Définitions GEMMA	4/4
4.3 Code ASCII	4/5
Ce chapitre se termine à la page	4/6

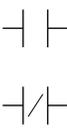
4.1 Temps d'exécution mémoire

Système

Elément		Temps en μs		
		TSX17-20	TSX 27	TSX 47
Gestion du terminal		100	100	100
Gestion automate		700 à 830 ⁽¹⁾	2100	2100
Module	4 entrées/4sorties	-	-	340
	8 entrées	-	-	260
	12 entrées	-	400	-
	16 entrées	-	-	400
	32 entrées	-	-	800
	8 sorties	-	280	220
	16 sorties	-	-	320
	32 sorties	-	-	640
Acquisition des entrées		148 à 490 ⁽¹⁾	-	-
Rafraîchissement des sorties		64 à 250 ⁽¹⁾	-	-

⁽¹⁾ selon configuration

Zone test

Elément graphique	Objet (ou type)	Temps en μs		Nombre d'octets
		TSX 17-20	TSX 27/47	
	I/Oxy,i (bac 0)	-	2	2
	I/Oxy,i (bac 1)	-	10	12
	Ix,i	2	-	2
	Ox,i	4	-	5
	B0 à B127; Xi; Ti,D(ex); SYi	4	4,5	5
	B128 à B255; I/Oxy,S; TXTi,D(E)	10	10	12
	Bit extrait de mots			
	Temporisateur	44 à 78	60	10
	Monostable	36 à 68	55	10
	Compteur	46 à 93	75	10
	Compteur rapide	76 à 132	-	10
	Registre	72 à 141	100	16
	Programmeur cyclique	69 à 116 ⁽¹⁾	75 ⁽¹⁾	14
	Horodateur	53 à 247	-	11
Bloc texte	Simplifié	-	50 ⁽²⁾	7
	Complet	65 ⁽³⁾	65 ⁽³⁾	7
	Comparaison	129 à 143	150	14
	Liaison verticale	10	10	8

⁽¹⁾ 400 + 40 x (nb bits d'ordre) si changement d'état,

⁽²⁾ 220 + 8 x (nb de blocs texte actifs) si bloc texte actif,

⁽³⁾ 270 + 8 x (nb de blocs texte actifs) si bloc texte actif

Zone action

Elément graphique	Objet (ou type)	Temps en μ s		Nombre d'octets
		TSX 17-20	TSX 27/47	
-()-	Oxy,i(bac 0)	-	10	5
	Oxy,i(bac 1)	-	20	16
	Ox,i	12	-	9
	B0 à B127	12	12	9
	B128 à B255	21	21	16
	Xi ou SYi	3	3	6
	Bits extraits de mots	12	12	13
-(/)- -(S)- -(R)- idem ci-dessus		+2	+2	+2
-(JUMP)-		24	24	16
-[OPER]-	Logique et arithmétique (+, -, /, x)	-	200	19
	Logique et arithmétique (+, -)	163	-	19
	Multiplication (16 bits)	207 à 227	-	19
	Division (16 bits)	196 à 733	-	19
	Complément CPL	170	170	17
	Conversion BCD → Binaire	285	285	17
	Conversion Binaire → BCD	400	400	17
	Conversion Binaire ↔ ASCII	175 + (70 x nb chiffres)	120 + (70 x nb chiffres)	10
	Décalage SRCn/SLCn	175 + (13 x n)	175 + (13 x n)	19
	Transfert :			
	mot → mot	145	145	14
	valeur → mot	140	140	14
	mot → mot indexé	195	195	16
16 bits → mot	500	500	23	
n bits → n bits	150 + (20 x n)	150 + (20 x n)	19	
n mots → n mots	150 + (60 x n)	150 + (60 x n)	13	
Supplément par réseau		16	16	22 + nc (*)

(*) nc = nombre de caractères pour commentaires

Nota : Si les conditions d'exécution des blocs comparaison arithmétique et logique ne sont pas remplies (ligne ladder non passante) leur exécution est réduite à 2 micro-secondes.

4.2 Définitions GEMMA

Marche de production normale	<p>C'est le fonctionnement normal. C'est l'état pour lequel l'automatisme a été conçu. A ce fonctionnement a été associé le Grafcet de base.</p>
Marche de préparation	<p>C'est une marche préparatoire à la marche de production normale. C'est une marche transitoire.</p>
Marche de clôture	<p>Comme la marche de préparation, c'est une marche transitoire qui peut être nécessaire en fin de campagne, de série, de journée dans le but de "vider" certaines machines, ou sous-ensembles.</p>
Marche dégradée	<p>Production tout de même. La production après une défaillance de la machine, est assurée soit par le forçage de certaines informations, soit par l'intervention des opérateurs.</p>
Arrêt normal	<p>De différents types, il n'est pas la conséquence d'une défaillance, l'arrêt peut s'effectuer :</p> <ul style="list-style-type: none">• dans l'état initial,• en fin de cycle,• dans un état déterminé.
Arrêt sur défaut	<p>C'est un arrêt subi, suite à une défaillance. L'arrêt très souvent est effectif après des cycles de dégagement ou après des procédures limitant les conséquences dues à la défaillance.</p>
Vérification dans l'ordre	<p>Le cycle de production est analysé par l'opérateur sous-ensemble par sous-ensemble ou séquence par séquence.</p>
Vérification dans le désordre	<p>Le contrôle de certaines fonctions ou de certains mouvements est effectué sans respecter l'ordre du cycle.</p>

4.3 Code ASCII

				<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>b7</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>b6</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>b5</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>b4</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> </table>								b7	0	0	0	0	0	0	0	0	b6	0	0	0	0	1	1	1	1	b5	0	0	1	1	0	0	1	1	b4	0	1	0	1	0	1	0	1
b7	0	0	0	0	0	0	0	0																																							
b6	0	0	0	0	1	1	1	1																																							
b5	0	0	1	1	0	0	1	1																																							
b4	0	1	0	1	0	1	0	1																																							
				<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>Colonne</td><td>→</td></tr> <tr><td>Ligne</td><td></td></tr> </table>		Colonne	→	Ligne		<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> </table>								0	1	2	3	4	5	6	7																						
Colonne	→																																														
Ligne																																															
0	1	2	3	4	5	6	7																																								
b3	b2	b1	b0																																												
0	0	0	0	0	NUL	DLE	SP	0	@	P	\	p																																			
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q																																			
0	0	1	0	2	STX	DC2	"	2	B	R	b	r																																			
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s																																			
0	1	0	0	4	EOT	DC4	\$	4	D	T	d	t																																			
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u																																			
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v																																			
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w																																			
1	0	0	0	8	BS	CAN	(8	H	X	h	x																																			
1	0	0	1	9	HT	EM)	9	I	Y	i	y																																			
1	0	1	0	A	LF	SUB	*	:	J	Z	j	z																																			
1	0	1	1	B	VT	ESC	+	;	K	[k	{																																			
1	1	0	0	C	FF	FS	,	<	L	\	l																																				
1	1	0	1	D	CR	GS	-	=	M]	m	}																																			
1	1	1	0	E	SO	RS	.	>	N	^	n	~																																			
1	1	1	1	F	SI	US	/	?	O	_	o	DEL																																			

F



LANGAGES PL7-2. Généralités

TRANSFERT TSX T407 <---> MAGNETOPHONE OU AUTRE PERIPHERIQUE

Généralités

La prise périphérique du terminal T407 V3.. permet l'enregistrement des programmes sur un magnétophone ou tout autre système d'archivage (micro-ordinateur par exemple) pourvu d'une liaison V24.

Afin d'offrir une sécurité au niveau de la cohérence du programme transmis, tout enregistrement est contrôlé par checksum, ce qui implique une attente d'une seconde environ après chaque enregistrement.

L'enregistrement d'un programme est structuré de la manière suivante :

- enregistrement de l'en-tête
- attente de 8 secondes (attente effacement mémoire de l'automate)
- enregistrement des données par paquets de 24 données (48 octets) maximum avec contrôle de checksum (1) et attente de 1 seconde après chaque paquet (2).
- enregistrement de fin

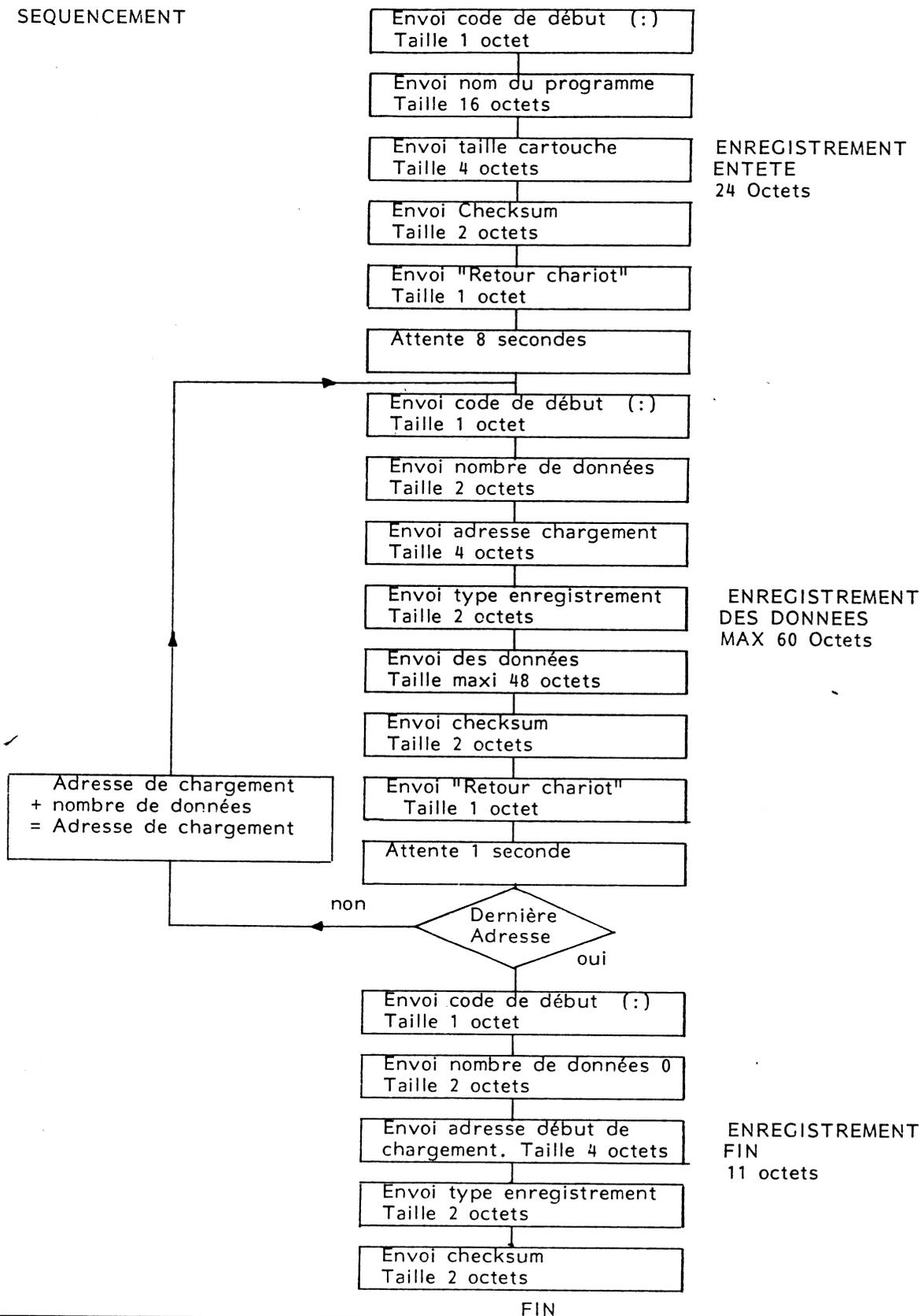
- (1) L'enregistrement des données est effectué au standard INTEL
- (2) Le transfert TR ---> PC ou INT peut être effectué en gérant le signal CTS afin d'optimiser le temps d'attente de 1 seconde entre 2 paquets.



LANGAGES PL7-2. Généralités

TRANSFERT TSX T407 <---> MAGNETOPHONE OU AUTRE PERIPHERIQUE

SEQUENCEMENT



LANGAGES PL7-2. Généralités

TRANSFERT TSX T407 <---> MAGNETOPHONE OU AUTRE PERIPHERIQUE

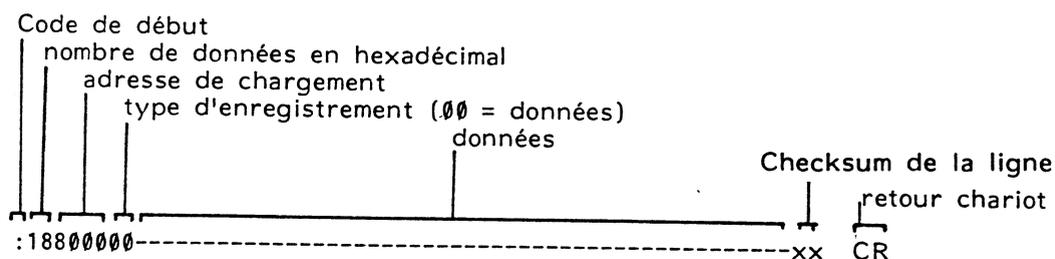
EXEMPLE

Entete



ATTENTE 8 SECONDES

Données



ATTENTE 1 SECONDE

:18801800-----xx CR

ATTENTE 1 SECONDE

:18803000-----xx CR

ATTENTE 1 SECONDE

:18804800-----xx CR

ATTENTE 1 SECONDE

:18806000-----xx CR

ATTENTE 1 SECONDE

:189FC800-----xx CR

ATTENTE 1 SECONDE

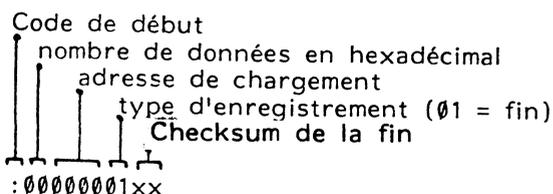
:189FE000-----xx CR

ATTENTE 1 SECONDE

:089FF800-----xx CR

ATTENTE 1 SECONDE

Fin



Sur la cartouche VL3.3 un CR sera envoyé après le checksum de fin

Cartouches PL7-2 TSX17 : Sauvegarde mots W en EEPROM

(Nota cette fonctionnalité est aussi dite : "FONCTION POTAIN")

Avertissement :

Ce service implémenté pour résoudre une application particulière, permet de s'affranchir des cas (hors caractéristiques) où la pile lithium ne jouerait pas son rôle (très basse température de stockage, condensation ...).

Les composants EEPROM perdant leur qualité au fur et à mesure des cycles d'écriture, nous ont poussé à introduire un compteur logiciel pour limiter à 10 000 le nombre de sauvegardes sur une même cartouche EEPROM. En conséquence : cette fonctionnalité ne doit pas être un argument de vente, et doit être utilisée que lorsque l'exception la justifie.

Description de la «FONCTION POTAIN»

Cette fonction permet la sauvegarde / restitution de la valeur des mots W0 à W127 entre la RAM et l'EEPROM. Cette sauvegarde / restitution est possible dès que l'automate est équipé d'une cartouche de sauvegarde EEPROM de 24Ko (TSXMC70E324).

La sauvegarde / restitution utilise une zone libre de cette cartouche :

- sans influencer la fonction de sauvegarde du programme application,
- indépendamment de la position de «travail / back-up» de l'interrupteur (switch) de la cartouche,
- indépendamment de l'état, protégé ou non, de l'application.

NOTA :

- La validité de la sauvegarde des W0..W127 est assurée par un contrôle de checksum et un pattern.
- Le nombre de cycles d'écriture de l'EEPROM est limité à 10 000, par un comptage de ces derniers, et est sauvegardé dans cette mémoire (Limitation due à la technologie des composants EEPROM).

Utilisation :

SW44 sert à la commande et au contrôle des transferts entre les mots W0... W127 et la zone de sauvegarde de la EEPROM. Il est remis à zéro par le système lors d'un redémarrage à froid. Les bits 0 ... 3 sont écrits par l'application pour commander les transferts :

- SW44,0 : =1, Demande de transfert (RAZ par le système après prise en compte)
- SW44,1 : Sens du transfert de données demandé
=0, Restitution (RAM ← EEPROM)
=1, Sauvegarde (RAM → EEPROM)
- SW44,2 et SW44,3 inutilisés

Les bits 4 ... 7 sont écrits par le système pour rendre compte à l'application du déroulement du transfert :

- SW44,4 : =1, Demande prise en compte, validation du compte-rendu (bits de 5 ... 7)
- SW44,5 : =0, Transfert correctement effectué
=1, Défaut
- SW44,6 : =1, Sauvegarde : écriture incorrecte (mauvaise cartouche, pas de cartouche, ou compteur saturé)
=1, Restitution : lecture incorrecte (mauvaise cartouche, pas de cartouche ou cartouche n'ayant jamais été écrite)
- SW44,7 : =1, nombre d'écritures permises (10.000) atteint

SW45 Contient la valeur du compteur d'écritures : nombre cumulé des transferts en écriture (sauvegarde) pour la cartouche considérée. Ce compteur est géré automatiquement par le système. Pour le rendre accessible à l'application ainsi qu'à tout terminal de réglage, il est remis à jour à chaque transfert (sauvegarde ou restitution).