

AUTOMATES TSX MICRO & PREMIUM



LOGICIEL PL7 PRO

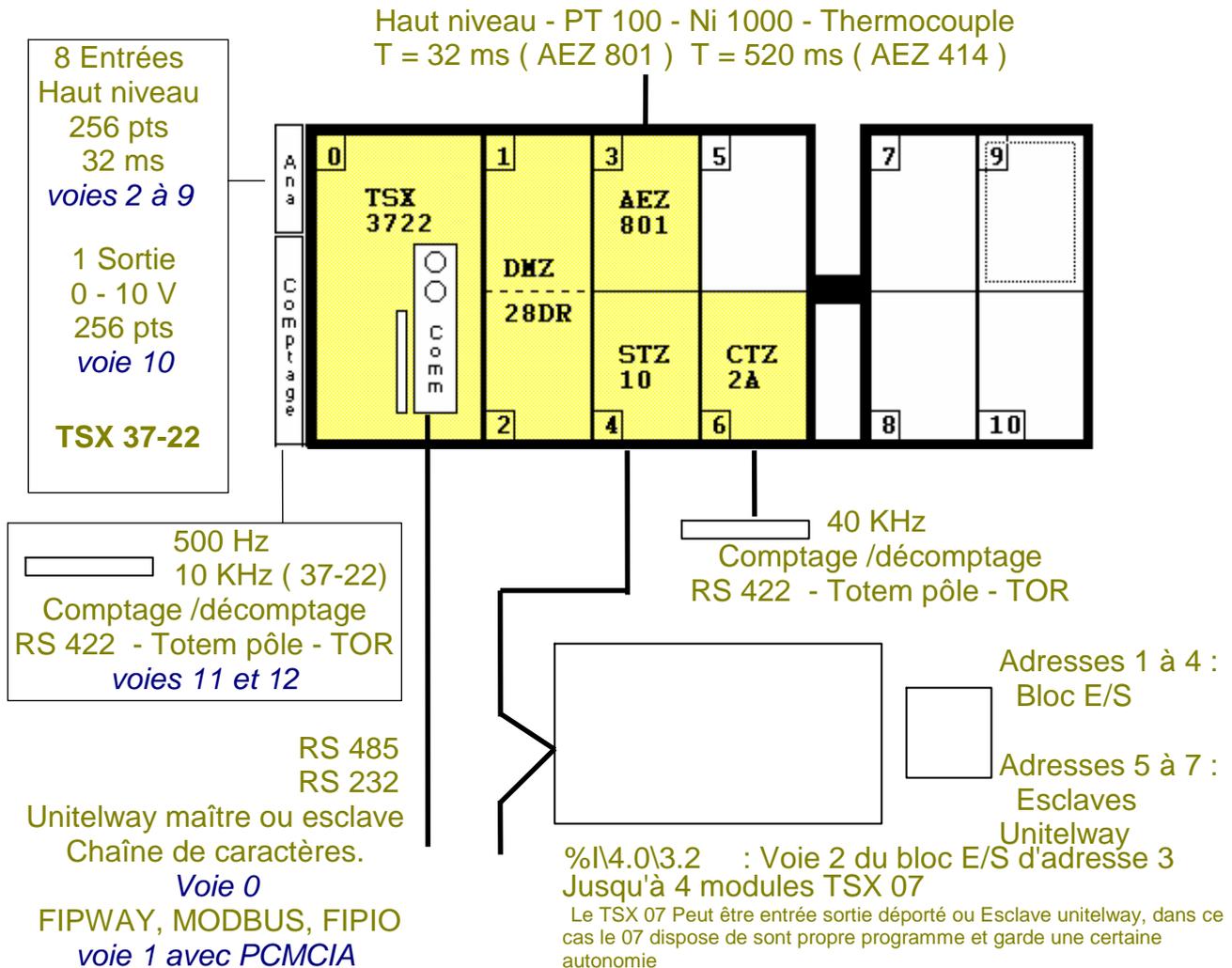
SOMMAIRE

TSX MICRO VUE D'ENSEMBLE	5
FONTION INTEGREES	5
LANGAGES	5
PRESENTATION DU LOGICIEL PL7 PRO	6
CREATION D'UNE NOUVELLE APPLICATION	6
PRESENTATION DU NAVIGATEUR	6
<i>Nom de la station</i>	7
<i>Dossier configuration</i>	7
<i>Dossier Programme</i>	7
CONFIGURATION	8
CONFIGURATION DES CARTES	8
<i>Configuration d'un coupleur analogique d'entrée</i>	9
<i>Configuration d'une voie MODBUS (Liaison intégrée)</i>	9
ADRESSAGES DES MODULES	10
TSX MICRO	10
TSX PREMIUM	10
STRUCTURE DE LA MEMOIRE	11
STRUCTURE MULTITÂCHE	12
SCRUTATION D'UNE TACHE	12
STRUCTURE D'UNE TACHE	12
STRUCTURE D'UNE TACHE	13
CONFIGURATION DE L'APPLICATION	14
CONFIGURATION PROCESSEUR	14
CONFIGURATION DE TACHE EVENEMENTIELLE	14
ADRESSAGE DES OBJETS BITS ET MOTS	15
PRESENTATION GENERALE	15
<i>Qu'est qu'un objet</i>	15
<i>Représentation des objets</i>	15
<i>Liste des différents types d'objets</i>	15
<i>Les formats des objets</i>	15
LES ELEMENTS DES OBJETS	16
LES OBJETS STRUCTURES	16
<i>Bit extrait de mots</i>	16
<i>Tableaux de bits ou chaines de bits</i>	16
<i>Tableau de mots</i>	16
LES FORMES D'ADRESSAGES	17
LES FORMES D'ADRESSAGES	18
<i>L'adressage direct</i>	18
<i>L'adressage indexé</i>	18
ORGANISATION MEMOIRE OBJETS	20
LE RANGEMENT DES MOTS EN MEMOIRE	20
REPRESENTAION DES VALEURS IMMEDIATES	21
<i>Entier décimal</i>	21
<i>Hexadécimal</i>	21
<i>Binaire</i>	21
<i>Chaîne de caractère</i>	21
<i>Les dates</i>	21
<i>Les heures</i>	21
<i>Les durées</i>	21
<i>Les flottants</i>	21
CREATION D'UNE SECTION	23

LANGAGE A CONTACTS	24
INSTRUCTION DE TEST D'OBJETS BITS	24
INSTRUCTION DE MISE A JOUR DE BITS	24
INSTRUCTIONS DE BRANCHEMENT	25
INSTRUCTIONS DE COMPARAISONS	25
OPERATION	25
FONCTIONS GRAPHIQUES	26
L'EDITEUR DE VARIABLES	28
LES BLOCS FONCTION PREDEFINIS	29
LES TEMPORISATEURS	29
<i>Principe général</i>	29
<i>Temporisateur %TM</i>	30
<i>Temporisateur SERIE 7 %T</i>	31
<i>Monostable</i>	32
COMPTEURS	33
<i>Principe et objets du compteur</i>	33
REGISTRES	34
PROGRAMMATEUR CYCLIQUE DRUM	35
BIBLIOTHEQUE DE FONCTIONS	37
PRESENTATION	37
FAMILLE DE FONCTION EF	38
<i>Conversions numériques</i>	38
<i>Chaînes de caractères</i>	38
<i>Dates et durées</i>	38
<i>Temporisation</i>	38
<i>Communication</i>	38
<i>Opérations sur entier double et simple longueur</i>	38
<i>Gestion des CCX 17</i>	38
<i>Gestion des événements</i>	39
<i>Fonctions de régulation</i>	39
<i>Fonctions d'échanges explicites</i>	39
<i>Fonctions sur tableaux de mots, bits, mots doubles</i>	39
LES FONCTIONS DBF	39
LES FONCTIONS EF TEMPORISATEURS	40
PRINCIPE DE FONCTIONNEMENT	40
RETARD A L'ENCLenchement FTON	41
<i>Syntaxe</i>	41
<i>Description</i>	41
<i>Chronogramme de fonctionnement</i>	41
<i>Exemple</i>	41
RETARD AU DECLenchement FTOF	42
<i>Syntaxe</i>	42
<i>Description</i>	42
<i>Chronogramme de fonctionnement</i>	42
<i>Exemple</i>	42
IMPULSION FTP	43
<i>Syntaxe</i>	43
<i>Description</i>	43
<i>Chronogramme de fonctionnement</i>	43
<i>Exemple</i>	43
GENERATEUR DE SIGNAL RECTANGULAIRE FPULSOR	44
<i>Syntaxe</i>	44
<i>Description</i>	44
<i>Chronogramme de fonctionnement</i>	44
<i>Exemple</i>	44
LANGAGE LISTE D'INSTRUCTION IL	46

PRESENTATION	46
LES INSTRUCTIONS	46
<i>Instructions booléennes</i>	46
<i>Valeur « Vrai » et « Faux »</i>	47
<i>Instruction sur blocs fonctions FB prédéfinis</i>	47
<i>Opération sur mots et comparaisons</i>	47
LANGAGE LITTERAL STRUCTURE	49
STRUCTURE DE PROGRAMME	49
<i>Structure inconditionnelle</i>	49
<i>Structures conditionnelle</i>	49
<i>Structures itératives</i>	49
<i>Structure répétitive</i>	49
<i>Le mot clé "EXIT"</i>	50
EXEMPLES DE PROGRAMME EN LITTERAL STRUCTURE	50
LE LANGAGE GRAFCET	52
OBJETS ASSOCIES AUX ENTREES SORTIES	55
RAPPEL SUR LA SCRUTATION D'UNE TACHE	55
OBJET A ECHANGE IMPLICITE	56
OBJET A ECHANGE EXPLICITE	56
OBJETS ASSOCIES AUX MODULES	57
VOIES ANALOGIQUES	58
<i>Exemple</i>	58
FONCTION DE COMPTAGE INTEGREE	59
CONFIGURATION	59
EXEMPLE DES COMPTEURS INTEGRES AUX UC TSX MICRO	60
COMPTAGE SUR MODULE TOR	61
COMPTAGE SUR CARTE CTY	62
COMMUNICATIONS	63
FONCTIONS DE COMMUNICATION	65
COMPTES RENDUS DE COMMUNICATION	66
AIDE A LA SAISIE D'UNE FONCTION DE COMMUNICATION	67
SAISIE DE L'ADRESSE	68
DIAGNOSTIQUE D'UNE LIAISON	69
MISE EN LIGNE AVEC UNE STATION	70
DIALOGUE OPERATEUR CCX 17	71
FONCTIONS DOP	72
PARAMETRES DES FONCTIONS DOP	74
AIDE A LA SAISIE DES FONCTIONS DOP	76
ERGONOMIE	77

TSX MICRO VUE D'ENSEMBLE



Les voies 0 à 4 du module 1 peuvent être configurées en comptage décomptage 500 Hz

Cartes PCMCIA pour communication MODBUS/JBUS, UNITELWAY, FIPWAY, Chaîne de caractères , FIPIO Agent.

FONCTION INTEGRESSES

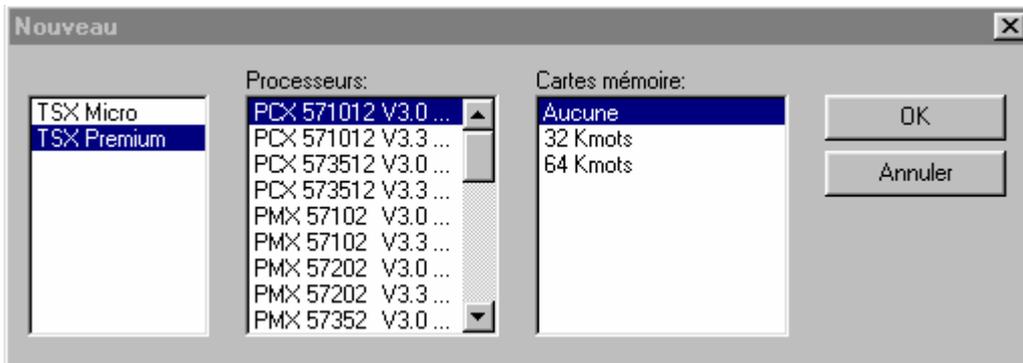
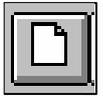
- ◆ Commande de mouvements
- ◆ Communication
- ◆ Régulation (PID et PWM)
- ◆ Dialogue opérateur (CCX 17)
- ◆ Affichage ...

LANGAGES

- ◆ Ladder
- ◆ Liste d'instructions
- ◆ Grafset et Macro étapes
- ◆ Littéral structuré

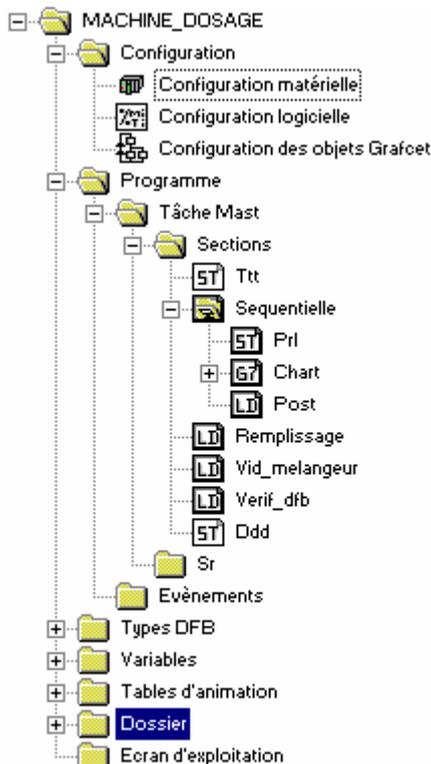
Présentation du logiciel PL7 Pro

CREATION D'UNE NOUVELLE APPLICATION

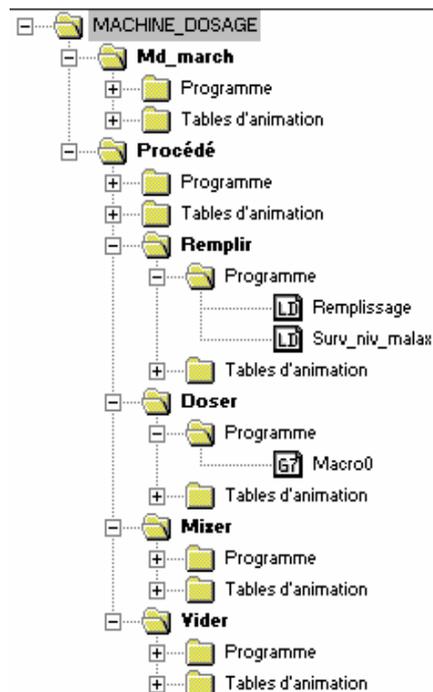


PRESENTATION DU NAVIGATEUR

VUE STRUCTURELLE



VUE FONCTIONNELLE (TSX 57)



NOM DE LA STATION

En sélectionnant la station à l'aide de la souris, puis en choisissant le menu Edition, option Propriété, on accède aux propriétés de la station. On pourra à partir de cet Ecran :

- ◆ Modifier le nom de la Station
- ◆ Gérer les numéros de versions du programme, de façon automatique ou manuelle
- ◆ Visualiser la dernière date de modification de l'application
- ◆ Visualiser la signature de l'application
- ◆ Protéger l'application

Attention !

Protéger une application implique qu'après sont transfert sur l'automate, l'application n'est plus modifiable, il faudra donc en transférer une nouvelle !

DOSSIER CONFIGURATION

Le dossier Configuration permet, en double cliquant dessus, d'avoir accès à la configuration:

- ◆ Matérielle pour accéder aux racks et modules
 - Configuration des racks
 - Configurations des modules
 - Eventuellement, changement du type de Microprocesseur
- ◆ Logicielle pour définir les paramètres de l'application
 - le nombre des différents types de blocs fonctions
 - le nombre de mots des registres,
 - le nombre de bits internes %M,
 - le nombre de mots internes %MW,
 - le nombre de constantes %KW.
- ◆ Grafcet pour définir les paramètres de l'application spécifiques au Grafcet.
 - le nombre d'étapes,
 - le nombre d'étapes actives,
 - le nombre de transitions valides.

DOSSIER PROGRAMME

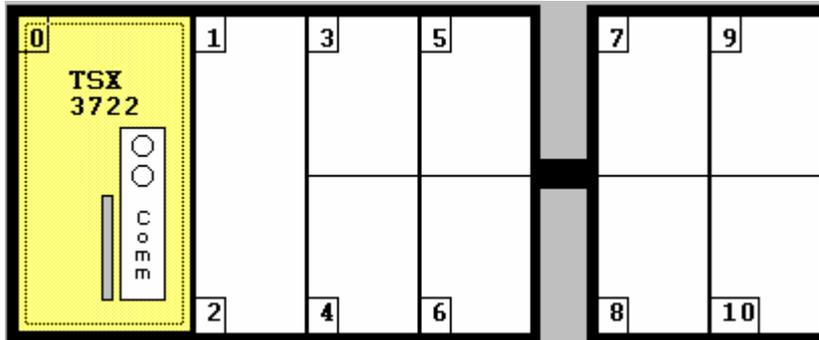
Ce dossier contient les différentes sections de programme, en LADDER, en LITTERAL, en LISTE, en GRAFCET.

Un programme est structuré en tâches (tâche maître, tâche rapide) elles mêmes découpée en sections, et d'événements déclenché par des cartes périphériques.

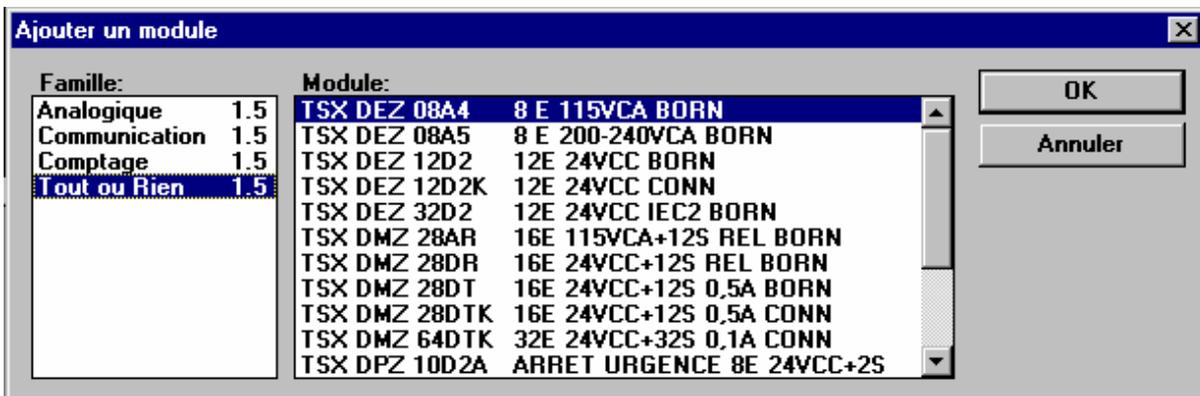
On peut visualiser les différentes sections d'un programme sous forme fonctionnelle.

CONFIGURATION

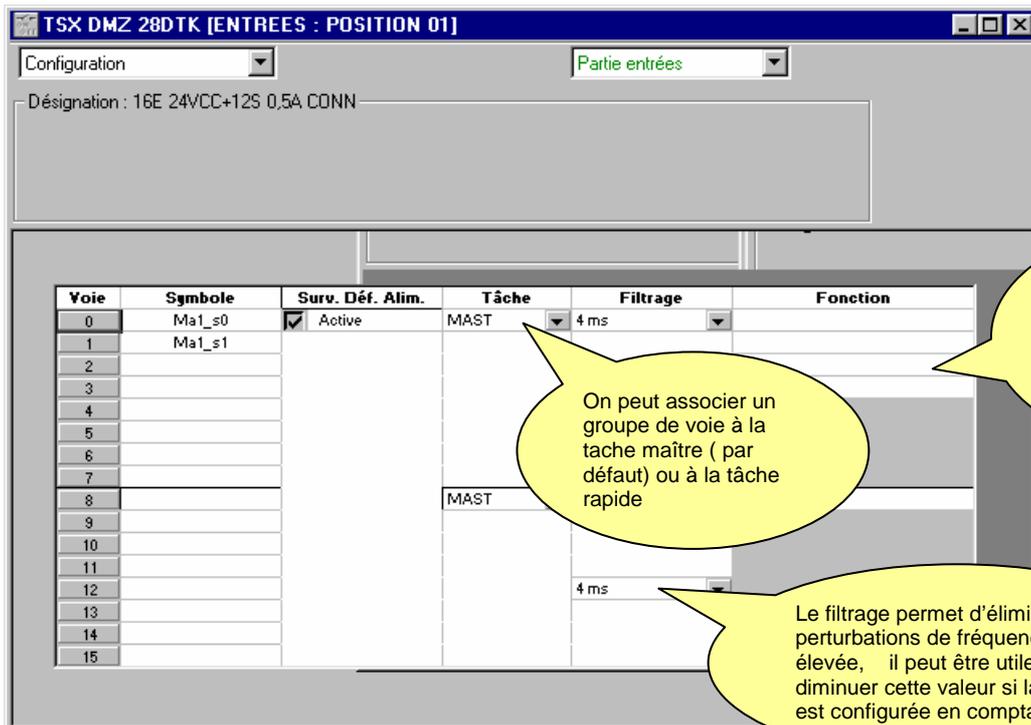
Double cliquer sur le dossier Configuration matérielle dans le navigateur



Double cliquez sur le module à configurer, puis choisissez dans la liste



CONFIGURATION DES CARTES



Double cliquez sur le module pour le configurer

On peut associer un groupe de voie à la tâche maître (par défaut) ou à la tâche rapide

Les 4 premières voies d'entrées sur un TSX 37 peuvent être configurées comme voies de comptage rapide ou événementielle ou voie

Le filtrage permet d'éliminer les perturbations de fréquence élevée, il peut être utile de diminuer cette valeur si la voie est configurée en comptage

CONFIGURATION D'UN COUPLEUR ANALOGIQUE D'ENTREE

TSX AEZ 801 [POSITION 03]

Configuration

Désignation : 8 ENTREES ANA. +/-10V

Cycle :

Normal

Rapide

Voie	Utilisée	Tâche	Symbole	Gamme	Echelle	Filtre
0	<input checked="" type="checkbox"/>	MAST		+/-10V	%..	0
1	<input checked="" type="checkbox"/>			+/-10V	%..	0
2	<input checked="" type="checkbox"/>			+/-10V	%..	0
3	<input checked="" type="checkbox"/>			+/-10V	%..	0
4	<input checked="" type="checkbox"/>			+/-10V	%..	0
5	<input checked="" type="checkbox"/>			+/-10V	%..	0
6	<input checked="" type="checkbox"/>			+/-10V	%..	0
7	<input checked="" type="checkbox"/>			+/-10V	%..	0

CONFIGURATION D'UNE VOIE MODBUS (LIAISON INTEGREE)

TSX 57252 [RACK 0 POSITION 1]

Configuration

Désignation : PROCESSEUR TSX P 57252

VOIE 1 :

VOIE 1 : TSX SCP 112 CARTE PCMCIA BC MP

LIAISON MODBUS/JBUS : MAST

Type : Maître

Vitesse de Transmission : 9600 Bits/s

Delai inter-Characteres : Par défaut 4 ms

Maître

Nombre de répétitions : 3

Délai de réponse : 100 X 10 ms

Données : ASCII (7 bits) RTU (8 bits)

Stop : 1 bit 2 bits

Esclave

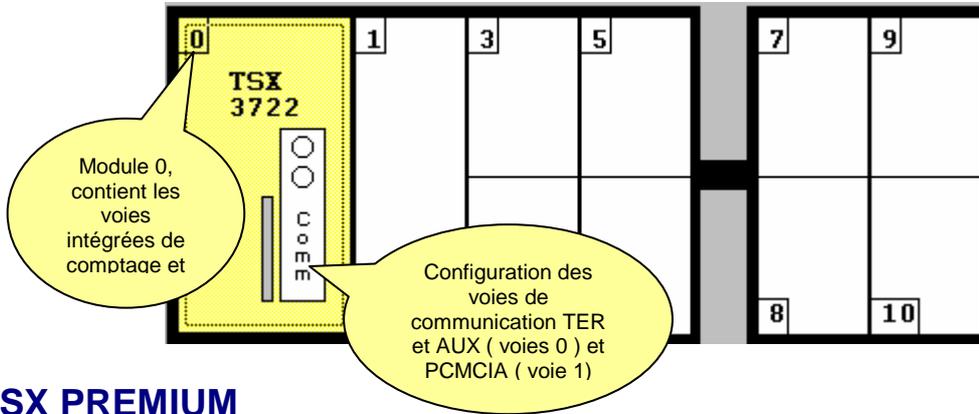
Numéro d'esclave : 1

Parité :

Boucle de Courant (PSR) : Multipoint Point à Point

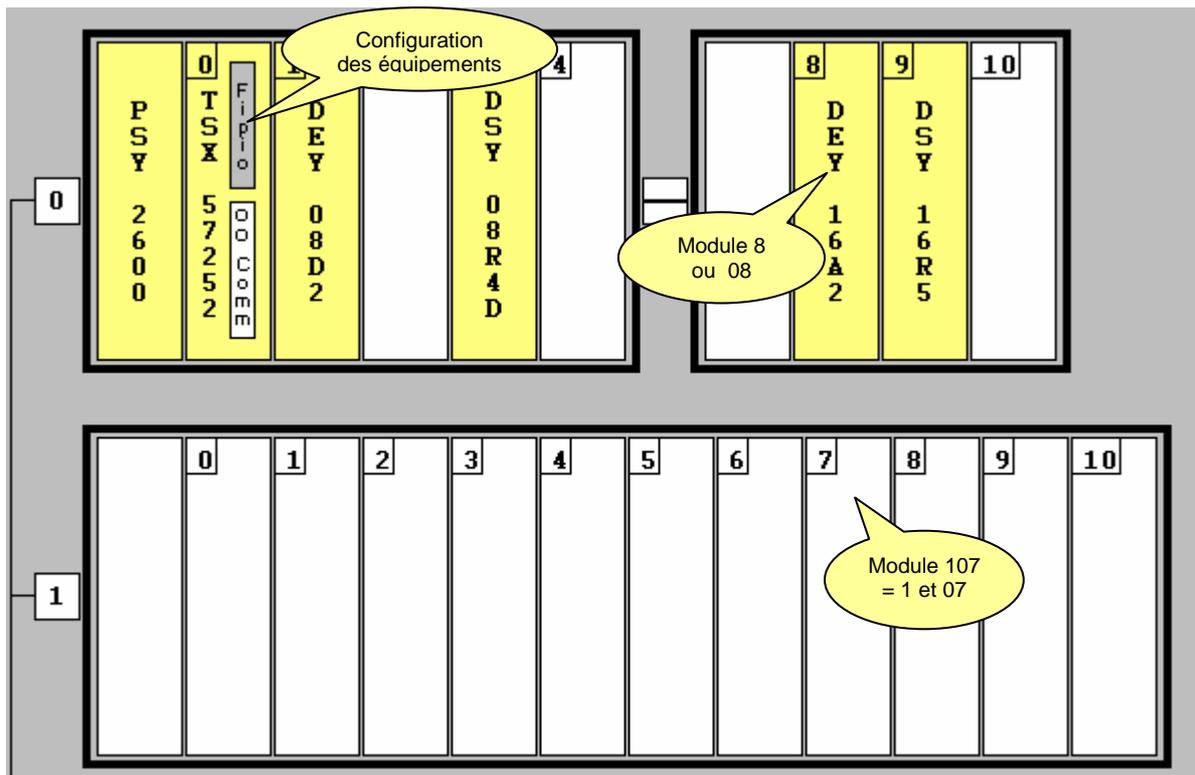
ADRESSAGES DES MODULES

TSX MICRO



TSX PREMIUM

Jusqu'à 7 racks d'extentions 1 à 7

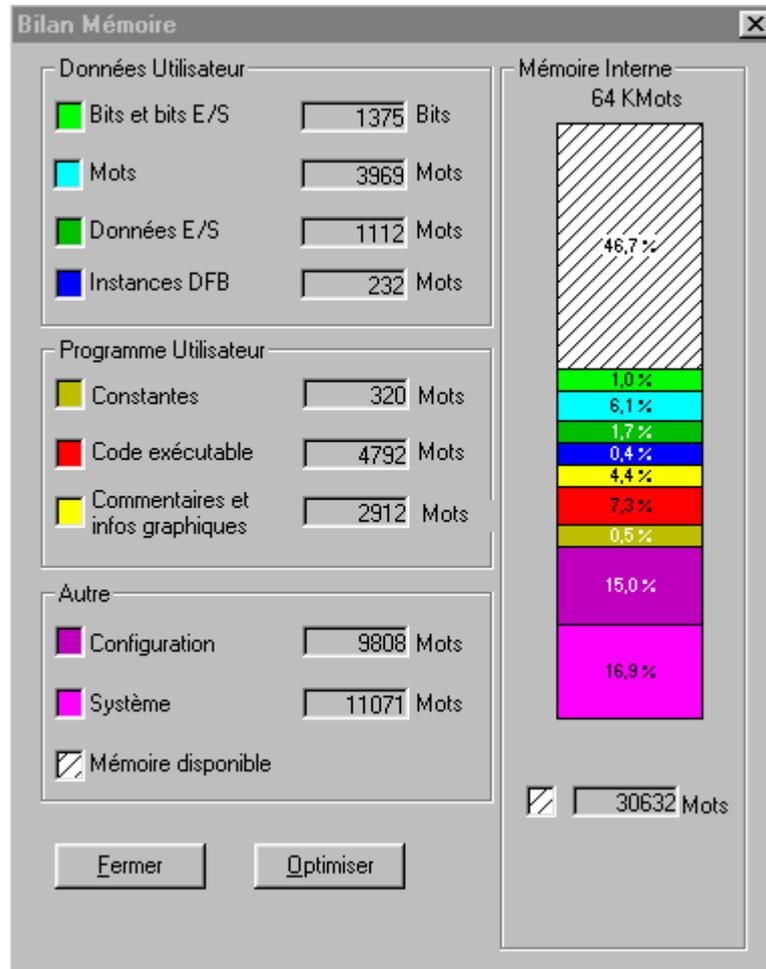


Le rack du TSX Micro porte par défaut le numéro 0

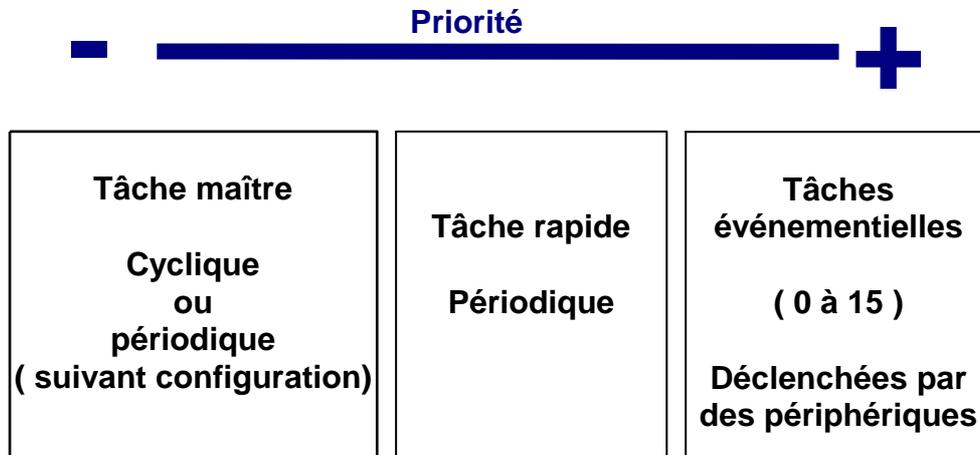
N°RackN°Module : N°rack sur 1 chiffre (peut être omis si rack 0)
N°Module sur 2 chiffres

EXEMPLES : **105** Module **05** du rack **1**
4 Module **4** du rack **0** (les 0 devant peuvent être omis)
410 Module **10** du rack **4**

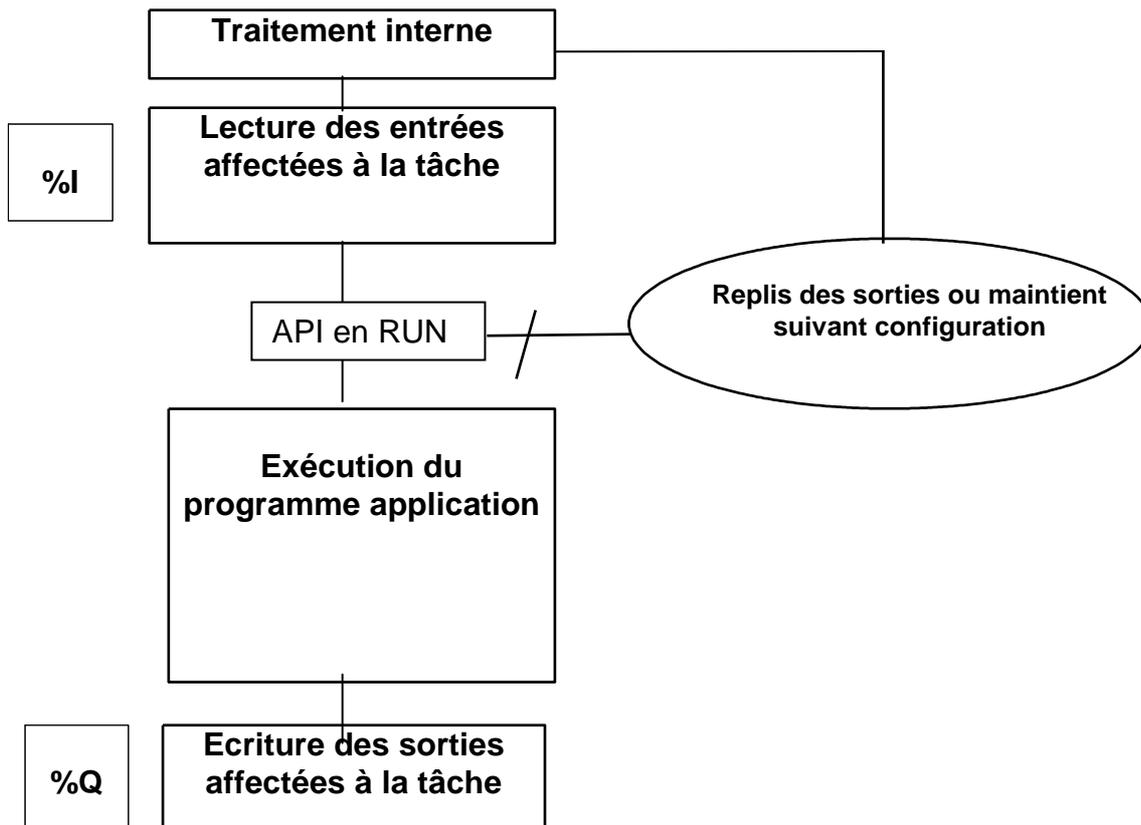
STRUCTURE DE LA MEMOIRE



STRUCTURE MULTITÂCHE



SCRUTATION D'UNE TÂCHE



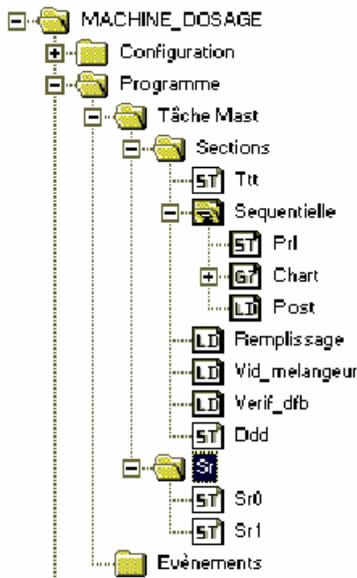
STRUCTURE D'UNE TACHE

Une tâche d'un programme PL7 se compose

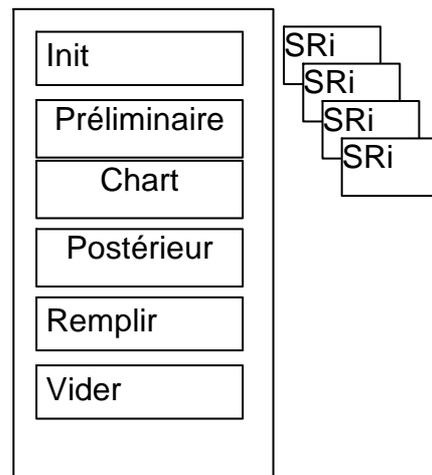
- ◆ de plusieurs parties appelées sections
- ◆ de sous-programmes.

Chacune de ces sections peut être programmée dans le langage approprié au traitement à réaliser (LD, IL, ST, Grafcet).

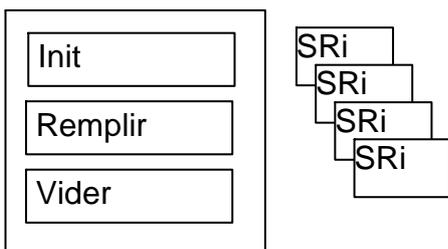
Exemple :



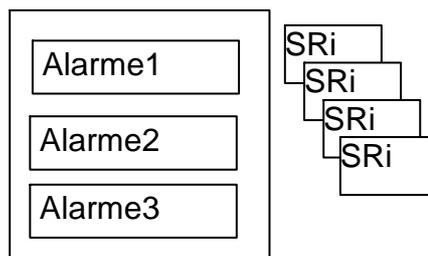
Tâche maître avec grafcet



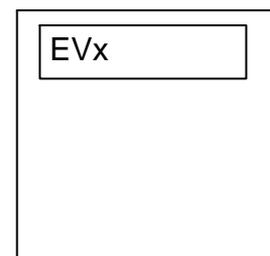
Tâche maître sans grafcet



Tâches rapide



Tâches événementielle



- ◆ Les sections et les SR sont créés à l'aide de la fonction Editer, option créer.
- ◆ Les sections sont exécutées dans l'ordre de leur apparition dans le dossier de la tâche. Il est possible de modifier de modifier cet ordre ensuite
- ◆ Toutes les sections peuvent être soumises à condition d'exécution. Cette condition se définit en accédant à la fenêtre «propriété» . (Edition, option Propriété)

CONFIGURATION DE L'APPLICATION



CONFIGURATION PROCESSEUR

Double cliquez sur la position du processeur

La tâche maître peut être cyclique ou périodique

On choisit la période de la tâche rapide

Configuration
 Désignation : PROCESSEUR TSX P 57252

Mode de Marche
 Entrée Run/Stop
 Démarrage Automatique en Run
 RAZ des %MWi sur reprise à froid

Carte Mémoire
 Aucune

Tâches

MAST
 Cyclique
 Périodique 40 ms
 Chien de Garde: 250 ms

Mode Fipio
 Asservi Temps de cycle réseau (calculé):
 Libre 0 ms

FAST
 Période: 5 ms
 Chien de Garde: 100 ms

Mode Fipio
 Asservi Temps de cycle réseau (calculé):
 Libre 6090

CONFIGURATION DE TACHE EVENEMENTIELLE

Double cliquez sur le module d'entrée

Configuration
 Désignation : 16E RAPID 24VCC SINK CONN

Voie	Symbole	Tâche	Filtrage	Fonction
0		MAST	4 ms	FM Evènement 0
1			4 ms	
2			4 ms	
3			4 ms	
4			4 ms	
5			4 ms	
6			4 ms	
7			4 ms	
8		MAST	4 ms	
9			4 ms	
10			4 ms	
11			4 ms	
12			4 ms	
13			4 ms	
14			4 ms	
15			4 ms	

En double cliquant sur la zone fonction, on peut choisir d'associer à la voie, un événement

Adressage des Objets BITS ET MOTS

PRESENTATION GENERALE

QU'EST QU'UN OBJET

Un objet est une entité pouvant être manipulée par programme, ce peut être une image d'entrée, un élément d'un temporisateur, un élément du système, un élément de communication etc.

REPRESENTATION DES OBJETS

Les objets sont représentés par le symbole « % » suivi d'une ou deux lettres précisant leurs type puis d'une lettre précisant leur format (bits, octet, mots , double, réel etc.)

Exemples :

%IW	% : objet	I	: Type image d'entrée	W	: Format mot
%MB	% : objet	M	: Type mémoire interne	B	: Format octet
%Q ou %Qx	% : objet	Q	: Type Image de sortie	X	: Format bit

Liste des différents types d'objets

- | | | |
|------------------------------|-----------|---|
| ◆ OBJETS D'ENTREES | %I | <i>Images des entrées process</i> |
| ◆ OBJETS DE SORTIES | %Q | <i>Images des sorties process</i> |
| ◆ VARIABLES INTERNES | %M | <i>Mémoire utilisateur</i> |
| ◆ VARIABLES GRACETS | %X | |
| ◆ CONSTANTES | %K | <i>Mémoire constante ou de configurations</i> |
| ◆ VARIABLES SYSTEME | %S | <i>Etats ou actions sur le système</i> |
| ◆ VARIABLES RESEAUX (FIPWAY) | %N | <i>Mots communs échangés automatiquement</i> |

Les types des objets des blocs fonctions prédéfinis (Pas de précision de format)

- | | | |
|-----------------------------|------------|---------------------------------------|
| ◆ TEMPORISATEURS | %TM | <i>Repos, travail</i> |
| ◆ TEMPORISATEURS SERIE 7 | %T | <i>Compatible série 7</i> |
| ◆ MONOSTABLES | %MN | <i>Monostables « retriggerables »</i> |
| ◆ COMPTEURS | %C | <i>Comptage, décomptage</i> |
| ◆ REGISTRES | %R | <i>Pile FIFO ou LIFO</i> |
| ◆ PROGRAMMATEURS
tambour | %DR | <i>Programmateurs cycliques à</i> |

Les types DFB (Pas de précision de format)

LES FORMATS DES OBJETS

BITS	X ou rien	<input type="checkbox"/>	0 - 1
OCTETS	B	<input style="width: 50px;" type="text"/>	Code ASCII uniquement
MOTS	W	<input style="width: 100px;" type="text"/>	16 bits signé ou pas
MOTS DOUBLES	D	<input style="width: 200px;" type="text"/>	32 bits signé
FLOTTANT	F	<input style="width: 200px;" type="text"/>	-3.402824E+38 et -1.175494E-38 et 1.175494E-38 et 3.402824E+38

LES ELEMENTS DES OBJETS

La plupart des objets sont constitués de plusieurs éléments ou sous éléments, par exemple, un compteur contient une valeur courante, une valeur de présélection, deux bits de dépassements et un bit de présélection atteinte

Les cartes d'entrées TOR disposent des bits représentant les états physiques des capteurs, mais aussi d'un bit de défaut du module et un bit de défaut par voie, certaines cartes disposent aussi de mots ou de tableaux de mots représentant leurs états ou leurs configurations.

On accède à un élément d'un objet en mettant un point derrière le numéro de l'objet puis l'identificateur de l'élément (un numéro ou un symbole réservé), puis si l'élément contient d'autre sous éléments on ajoute un point suivi de l'identificateur etc.

Exemples :

%I0.1	Module à l'emplacement 0 , la voie 1
%I0.1.ERR	Module à l'emplacement 0 , la voie 1 , défaut module
%C5.V	Le compteur 5 , la valeur courante
%X3	L'étape 3
%X3.T	L'étape 3 , sont temps d'activité
%IW0.12.2	Module à l'emplacement 0 , voie 12 , mot 2

OBJETS DES CARTES

Pour identifier un objet d'une carte, il faut préciser son numéro de module puis le numéro de voie.

Si plusieurs objets sont associés à une voie, il faudra préciser le numéro d'objet. (sauf pour l'objet N°0)

Exemples:

%IW0.12.2	Mot 2 de la voie 12 du module 0
%IW0.12	Mot 0 de la voie 12 du module 0

LES OBJETS STRUCTURES

(bits extraits de mots, tableaux de bits, tableaux de mots)

BIT EXTRAIT DE MOTS

mot : Xi	i : numéro du bit
Exemples : %MW5 :X4	Le mot interne 5 , le bit 4
%IW0.12.2 :X3	Le module 0 , la voie 12 , le registre 2 , le bit 3

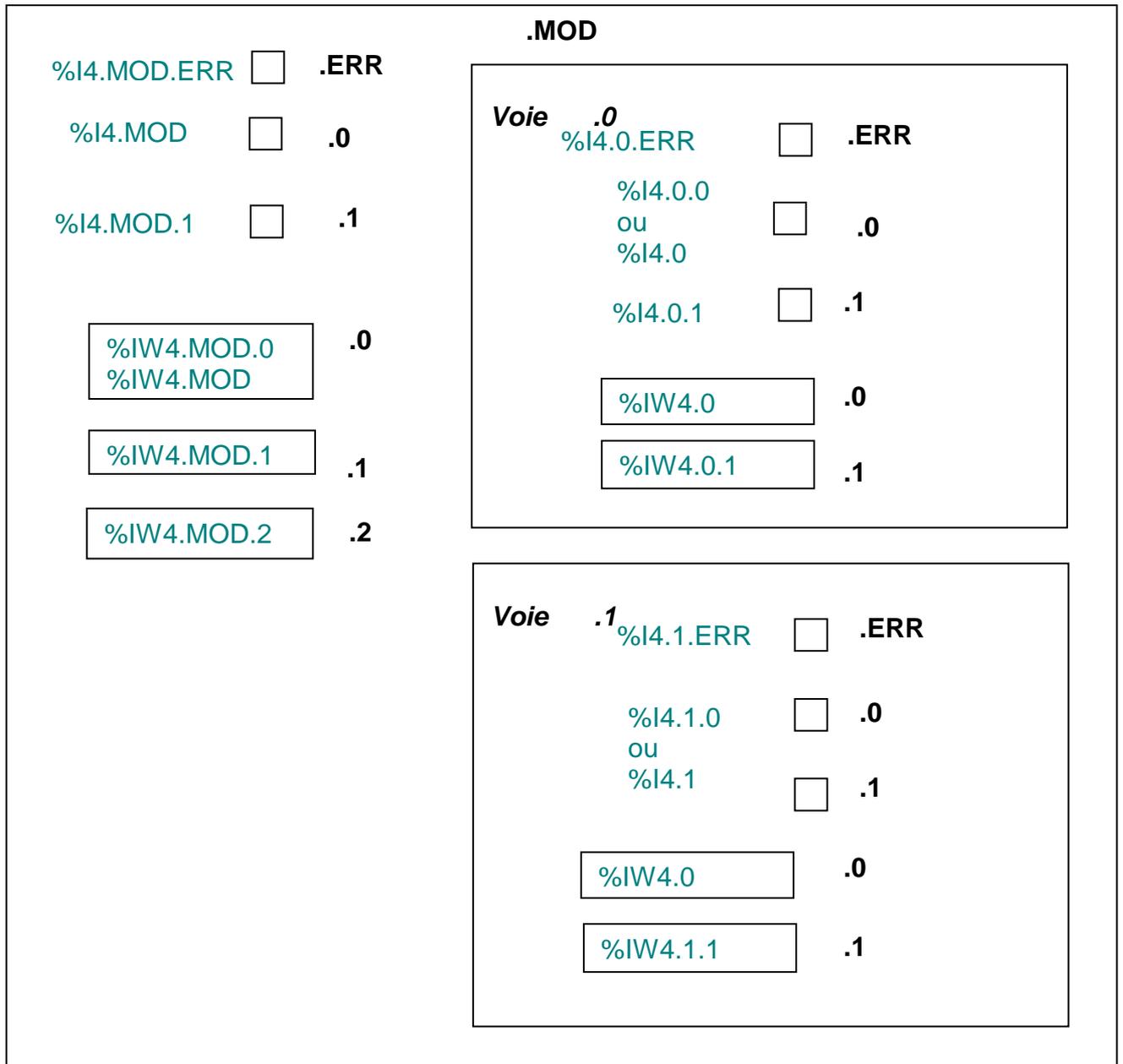
TABLEAUX DE BITS OU CHAINES DE BITS

bit de départ : nombre	nombre compris entre 1 et 32
Exemples : %I0.0 :16	Module 0 , voie 0 , 16 bits (voie 0 à 15)
%M5 :32	Bit 5 , 32 bits (%M5 à %M36)

TABLEAU DE MOTS

mot de départ : nombre	nombre compris entre 1 et maximum mémoire
Exemples : %MW10 :50	Mot interne 10 , 50 mots (%MW10 à %MW59)
%KW25 :100	Mot constant 25 , 100 mots (%KW25 à %KW124)
%MD20 :10	Mot double 20 , 10 mots (%MD20 à % MD38)

Exemple d'objets d'une carte d'entrée à l'emplacement 4



LES FORMES D'ADRESSAGES

L'ADRESSAGE DIRECT

Exemple d'adresse directe : %MW20
L'adresse est figée, et définie par programme.

L'ADRESSAGE INDEXE

L'adresse directe est complétée d'un index, contenu dans un mot.

Indexation sur mot simple

Exemples : %MW10[%MW5]

Si %MW5 = 2, le mot adressé sera %MW12

L'adresse se calcule : 10 + le contenu de %MW5

Indexation sur mot double

Exemple : %KD4[%MW20]

Si %MW20 = 5, le mot double adressé %MW14

L'adresse se calcule : 4 +2 fois le contenu de %Mw20

<i>Type</i>	<i>Format</i>	<i>Adresse</i>
Bits d'entrées	Booléen	%I[%MWj]
Bits de sorties	Booléen	%QI[%MWj]
Bits internes	Booléen	%MI[%MWj]
Mots internes	Simple longueur	%MWi[%MWj]
	Double longueur	%MDi[%MWj]
	Flottant	%MFi[%MWj]
Mots constantes	Simple longueur	%KWi[%MWj]
	Double longueur	%KDi[%MWj]
	Flottant	%KFi[%MWj]
Tableau de mots	<Objet> [%MWj]:L	%MWi[%MWj]:L

EXERCICE 1.ADRESSAGE DES OBJETS

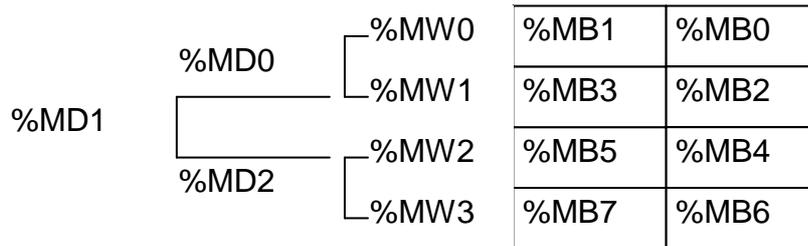
Identifiez les objets suivants

Mot interne n°25	Exemple	%MW25
Bit interne n°57		
Temporisateur normalisé n°0		
Bit d'entrée de la voie 5 du module 2		
Bit de sortie de la voie 12 du module 1		
Compteur n°7		
Bit d'étape 14		
Temps d'activité de l'étape 14		
Bit 5 du mot interne n°12		
Bit 15 du mot d'entrée de la voie 1 du module 7		
Bit 3 du mot d'entrée 2 de la voie 12 du module 0 de comptage		
Chaîne de bit d'entrées des voies 4 à 7 du module 204		
Chaîne des bits internes 20 à 35		
Tableau des mots constants 1 à 30		
Tableaux des mots internes 100 à 499		

ORGANISATION MEMOIRE OBJETS

LE RANGEMENT DES MOTS EN MEMOIRE

Les octets, mots simples, double longueur et flottants sont rangés à l'intérieur de l'espace données dans une même zone mémoire de la façon suivante :



Possibilité de recouvrement entre objets

Les octets %MB ne sont adressables que sous forme de chaîne de caractères
Les %MF et %MD occupent le même espace mémoire.

REPRESENTAION DES VALEURS IMMEDIATES

ENTIER DECIMAL

Entier décimal	12345	16 ou 32 bits
Exemples	%MW5 := 12345 %MD10 := 3657541	

HEXADECIMAL

Hexadécimal	16#7FE3	16 ou 32 bits
Exemples	%MW5 := 16#7FE3 %MD5:=16#F8FA0F2C	

BINAIRE

Binaire	2#1011001011000	16 bits
Exemple	%MW6 := 2#1011001011000	

CHAINE DE CARACTERE

Chaîne ANSI	'abcdABCD1234'
-------------	----------------

Le caractère de fin de chaîne est la valeur NULL 0

Une chaîne de caractère est stockée dans des octets consécutifs rangés dans des mots de 16 bits.

Exemple %MB10:8 := 'BONJOUR'

Le bit %S15 surveille les dépassements d'adresses

LES DATES

Le format Date	1997-05-23	32 bits
	Une date doit être représentée en BCD sur 32 bits	

y	y	y	y	m	m	d	d
---	---	---	---	---	---	---	---

Exemple %MD5:=1997-5-23

LES HEURES

Le format Heure	23:12:05	32 bits
	Une heure doit être représentée en BCD sur 32 bits	
	Ce format est exclusivement employé avec la fonction trans_time	

		h	h	m	m	s	s
--	--	---	---	---	---	---	---

Exemple %MD10:=TRANS_TIME(1252)

LES DUREES

le format Durée	3600.2	32 bits
	Représente une durée en 1/10em de secondes de 0 à 42949667295	

LES FLOTTANTS

le format flottant

1285.28

ou 12.8528E2

32 bits

le flottants sont compris entre $3.402824E+38$ et $-1.175494E-38$, et $1.175494E-38$ et $3.402824E+38$.

Exemples

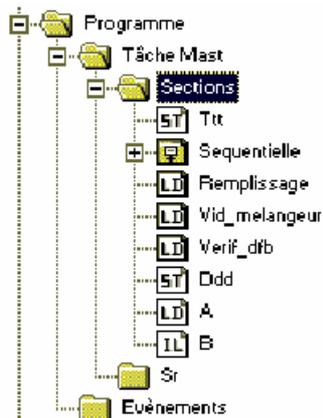
%MF6:=1234.25

%MF5:=1.2e24

%MF6 occupe le même espace mémoire que %MD6 ainsi que %MB12, %MB13, %MB14 et %MB15

Création d'une section

Dans le navigateur, se positionner sur le dossier Section, puis choisir le menu Edition, option Créer



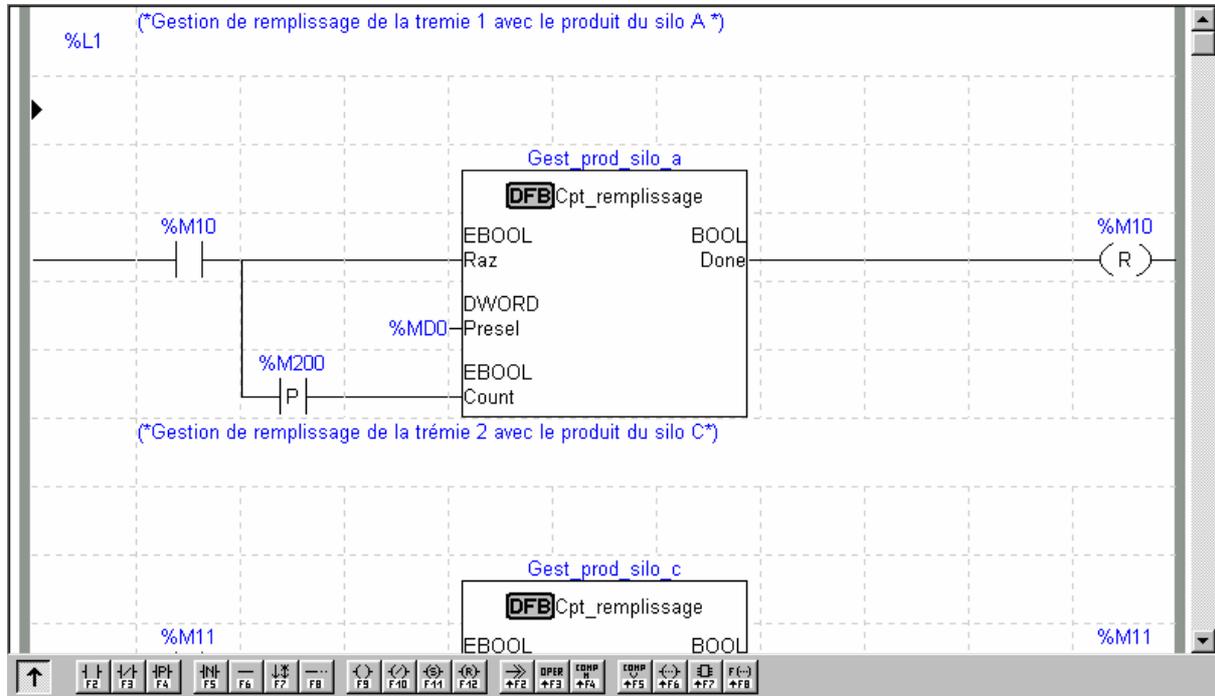
La fenêtre de création de section, permet de :

- ◆ Choisir le nom de la section
- ◆ Définir le langage à utiliser
- ◆ Choisir la condition d'activation de la section s'il y a lieu
- ◆ Associer un commentaire à la section
- ◆
- ◆ Toutes ces éléments seront modifiables par la suite en se positionnant sur la section, puis en affichant la fenêtre « Propriété de la section ».

The image shows a dialog box titled 'Créer...'. It has the following fields and controls:

- 'Nom' : a text input field.
- 'Tâche' : 'MAST' (displayed next to a small icon).
- 'Langage' : a dropdown menu showing 'ST'.
- 'Condition' : a section containing:
 - 'Requière' : a text input field.
 - 'Synchrone' : a text input field.
 - 'Forçable' : a checkbox.
- 'Commentaire' : a text input field.
- 'Commentaire' : a larger text input field at the bottom.
- 'OK' and 'Annuler' buttons.

LANGAGE A CONTACTS



Le langage à contacts permet de manipuler l'ensemble des objets automates

INSTRUCTION DE TEST D'OBJETS BITS

%I %Q %M %X %MW :Xi %Ti.D Etc.



- Test un front descendant
- Test un front montant
- Test un bit à 0
- Test un bit à 1

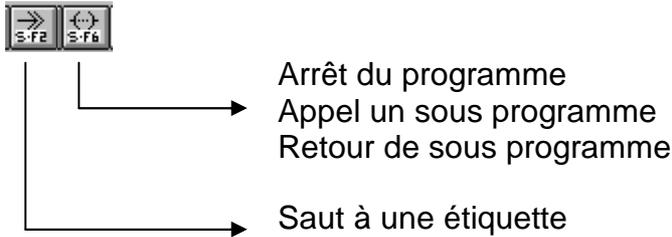
Uniquement sur bits
%I %Q %M

INSTRUCTION DE MISE A JOUR DE BITS

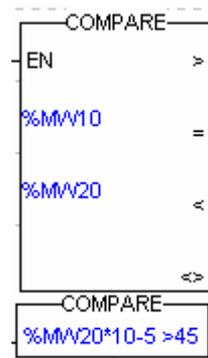
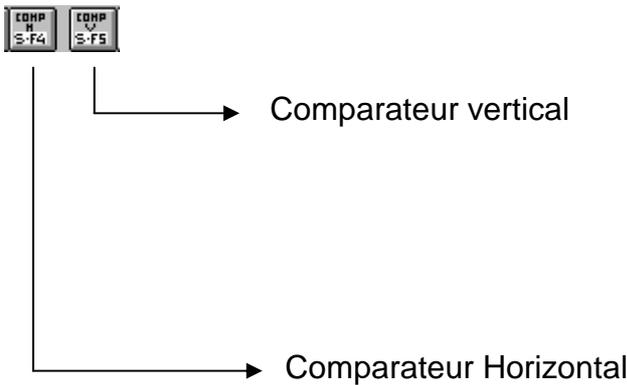


- Mise à 0 d'un bit
- Mise à 1 d'un bit
- Affecte le complément du test
- Affecte le résultat du test

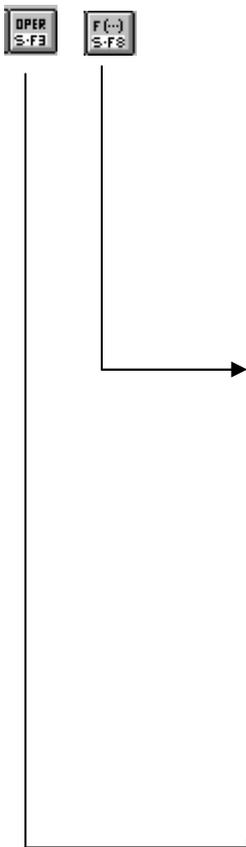
INSTRUCTIONS DE BRANCHEMENT



INSTRUCTIONS DE COMPARAISONS



OPERATION



PL7 : Fonctions en bibliothèque

EF DFB

Informations Fonctions : Paramètres

Famille	V.Bib	V.App	Nom	Commentaire
Chaînes de caractères	2.00	-	CONCAT	Concaténation de deux chaînes
Commande de Mouvement	1.00	-	DELETE	Suppression d'une sous chaîne
Communication	3.00	-	DINT_TO_STRING	Conversion d'un entier double longu >
Conversions numériques	2.00	-	EQUAL_STR	Recherche du premier caractère différent

Format d'appel

Paramètres de la FONCTION :

Nom	Type	Nature	Commentaire	Zone de saisie
IN1	STRING	IN	Tableau contenant la 1ere chaîne	
IN2	STRING	IN	Tableau contenant la 2eme chaîne	

Résultat de la FONCTION :

Nom	Type	Commentaire
OUT	STRING	Tableau contenant la chaîne concaténée

Visualisation de l'appel

CONCAT ()

OK Annuler

Opérations :
 + - * / REM SQRT ABS
 INC DEC
 AND OR XOR NOT



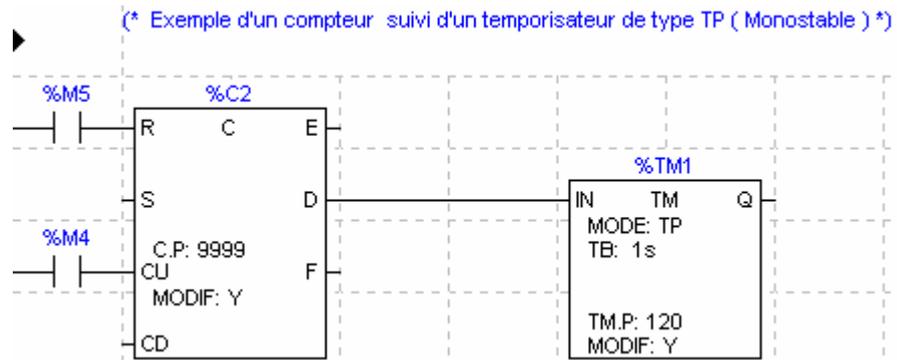
FONCTIONS GRAPHIQUES

Les fonctions graphiques comprennent :

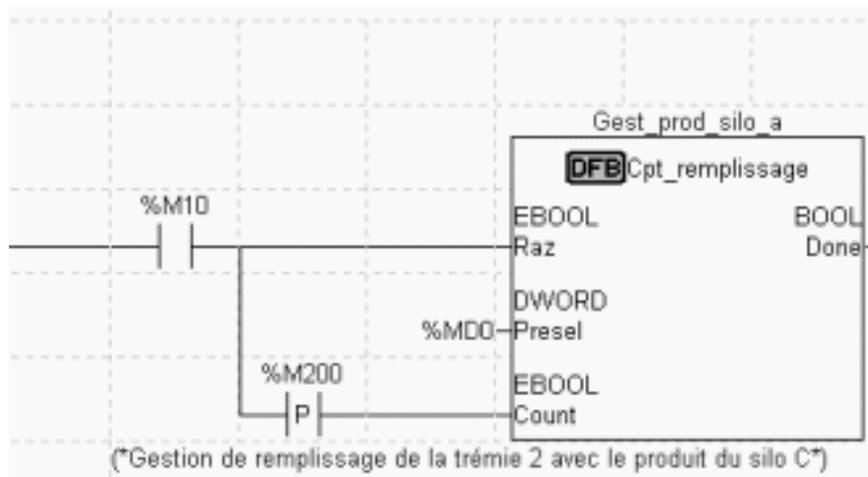
- ◆ Les appels aux blocs fonctions prédéfinis
- ◆ Les appels au DBF (« Blocs fonctions » créer par le programmeur)



Exemple de bloc fonction compteur et temporisation



Exemple de DBF



EXERCICE 2.LANGAGE LADDER

1 / Réaliser une commande Marche / Arrêt d'un moteur

Liste des entrées

- ◆ BP_Acquit
- ◆ BP_Marche
- ◆ BP_Arret
- ◆ R_KM1

Liste des sorties

- ◆ V_DEF
- ◆ KM1

a : Définissez les entrées sorties dans l'éditeur de variable

b : Ecrivez le programme en LADDER dans la section « moteur » de la tâche MAITRE

c : Transférer le programme dans l'automate

d : Testez et visualisez le programme en ligne

2 / Allumez le voyant de défaut lorsqu'il y a discordance entre la sortie automate et le retour du contacteur, le défaut doit être mémorisé, il faudra actionner le Bp_acquit pour faire disparaître le défaut.

L'EDITEUR DE VARIABLES

Tous les objets de l'automate peuvent être symbolisés, on devrait dire **doivent** être symbolisés, la programmation moderne s'oriente vers la manipulations d'objets **concrets** tels que "Chauffage_gaine", "monter_pince", "porte_ouverte", "température_trop_basse" etc. La programmation fait abstraction de l'adresse de la variable, voire même de l'appellation constructeur (c'est le principe de la norme IEC 1131 qui tend à homogénéiser les langages de programmation).

Avantages de la programmation symbolique

- 1 Lecture et compréhension du programme facilité pour la maintenance.
- 2 Diminution du risque d'erreurs de programme (utilisation d'une variable à la place d'une autre, réutilisation de la même variable etc.)
- 3 Passage de l'analyse fonctionnelle à l'analyse organique puis analyse détaillée et enfin programmation facilité puisque les termes utilisés restent identiques, les termes représentent les objets du terrain sans ambiguïté.
- 4 Mise au point et modification du programme facilité par la clarté de la lecture.

UTILISEZ DES SYMBOLES

Les symboles reste sur disque, ils ne sont pas sauvegarder dans l'API !!!



Variables - MEMOIRE

Repère	Type	Symbole	Commentaire
%M0	EBOOL	Dcy	Départ de cycle
%M1	EBOOL	Evt_1	Vanne silo A (=0 fermée = 1 ouverte)
%M2	EBOOL	Evt_2	Vanne silo B (=0 fermée = 1 ouverte)
%M3	EBOOL	Evt_3	Vanne silo C (=0 fermée = 1 ouverte)
%M4	EBOOL	Evt_4	Vanne trémie B1 (=0 fermée = 1 ouverte)
%M5	EBOOL	Evt_5	Vanne trémie B2 (=0 fermée = 1 ouverte)
%M6	EBOOL	Evt_6	Vanne mélangeur (=0 fermée = 1 ouverte)
%M7	EBOOL	Tre_b1_vid	Trémie B1 vide
%M8	EBOOL	Tre_b2_vid	Trémie B2 vide
%M9	EBOOL	Mel_vid	Mélangeur vide
%M10	EBOOL	Trig_prod_a	Triger pour remplissage produit A dans trémie 1
%M11	EBOOL	Trig_prod_c	Triger pour remplissage produit C dans trémie 2
%M12	EBOOL	Trig_prod_b	Triger pour remplissage produit B dans trémie 1
%M13	EBOOL	Trig_melang	Triger pour remplissage produit A B C dans mélangeur.
%M14	EBOOL	Cd_moteur	Commande moteur malaxage (= 1 marche = 0 arrêt)
%M15	EBOOL	Fin_malax	Fin du malaxage produits A B C (10 sec)
%M16	EBOOL	Trig_vid_melang	Triger de vidange mélangeur
%M17	EBOOL	Trig_tempo	Triger temporisateur de malaxage

Variables - DFB

Nom	Type	Nature	Val. Init	Commentaire
Gest_vidange_melangeur	Cmpt_vidange	-	-	
Vidange_melangeur	Cmpt_vidange	-	-	

Les blocs fonction Prédefinis

LES TEMPORISATEURS

%TM ou **%T** ou **%MN**

Il existe 3 modèles de temporisateurs, le temporisateur normalisé **%TM**, le temporisateur de la série 7 **%T** et le monostable **%MN**.

PRINCIPE GENERAL

Prédéfini en configuration, peut être redéfini par programme ou en mode table de variables

Valeur de présélection

0 à 9999



1 - Chargement de la valeur courante

Peut être lue par programme

Valeur courante

Défini en configuration

Base de temps 10 ms, 100 ms, 1 s, 1 mn

2 - Ecoulement de la valeur courante au rythme de la base de temps

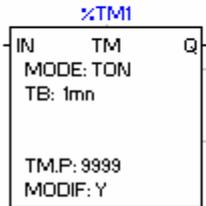
2 bis - Arrêt de l'écoulement de la valeur courante (Uniquement sur %T)

3 - Test de l'état de la temporisation

Remarque

Les temporisateur **%TM** sont mis à 0, puis la valeur courante est incrémentée jusqu'à la valeur de présélection.

TEMPORISATEUR %TM



Le temporisateur dispose de 3 modes de fonctionnement :

.TON : ce mode permet de gérer des retards à l'enclenchement. Ce retard est programmable et peut être modifiable ou non par terminal.

.TOF : ce mode permet de gérer des retards au déclenchement. Ce retard est programmable et peut être modifiable ou non par terminal.

.TP : ce mode permet d'élaborer une impulsion de durée précise. Cette durée est programmable et peut être modifiable ou non par terminal.

Valeur de présélection %TMi.P

Valeur courante %TMi.V

Etat de la sortie %TMi.Q

Remarque : La valeur courante part de 0 puis croit jusqu'à la valeur de présélection (à l'inverse des tempos série 7 %T)

La configuration se fait depuis l'éditeur de variables

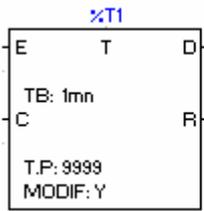
Programmation en langage IL

LD	bit	Test un bit
IN	%TMi	Démarre la tempo
LD	Q	Test la sortie
ST	bit	Range dans un bit

Programmation en langage Structuré

Start	%Mi	Démarrage
Down	%Mi	Mise à 0 de la valeur courante

TEMPORISATEUR SERIE 7 %T



Ce bloc fonction temporisateur compatible avec les blocs série 7 PL7-2/3 permet de commander avec retard des actions spécifiques. La valeur de ce retard est programmable et peut être modifiable ou non par terminal.

1 - Préselectionner la valeur courante

L'entrée E = 0

2 - Faire écouler la valeur courante

L'entrée C = 1

2 bis - Arrêter l'écoulement (temporairement)

L'entrée C = 0

Valeur de préselection **%Ti.P**

Valeur courante **%Ti.V**

Etat tempo écoulée **%Ti.D**

Etat tempo en cours **%Ti.R**

Programmation en langage IL

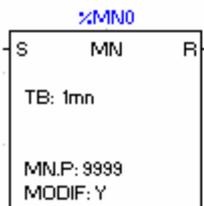
On ne peut pas agir sur une tempo %T en langage IL, mais on peut tester les objets internes %Ti.D, %Ti.R, %Ti.P, %Ti.V

ex : **LD** **%T2.D**
ST **%M54**

Programmation en langage structuré

Preset	%Ti	Initialisation de la valeur courante à %Ti.P
Start	%Ti	Écoulement
Stop	%Ti	Arrêt de l'écoulement

MONOSTABLE



Le bloc fonction monostable permet d'élaborer une impulsion de durée précise. Cette durée est programmable et peut être modifiable ou non par terminal.

Valeur de préselection %MNi.P

Valeur Courante %MNi.V

Etat impulsion en cours %MNi.R

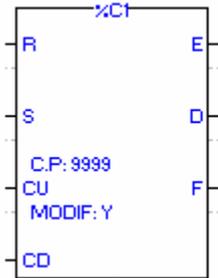
Programmation en langage IL

S %MNi Démarrage d'une impulsion

Programmation en langage structuré

Start %MNi Démarrage d'une impulsion

COMPTEURS



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

PRINCIPE ET OBJETS DU COMPTEUR

0 à 9999			
Valeur Courante	%Ci.V		MOT
0 à 9999			
Valeur de présélection	%Ci.P		MOT
Présélection atteinte	<input type="checkbox"/> %Ci.D		BIT
Dépassement > 9999	<input type="checkbox"/> %Ci.F		BIT
Dépassement < 0	<input type="checkbox"/> %Ci.E		BIT

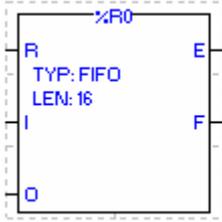
Programmation en langage IL

R	%Ci	Mise à 0 du compteur
S	%Ci	Mise de la valeur courante à la valeur de présélection
CU	%Ci	Incrémentation du compteur
CD	%Ci	Décrémentation du compteur

Programmation en langage structuré

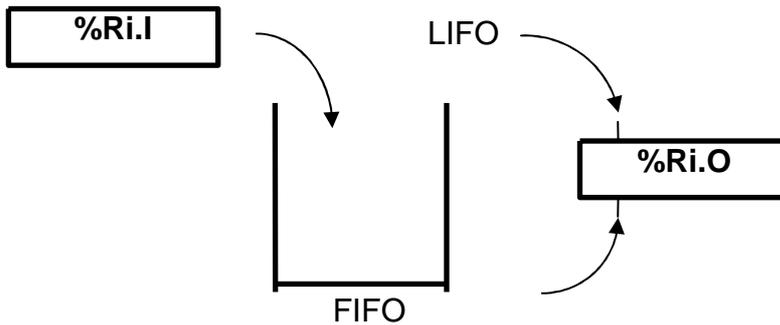
Reset	%Ci	Mise à 0 du compteur
Preset	%Ci	Mise de la valeur courante à la valeur de présélection
Up	%Ci	Incrémentation du compteur
Down	%Ci	Décrémentation du compteur

REGISTRES



Un registre est un bloc mémoire permettant de stocker jusqu'à 255 mots de 16 bits de deux manières différentes :

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



Mot d'entrée	%Ri.I
Mot de sortie	%Ri.O
Etat pile pleine	%Ri.F
Etat pile vide	%Ri.E

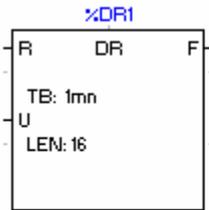
Programmation en langage IL

I	%Ri	Empile
O	%Ri	Dépile
R	%Ri	Vide la pile

Programmation en langage structuré

Put	%Ri	Empile
Get	%Ri	Dépile
Reset	%Ri	Vide la pile

PROGRAMMATEUR CYCLIQUE DRUM



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

Numéro du pas en cours	%DRi.S	mot
Temps écoulé du pas	%DRi.V	mot
Etats des cames d'un pas "j"	%DRi.Wj	mot
Dernier pas atteint	%DRi.F	bit

Base de temps pour DRi.V défini à la configuration du drum (10ms, 100 ms, 1sec, 1min)

Configuration depuis l'éditeur de variable, choisir **FB PREDEFINIS** et **DR**, cliquez sur la colonne "pas"

		PAS				
		0	1	2	3	Repère
BIT	0	0	1	1	0	%M1
	1	0	0	1	0	%Q2.1
	2	0	0	1	1	%Q2.2
	3	0	0	0	1	%M3
	4	0	0	0	1	%M4
	5	0	0	0	0	
	6	0	0	0	0	
	7	0	0	0	0	
	8	0	0	0	0	
	9	0	0	0	0	
	A	0	0	0	0	
	B	0	0	0	0	
	C	0	0	0	0	
	D	0	0	0	0	
	E	0	0	0	0	
	F	0	0	0	0	

EXERCICE 3.BLOCS FONCTIONS FB prédéfinis

1/ Temporisation de la discordance

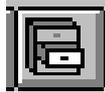
Reprenez l'exercice précédent, temporiser la mise à "1" du défaut discordance (10 secondes)

BIBLIOTHEQUE DE FONCTIONS

PRESENTATION

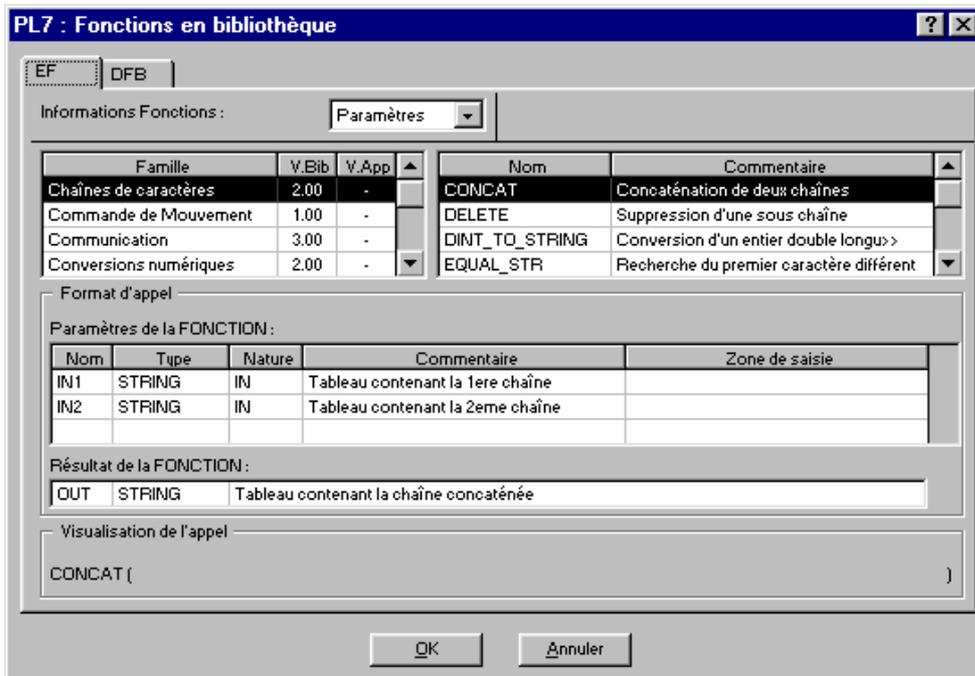
Le logiciel PL7 Pro disposent de

- ◆ Fonctions classées par familles **EF**
- ◆ Fonction créer par le programmeur **DFB**



Accès à la liste des fonctions disponibles

En mode programmation on peut accéder à l'éditeur de fonction par le menu SERVICE, SAISIR L'APPEL D'UNE FONCTION ou directement par SHIFT+F8



FAMILLE DE FONCTION EF

CONVERSIONS NUMERIQUES

- ◆ Conversion BCD<->BINAIRE,
- ◆ GRAY->BINAIRE,
- ◆ FLOTTANTS<->BINAIRE

CHAINES DE CARACTERES

- ◆
- ◆ Conversions CHAINE<->NUMERIQUE,
- ◆ Concaténation de chaînes,
- ◆ Extractions de sous chaînes,
- ◆ remplacement de chaînes,
- ◆ recherche d'une sous chaîne,
- ◆ comparaison de chaînes

DATES ET DUREES

- ◆ Conversion de date et durées en chaînes,
- ◆ jour de semaine,
- ◆ écart entre deux dates, etc.

TEMPORISATION

- ◆ Sortie créneau
- ◆ Au déclenchement
- ◆ A l'enclenchement
- ◆ impulsion

COMMUNICATION

- ◆ Echanges de données,
- ◆ lecture/écriture,
- ◆ requêtes,
- ◆ lecture de télégrammes,
- ◆ décalage des trames reçut d'un octet vers la droite, etc.

OPERATIONS SUR ENTIER DOUBLE ET SIMPLE LONGUEUR

- ◆ Valeur absolue,
- ◆ Décalage à gauche, à droite ,
- ◆ Décalage circulaire
- ◆ Racine carré

GESTION DES CCX 17

- ◆ Affichages de message,

- ◆ lectures, émission d'alarme,
- ◆ configuration des clefs, etc.

GESTION DES EVENEMENTS

- ◆ Masquage,
- ◆ démasquage.

FONCTIONS DE REGULATION

- ◆ Algorithmes PID,
- ◆ commande PWM,
- ◆ commande servo moteur

FONCTIONS D'ECHANGES EXPLICITES

- ◆ Lecture, écriture de mots d'états ou de commande,
- ◆ lecture écriture de paramètres.

FONCTIONS SUR TABLEAUX DE MOTS, BITS, MOTS DOUBLES

- ◆ Opérations Comparaisons,
- ◆ recherches,
- ◆ nombre d'occurrence,
- ◆ Décalages circulaires,
- ◆ tris,
- ◆ somme etc.

LES FONCTIONS DBF

Les DBF sont des blocs fonctions créer par le programmeur

Les DBF peuvent être créer avec le logiciel PL7 pro, mais ne peuvent être appeler qu'à partir des logiciels PL7 pro et PL7 Junior, uniquement sur automate TSX, PMX, PCX **57**

La création et la mise au point de DBF fait l'objet d'un chapitre particulier.

Les fonctions EF temporisateurs

PRINCIPE DE FONCTIONNEMENT

Toutes les fonctions de temporisations possède :

- ◆ Une entrée Armement : **EN**, booléen
- ◆ Déclenchement sur front montant ou front descendant suivant le type de temporisation
- ◆ Une ou deux valeur de présélection :
Mot d'entrée qui détermine la durée.
Cette durée est exprimée en centième de seconde (base de temps 10 ms)
Valeur maximum : 5 min et 27 s
- ◆ Mot de sortie : **ET** indique la valeur courante de la temporisation (écoulement croissant)
- ◆ Variable de calcul : **PRIV** Double mot utilisé pour mémoriser des états internes. Il faut associer à ce double mot une variable de l'application exclusivement réservée à cet effet.

Les valeurs de présélection sont modifiable par programme.

RETARD A L'ENCLenchement FTON

Permet de gérer des retards à l'enclenchement. Ce retard est programmable .

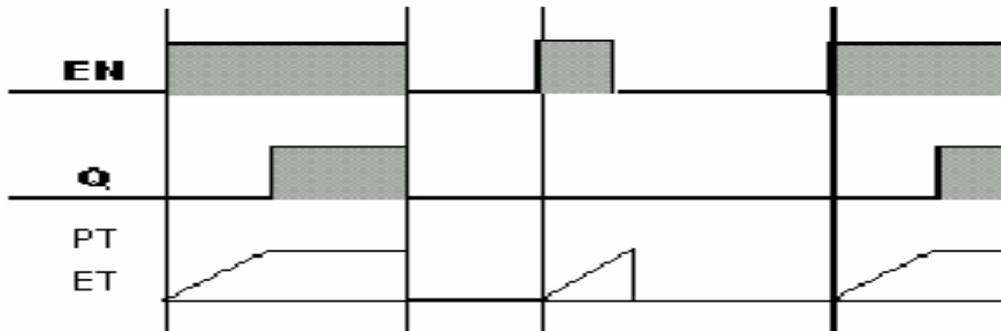
SYNTAXE

FTON (EN, PT, Q, ET, PRIV)

DESCRIPTION

Entrée "Armement"	EN	Sur front montant démarre la temporisation
Valeur de présélection	PT	Mot d'entrée.
Sortie "Temporisateur"	Q	Sortie mise à 1 en fin de temporisation
Valeur courante	ET	Mot de sortie
Variable de calcul	PRIV	Double mot

CHRONOGRAMME DE FONCTIONNEMENT



EXEMPLE

// Cette fonction peut s'écrire en langage littéral ou dans une opération en LADDER

```
FTON(%M52,100,%M53,%MW50,%MD200)
```

%M52 : Bit de déclenchement de la tempo

100 : Valeur de la tempo en 1/100em de seconde

%M53 : Bit de fin de tempo

%MW50 : Mot pour la valeur courante

%MD200 : Double mot utilisé par la fonction (obligatoire)

RETARD AU DECLENCHEMENT FTOF

Permet de gérer des retards au déclenchement. Ce retard est programmable .

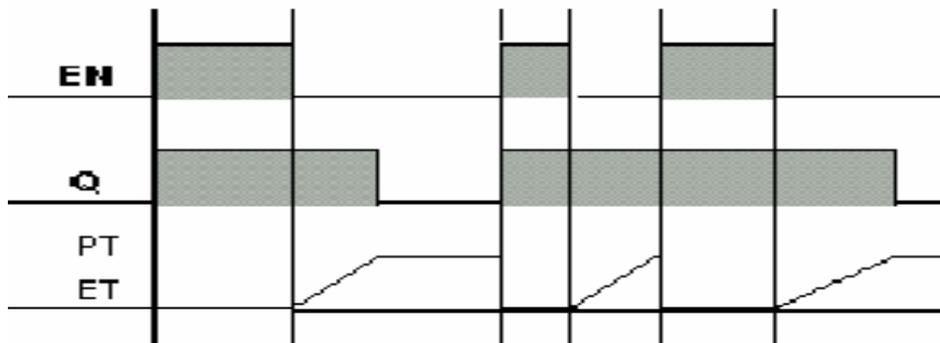
SYNTAXE

FTOF (EN, PT, Q, ET, PRIV)

DESCRIPTION

Entrée "Armement"	EN	Sur front descendant démarre la temporisation
Valeur de présélection	PT	Mot d'entrée
Sortie "Temporisateur"	Q	Sortie mise à 1 sur front montant de EN et mis à 0 en fin de temporisation.
Valeur courante	ET	Mot de sortie
Variable de calcul	PRIV	Double mot

CHRONOGRAMME DE FONCTIONNEMENT



EXEMPLE

// Cette fonction peut s'écrire en langage littéral ou dans une opération en LADDER

```
FTOF(%M52,100,%M53,%MW50,%MD200)
```

%M52 : Bit de déclenchement de la tempo

100 : Valeur de la tempo en 1/100em de seconde

%M53 : Bit de sortie de la tempo

%MW50 : Mot pour la valeur courante

%MD200 : Double mot utilisé par la fonction (obligatoire)

IMPULSION FTP

Permet d'élaborer une impulsion de durée précise. Ce retard est programmable .

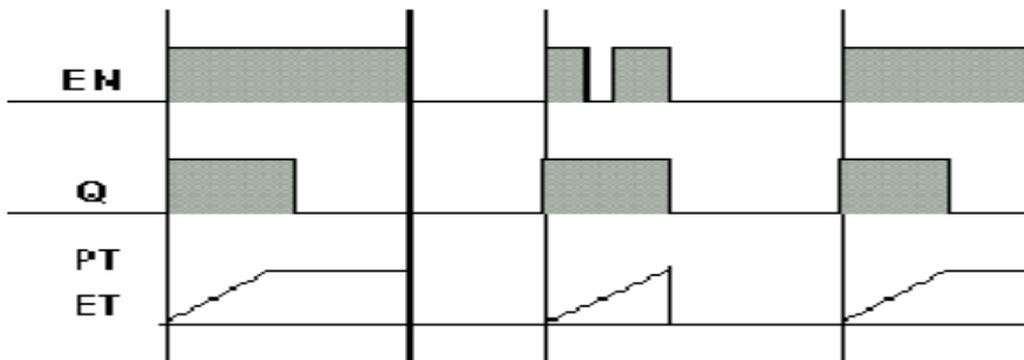
SYNTAXE

FTP (EN, PT, Q, ET, PRIV)

DESCRIPTION

Entrée "Armement"	EN	Sur front montant démarre la temporisation
Valeur de présélection	PT	Mot d'entrée
Sortie "Temporisateur"	Q	Sortie mise à 1 pendant la durée de la temporisation
Valeur courante	ET	Mot de sortie
Variable de calcul	PRIV	Double mot

CHRONOGRAMME DE FONCTIONNEMENT



EXEMPLE

// Cette fonction peut s'écrire en langage littéral ou dans une opération en LADDER

```
FTP(%M52,100,%M53,%MW50,%MD200)
```

%M52 : Bit de déclenchement de la tempo

100 : Valeur de la tempo en 1/100em de seconde

%M53 : Bit de sortie de la tempo

%MW50 : Mot pour la valeur courante

%MD200 : Double mot utilisé par la fonction (obligatoire)

GENERATEUR DE SIGNAL RECTANGULAIRE FPULSOR

Permet de générer un signal rectangulaire périodique dont on peut faire varier la largeur du créneau à 1 et du créneau à 0 par programme au moyen de 2 temporisateurs :

TON : temporisation à la montée (pour le créneau à 1).

TOFF : temporisation à la retombée (pour le créneau à 0).

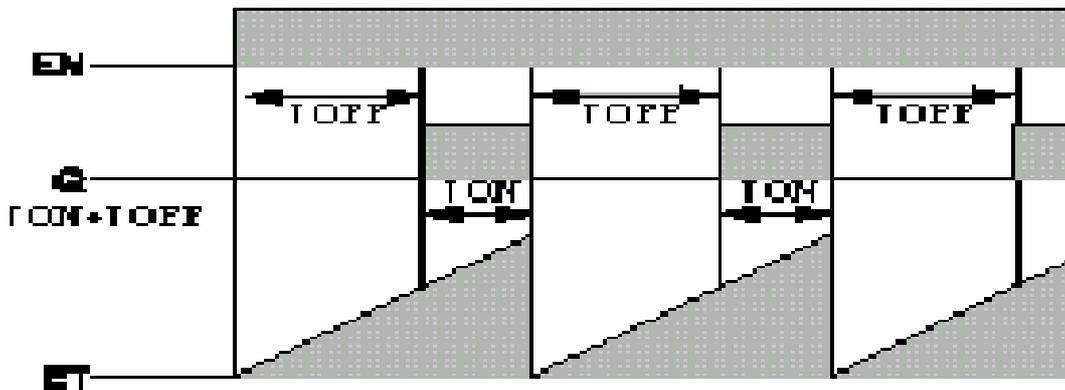
SYNTAXE

FPULSOR (EN, TON, TOFF, Q, ET, PRIV)

DESCRIPTION

Entrée "Armement"	EN	Sur front montant démarre la temporisation
Valeur de présélection	Ton	Mot d'entrée qui détermine la durée du créneau à 1
Valeur de présélection	Toff	Mot d'entrée qui détermine la durée du créneau à 0
Sortie "Temporisateur"	Q	Sortie Clignotante
Valeur courante	ET	Mot de sortie
Variable de calcul	PRIV	Double mot

CHRONOGRAMME DE FONCTIONNEMENT



EXEMPLE

FPULSOR(%M52,50,60,%M53,%MW50,%MD100)

%M52 : Bit de déclenchement de la tempo

50 : Valeur de la durée à l'état 1 en 1/100em de seconde

60 : Valeur de la durée à l'état 0 en 1/100em de seconde

%M53 : Bit clignotant de la tempo

%MW50 : Mot pour la valeur courante

%MD200 : Double mot utilisé par la fonction (obligatoire)

Exercice 4.Appel de Fonction « temporisateur »

1/ Modifier la gestion du voyant défaut de la façon suivante :

Sur apparition du défaut, faire clignoter le voyant défaut.

Sur appui de BP_Acquit, si le défaut est toujours présent, le voyant doit passer à l'état fixe, allumé, sinon si le défaut a disparu, éteindre le voyant

Le voyant défaut doit clignoter avec la contrainte suivante :



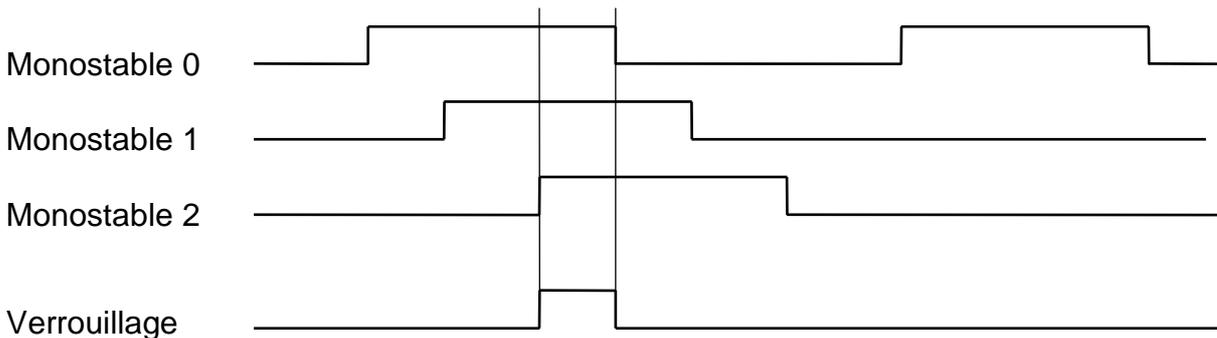
Pour cela, utiliser les fonctions temporisateurs, présentées au chapitre précédent.

2/ Limiter le nombre de démarrage dans l'heure (3 maxi)

A chaque démarrage, on active un monostable d'une heure, lorsqu'on a 3 monostables en services, on interdit tous nouveau démarrage.

Au moins deux solutions sont possibles

- ◆ Une solution avec compteur :
On compte les démarrages modulo 3 (0 à 2), si compteur = 0 on active le monostable %MN0, si compteur = 1 on active le monostable %MN1, si compteur = 2 on active le monostable %M2.
- ◆ Une solution sans compteur, mais avec détection de front montant : Plus difficile !



a - Comptez les démarrages modulo 3

b - Activer les monostables %M0 à %M1 en fonction de la valeur du compteur

c - verrouillez la commande de marche si 3 monostables en service

LANGAGE LISTE D'INSTRUCTION IL

PRESENTATION

Le langage IL se programme sous forme de phrases de 128 instructions maximum, comportant 1 commentaire et une étiquette.

Exemple

```
! (* Attente séchage *)           ! indique le début de la phrase  (* *) commentaire
%L2:                               Etiquette facultative
    LD   %M2                       Liste d'instructions
    OR   %I1.1
    ST   %Q2.4
```

```
! (* Une autre phrase *)
%L7:
    LD   TRUE
    [%MW2:=%MW4*5/SQRT(%MW20)]
```

Etc.

REMARQUE : Une phrase peut s'écrire en une seule ligne, l'éditeur la présentera sous la forme ci dessus après validation.

Exemples

```
! (*phrase IL*) %L4: LD [%MW10<4] AND [%MW20>10] [%MW40:=%MW50/2]
!(* autre phrase *) %L20 : LD %M10 AND %M11 ST %M12
!(* Une autre *) %L100 : LD %M20 AND %M21 OR ( %M22 AND %M23 ) ST %M24
Etc.
```

LES INSTRUCTIONS

INSTRUCTIONS BOOLEENNES

LD	Charge un résultat booléen (commence une phrase)
LDN	Charge le complément
LDF	Charge le front montant
LDR	Charge le front descendant
AND	Et
OR	Or
ANDN	Et pas
ORN	Ou pas
ANDF	Et front montant
ORF	Ou front montant
ANDR	Et front descendant
ORR	Ou front descendant
XOR	Ou exclusif

XORN	Ou pas exclusif
XORF	Ou front montant exclusif
XORR	Ou front descendant exclusif
ST	Range le résultat
STN	Range le complément
MPS	Stock (empile) le résultat booléen (pour une utilisation ultérieure)
MPP	Destock (dépile) un résultat
MRD	Lit la dernière valeur stockée sans la dépiler

VALEUR « VRAI » ET « FAUX »

Les valeurs "vrai" ou "faux" peuvent être utilisés dans des équations booléennes, en général pour commencer une équation.

TRUE	toujours Vrai
FALSE	toujours Faux

Exemple **LD TRUE**
[%MW2:=%MW10/5]

INSTRUCTION SUR BLOCS FONCTIONS FB PREDEFINIS

(voir la description du fonctionnement des blocs fonctions au chapitre LADDER)

Instructions d'actions sur les blocs fonctions

FONCTION	INSTRUCTIONS	ROLE
Temporisateur %TM	IN %Tmi	Démarre la tempo
Temporisateur %T	Ne peut être démarré en IL	
Monostable %MN	S %Mni	Démarre une impulsion
Compteur %C	R %Ci S %Ci CU %Ci CD %Ci	Raz compteur Préselectionne Incrémente Décrémente
Registre %R	R %Ri I %Ri O %Ri	Vide la pile Empile Dépile
Programmateur %DR	R %Dri U %Dri	Init au pas zéro Pas suivant

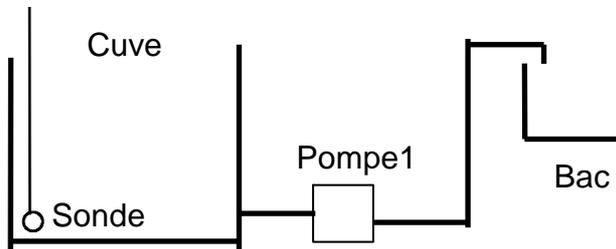
OPERATION SUR MOTS ET COMPARAISONS

Les opérations sur mots et comparaisons s'écrivent entre crochets "[....]"

Exemple
LD [%MW10 < 5]
[%MW50:=%MW10-5*(SQRT(%MW20))/%MW30]

EXERCICE 5.LANGAGE LISTE D'INSTRUCTIONS

Remplissage d'un bac d'échantillonnage



La mesure de niveau est lue dans le mot %MW10, lorsqu'on appuie sur le Bp_bac, la pompe fonctionne pendant une durée proportionnelle au niveau dans la cuve, Temps de fonctionnement = Niveau * 3.

- a - Définissez les entrées, sorties et variables à utiliser
- b - Ecrivez le programme en langage liste d'instructions dans le SR 2
- c - N'oubliez pas d'appeler le SR 2 depuis le MAIN
- d - Testez

LANGAGE LITTERAL STRUCTURE

Le langage littéral structuré se programme sous forme de phrases, comportant éventuellement un commentaire et une étiquette, suivant le même principe que le langage liste d'instructions.

Le littéral structuré permet comme son nom l'indique de structuré la programmation

STRUCTURE DE PROGRAMME

STRUCTURE INCONDITIONNELLE

Une suite d'actions séparées par des ";"

```
<Action>;<Action>;<Action>;  
<Action>;  
<Action>;
```

Une action fini toujours par un ";"

STRUCTURES CONDITIONNELLE

```
IF <condition> THEN  
    <programme>  
ELSE  
    <programme>  
END_IF;
```

```
IF <condition> THEN  
    <programme>  
ELSEIF <condition> THEN  
    <programme>  
ELSE  
    <Programme>  
END_IF;  
( Le nombre de ELSEIF est illimité )
```

STRUCTURES ITÉRATIVES

```
WHILE <condition> DO  
    <programme>  
END_WHILE;  
  
REPEAT  
    <programme>  
UNTIL <condition> END_REPEAT;
```

!!! Attention : l'UC ne fait rien d'autre pendant le programme

STRUCTURE REPETITIVE

```
FOR <indice>:=<valeur départ> TO <Valeur arrivé> DO  
    <programme>  
END_FOR;
```

LE MOT CLE "EXIT"

Le mot clé "EXIT" permet de sortir prématurément d'un boucle

EXEMPLES DE PROGRAMME EN LITTERAL STRUCTURE

```
! %M11:=%M10 AND (%I1.2 OR %I1.3);      (* structure inconditionnelle *)
```

```
! FOR %MW50:=0 TO 20 DO  
    IF (%MW100[%MW50]> 10) THEN  
        SET %M20;  
        EXIT; (* Quitte la boucle FOR *)  
    END_IF;  
END_FOR;
```

```
! REPEAT  
    INC %MW4;  
    SET %M10[%MW4];  
UNTIL (%MW4 >=10) END_REPEAT;
```

```
! L50 :  
IF(%MW10<>0)THEN  
    WHILE NOT %M0[%MW10] and (%MW10<16)DO  
        IF(%MW10 REM 3=0)THEN  
            SET %M0[%MW10]; (* mise à 1 des bits modulo 3 *)  
        END_IF;  
        INC %MW10;  
    END_WHILE;  
END_IF;
```

EXERCICE6. LITERAL STRUCTURE

Consignateur d'états

Mot d'état	%MW20	Mot_etat, chaque bit représente un état TOR de l'installation
Mot de référence	%MW21	Mot_reference, Chaque bit représente les états TOR de la scrutation précédente
Mot de discordance	%MW22	Mot_discordance, Chaque bit à « 1 » représente une discordance entre « Mot d'état » et « Mot référence »

Le mot d'états contient les états à consigner

Le mot de référence contient les états antérieurs

Le mot de discordance est obtenu en réalisant un ou exclusif entre les deux mots précédents et contient des bits à 1 là où il y a une différence.

Numéro d'état	%MW23	Numero_etat, ce mot recevra le numéro du bit en discordance Le numéro d'état correspond au numéro du bit à 1 dans le mot de discordance
Pile des états	%R0	Pile_etat, sauvegarde des défauts (historique) à chaque changement d'état le numéro d'état est empilé en FIFO

a - Définir les variables dans l'éditeur de variables

b - Traduire l'algorithme suivant en langage littéral structuré

Mot_discordance = OU exclusif entre mot_etat et mot_référence
SI Mot_discordance <> 0 ALORS (c'est qu'il y a un changement d'état)

```

    REPETER
        Numero_etat = 0    ( initialiser le numéro à 0 )
        TANT QUE bit 0 du mot discordance = 0 et numero_etat < 16 FAIRE
            Décaler mot_discordance de 1 bits vers la droite
            Ajouter 1 à Numero_etat
        FINTANTQUE
        Empiler Numero_etat
        Mettre à 0 le bit 0 du mot_discordance
    JUSQU'A Mot_discordance = 0
FINSI
Mot_référence := Mot_etat    ( Actualise l'état antérieur )
    
```

1/ Mettre Numero_etat dans le mot d'entrée du registre %Ri.I
2/ Empiler (PUT %Ri)

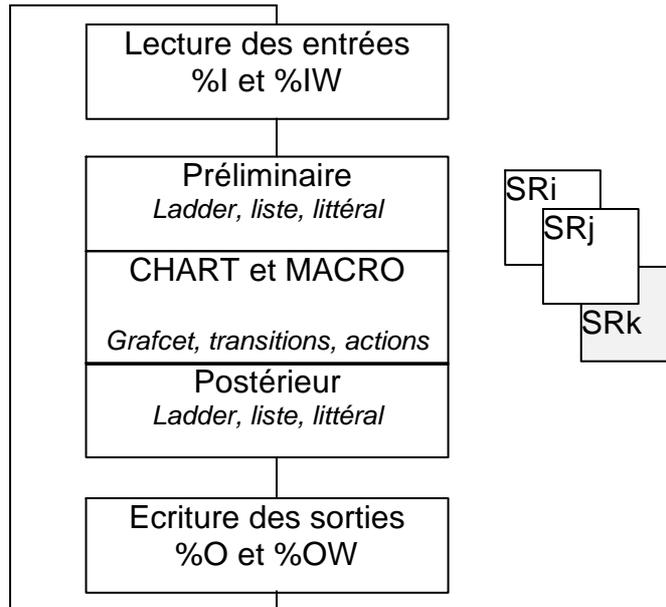
c - Programmer dans le SR3

d - Tester le programme

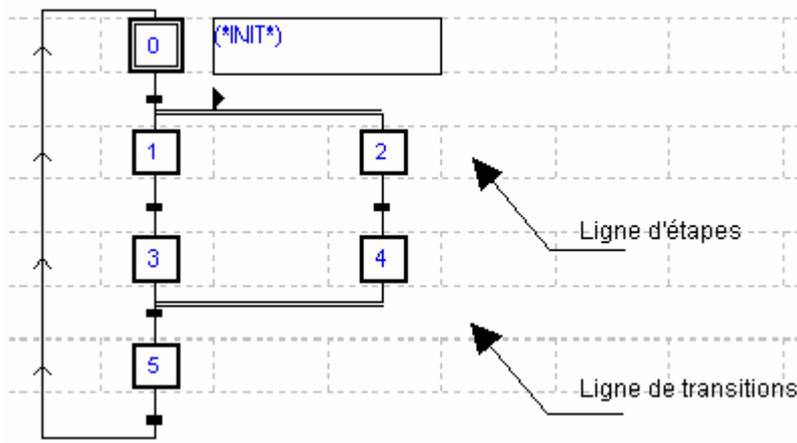
LE LANGAGE GRAFCET

Le grafcet se programme uniquement dans la tâche maître, à condition de l'avoir déclaré lors de la création de la station, **ce choix n'est pas modifiable**.

Structure de la tâche maître avec grafcet



Le(s) grafcet(s) se programme(nt) dans le module CHART sur 8 pages, chaque page contient 14 lignes et 11 colonnes définissant 154 cellules, chaque cellule pouvant recevoir soit une étape soit une transition.



LIMITES DU CHART

	TSX 37-10	TSX 37-2x	TSX 57-xx
Actions à l'activation	oui	oui	oui
Actions continues	oui	oui	oui
Actions à la désactivation	oui	oui	oui
Plusieurs éléments programme par action	non	oui	oui
Nombre maximum d'étapes	96	128	128
Nombre maximum de transitions	192	256	256

Seule la tâche MAST peut être programmée en Grafcet.
Les repères associés aux étapes ne sont pas symbolisables.

OBJETS ASSOCIES AU GRAFCET

%Xi Bits d'étapes à tester dans la partie postérieur pour réaliser les actions, ces bits peuvent être mis à "1" ou à "0" à l'aide des instruction **SET** ou **RESET** dans la partie préliminaire pour positionner le grafcet.

%Xi.T Mot de 16 bits représentant le temps d'activité d'une étape en 1/10em de secondes.

%S21 Bit système à positionner à "1" dans la partie préliminaire pour initialiser les grafcets, ce bit est remis automatiquement à "0".

%S22 Bit système à positionner à "1" dans la partie préliminaire pour mettre à 0 toutes les étapes de tous les grafcets, ce bit est remis automatiquement à "0".

%S23 Bit système, mis à 1 fige le grafcet, mis à "0" le grafcet repart.

%S26 Bit système mis à "1" par l'interpréteur grafcet, indique un dépassement des capacités grafcet, doit être remis à "0" par l'utilisateur.

PROGRAMMATION DES TRANSITIONS

1 - Double cliquez sur le **bouton droit** de la souris.

2 - Choisissez un langage LADDER, LISTE D'INSTRUCTIONS, LITTERAL STRUCTURE.

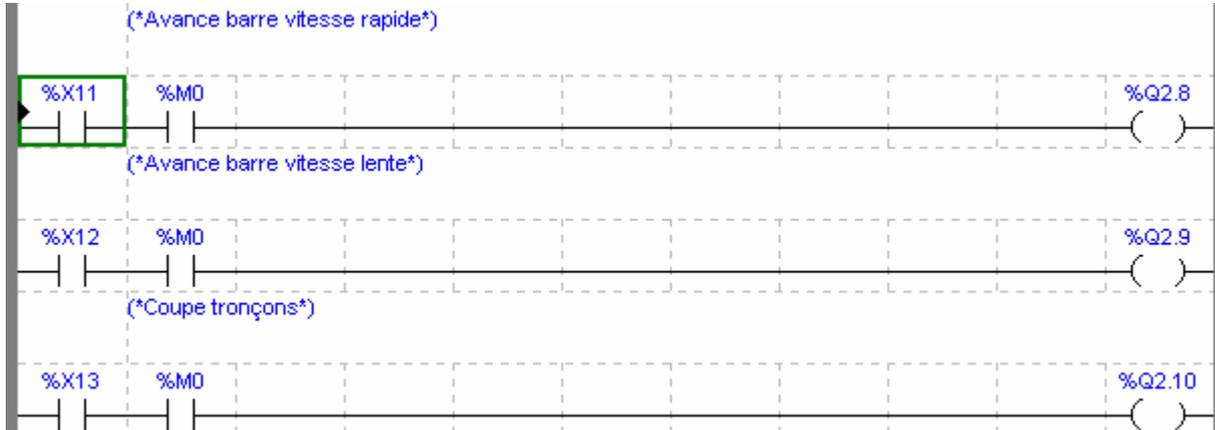
En LADDER la réceptivité est représenté par le symbole (**#**).

En Liste ou en littéral écrivez simplement l'équation de la transition.

PROGRAMMATION DES ACTIONS

Les actions sont programmées dans la partie postérieure en testant les bits d'étapes %Xi.

Exemple



Attention Si une action doit être exécuté sur plusieurs étapes, faites un OU entre les bits d'étapes.

PROGRAMMES ASSOCIES AUX ETAPES

A chaque étape peuvent être associés trois programmes.

Un programme exécuté à l'activation

Un programme exécuté à la désactivation

Un programme exécuté tant que l'étape est active (continu)

- 1 - Cliquez le **bouton droit** de la souris
- 2 - Choisissez le module (activation, continu, désactivation)



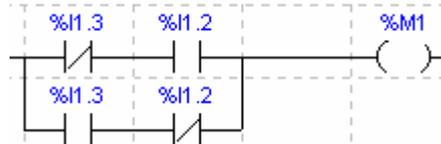
OBJETS ASSOCIES AUX ENTREES SORTIES

RAPPEL SUR LA SCRUTATION D'UNE TACHE

Lecture des cartes d'entrées
Mise à jour de la mémoire image d'entrées
%I %IW

**PROGRAMME
LADDER - LISTE - LITERAL + GRAFCET**

TESTS DES ENTREES



ECHANGES EXPLICITES

- Lecture des états des coupleurs
- Lecture des paramètres de réglage des coupleurs
- Commande des coupleurs
- Ecriture des paramètres de réglage des coupleurs
- Lecture des paramètres de configuration des coupleurs
- Ecriture des paramètres de configuration des coupleurs

- READ_STS**
- READ_PARAM**
- WRITE_CMD**
- WRITE_PARAM**
- SAVE_PARAM**
- RESTORE_PARAM**

ACTIONS SUR LES SORTIES



Ecriture des cartes de sorties
depuis la mémoire image des sorties
%Q %QW

OBJET A ECHANGE IMPLICITE

%Ix.i	état d'une voie TOR d'entrée
%Iwx.i	Mot d'entrée
%Qx.i	état d'une voie TOR de sortie
%QWx.i	Mot de sortie
%Ix.i.ERR	Défaut d'une voie
%Ix.MOD.ERR	Défaut module
%IWx.i:Xj	Bit d'un mot d'entrée
%QWx.i:Xj	Bit d'un mot de sortie

OBJET A ECHANGE EXPLICITE

Les échanges explicites se font à l'aide des fonctions **READ_STS**, **WRITE_CMD**, **READ_PARAM**, **WRITE_PARAM**.

Exemples	READ_STS %CH3.MOD	Lecture explicite des paramètres d'états du module 3
	READ_STS %CH4.1	Lecture explicite des paramètres d'états de la voie 1 du module 4.
	WRITE_CMD %CH3.MOD	Ecriture explicite des paramètres de commande du module 3
	WRITE_CMD %CH4.1	Ecriture explicite des paramètres de commande de la voie 1 du module 4.
	READ_PARAM %CH3.0	Lecture explicite des paramètres de réglage de la voie 0 du module 3.
	WRITE_PARAM %CH3.0	Ecriture explicite des paramètres de réglage de la voie 0 du module 3.

Les fonctions **READ** mettent à jour les zones mémoires %MWx.i.r OU %MWx.MOD.r
 x : Numéro de module i : Numéro de voie r : Numéro de registre

Les fonctions **WRITE** transfèrent les zones mémoires %MW vers les coupleurs.

OBJETS ASSOCIES AUX MODULES

<input checked="" type="checkbox"/> Paramètres		EIS	Adresse Module	1	<input type="checkbox"/> Zone de saisie
Repère	Type	Symbole		Commentaire	
%CH1.MOD	CH			Echange explicite avec module 1	▲
%I1.MOD.ERR	BOOL			Défaut globale module 1	▲
> %Mw1.MOD	WORD			Mot d'état N° 0 du module 1	▲
> %Mw1.MOD.1	WORD			Mot d'état N° 1 du module 1	
> %Mw1.MOD.2	WORD			Mot d'état N° 2 du module 1	
%CH1.0	CH			Echange explicite avec voie 0 du module 1	
%I1.0.ERR	BOOL			Défaut voie 0 module 1	
%I1.0	EBOOL			Etat de la voie 0	
> %Iw1.0	WORD			Mot 0 de la voie 0 d'entrée du module 1	
> %Qw1.0	WORD			mot 0 de la voie 0 de sortie du module 1	
> %Mw1.0	WORD			Mot d'état 0 de la voie 0 du module 1	
> %Mw1.0.1	WORD			Mot d'état 1 de la voie 0 du module 1	
> %Mw1.0.2	WORD			Mot d'état 2 de la voie 0 du module 1	
> %Kw1.0	WORD			Mot de configuration 0 de la voie 0 module 1	
> %Kw1.0.1	WORD			Mot de configuration 1 de la voie 0 module 1	
> %Kw1.0.2	WORD			Mot de configuration 2 de la voie 0 module 1	
> %Kw1.0.3	WORD			Mot de configuration 3 de la voie 0 module 1	
> %Kw1.0.4	WORD			Mot de configuration 4 de la voie 0 module 1	
> %Kw1.0.5	WORD			Mot de configuration 5 de la voie 0 module 1	
> %Kw1.0.6	WORD			Mot de configuration 6 de la voie 0 module 1	
> %Kw1.0.7	WORD			Mot de configuration 7 de la voie 0 module 1	
> %Kw1.0.8	WORD			Mot de configuration 8 de la voie 0 module 1	▼
%I1.1.ERR	BOOL			Défaut de la voie 1	▼
%I1.1	EBOOL			Etat de la voie 1	▼

VOIES ANALOGIQUES

Voir le chapitre configuration.

Les valeurs analogiques d'entrées sont accessible dans les mots
 $\%IW\ m.v$ m : Numéro de module, v : Numéro de voie

Les valeurs analogiques de sortie sont accessible dans les mots
 $\%QW\ m.v$ m : Numéro de module, v : Numéro de voie

EXEMPLE

Conversion en valeur flottante et mise à l'échelle d'une mesure analogique

Cette instruction peut s'écrire en littéral structuré ou dans une opération en LADDER

$\%MF50:=INT_TO_REAL(\%IW2.0)*0.025;$

MISE AU POINT DES ENTREES ANALOGIQUES

The screenshot shows the 'Mise au point' window for a TSX AEY 810 module. The window title is 'Mise au point' and it displays 'Désignation : 8E ANA. ISOLEES HN' and 'Version : 1.0'. There is a 'Déforçage global' button and status indicators for RUN (green), ERR (grey), IO (red), and DIAG... (red).

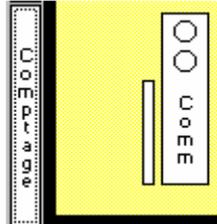
Voie	Symbole	F	ERR	<	Valeur	>	Filtre	A
0	Mesure_01		DIAG...		2121	0		0
1			DIAG...		-1250	0		0
2			DIAG...		1	0		0
3			DIAG...		0	0		0
4			DIAG...		5	0		0
5			DIAG...		6	0		0
6			DIAG...		0	0		0
7			DIAG...		1	0		0

The 'Réglages Voie 0' dialog box on the right has the following settings:

- Affichage:** Gamme 4..20mA, 0 à 10000
- Forçage:** 500, Forcer (checked), Déforcer
- Filtrage:** 0, Valider
- Alignement:** Valeur cible: 0, Offset: 0, Valider, Reset

FONCTION DE COMPTAGE INTEGREE

CONFIGURATION



Les compteurs / décompteurs intégrés aux bases de TSX MICRO se trouvent dans le **module 00, voie 11** pour le 1^{er} compteur, **voie 12** pour le 2^{em} compteur.

Chaque voie dispose de 4 entrées

IA	Comptage ou décomptage	Capteur ou codeur
IB	Décomptage ou sens de rotation	Codeur et/ou sens
IZ	RAZ	Capteur ou codeur
IPRES	Préselection	Capteur

Les voies **IA, IB, IZ** peuvent être alimentées en **5 V** ou en **10..30V** selon la câblage, la voie **IPRES** ne peut être alimentée qu'en **24 V**.

COMPTAGE

Sur l'entrée **IA**, la valeur courante peut être lue dans **%ID0.11.0** pour le 1^{er} compteur et **%ID0.12.0** pour le 2^{em} compteur.

EXEMPLE DES COMPTEURS INTEGRES AUX UC TSX MICRO

Les entrées IA, IB et IZ peuvent être alimentées en 24V ou en 5V, l'entrée IPres ne peut être alimentée qu'en 24V.

Compteur 0 voie 11
 ou
 compteur 1 voie 12

Choix de la fonction
 Comptage
 Décomptage
 Comptage/décomptage

Evénement
 EVT 0

Déclenche un événement sur franchissement de seuil (définis dans la partie réglage)

Choix du type d'entrées

Comptage sur l'entrée IA / décomptage sur l'entrée IB
 Comptage/décomptage sur IA, sens déterminé par IB
 Comptage/décomptage sur IA, sens déterminé par programme
 Codeur incrémental :
 IA et IB comptage décomptage par discrimination de sens de marche
 IZ = prise d'origine came courte (RAZ compteur)
 IPres = présélection sur front montant ou descendant

COMPTAGE SUR MODULE TOR

Symbole :		
Compteur :	Fonction :	Tâche :
Compteur 0	Comptage	MAST

Interface d'entrées

1 entrée IA

RAZ sur IReset

Front montant IReset

Fonctionnement sur franchissement consigne

Sans RAZ compteur

Avec RAZ compteur

COMPTAGE SUR CARTE CTY

TSX CTY 4A [RACK 0 POSITION 3]

Configuration

Désignation : MOD.COMPT. 4 VOIES 40KHZ

Symbole :

Compteur : Compteur 2 Fonction : Comptage/Décomptage Tâche : MAST

Interfaces d'entrée

Codeur incrémental

Contact statique

Contrôle de ligne Multiplication : par 1 par 4

Présélection sur IPres : Front montant IPres

Capture sur ICapt : Front montant ICapt

Evénement

EVT

Réarmement des sorties

Manuel Automatique

Mode de repj

RAZ Maintien

Symbole :

Compteur : Compteur 2 Fonction : Comptage/Décomptage

CH2 DIAG...

Validation

Active

IVal

Validation

Directe

Mesure courante

-3771

Mesure invalide

Sens : -

IA

IB

Comparaison

	Seuil 0	Seuil 1	Cons.H	Cons. B
	0	0	0	0
Mesure/	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Capture/	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Présélection

0

Effectuée

IPres

Validation

Directe

RAZ

Capture

0

Effectuée

ICapt

Validation

Directe

RAZ

Bascules de sortie

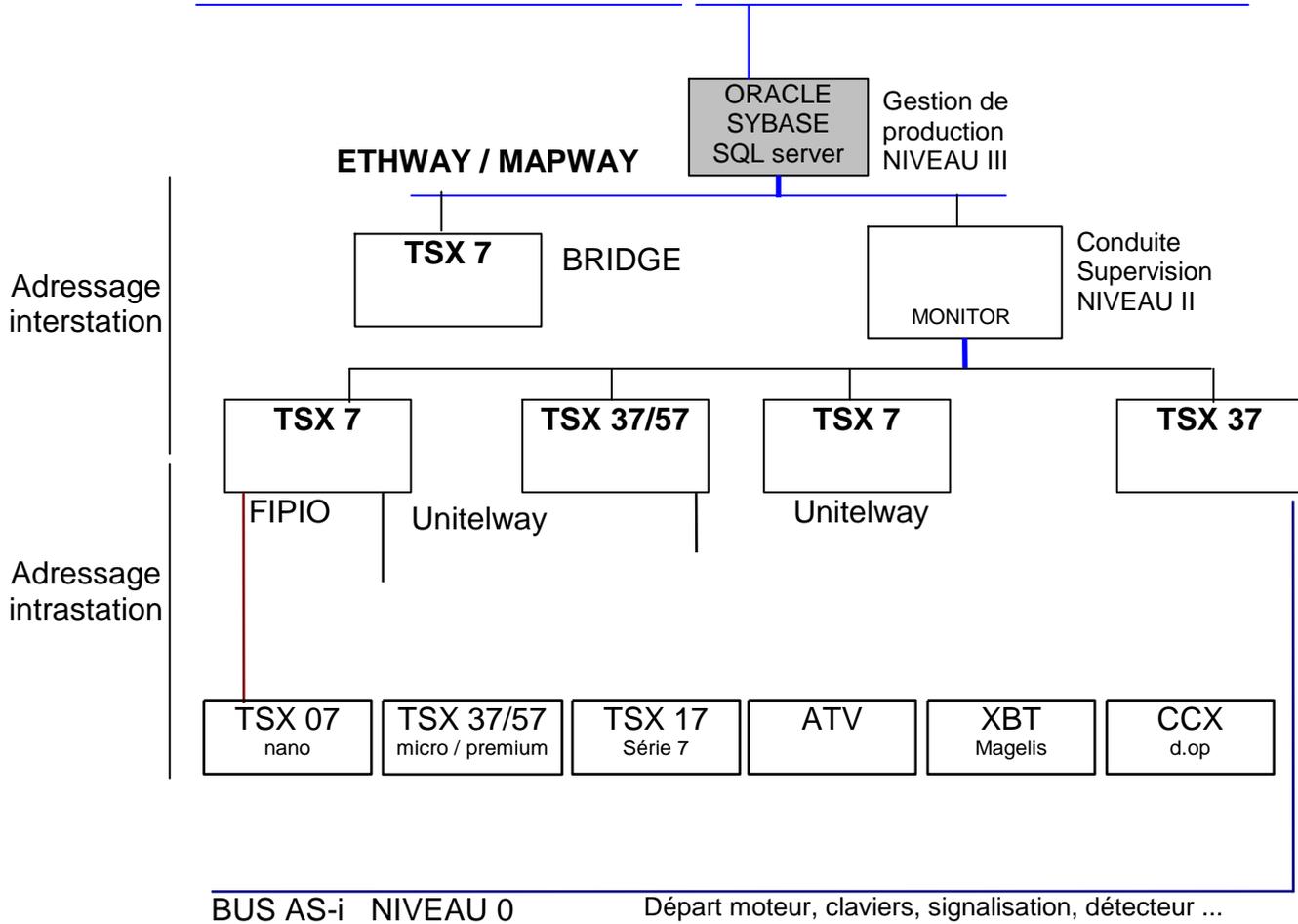
	B0	B1
Etat	<input type="radio"/>	<input type="radio"/>
Mise à 1	<input type="checkbox"/>	<input type="checkbox"/>
Mise à 0	<input type="checkbox"/>	<input type="checkbox"/>

Sortjes

	Q0	Q1
Etat	<input type="radio"/>	<input type="radio"/>
Validation auto	<input type="checkbox"/>	<input type="checkbox"/>
Mode auto	<input type="checkbox"/>	<input type="checkbox"/>
Cde manuelle	<input type="checkbox"/>	<input type="checkbox"/>
Réarmement Q0 et Q1	<input type="checkbox"/>	

COMMUNICATIONS

Réseau usine NIVEAU IV (ETHERNET, TOKEN RING ... , Routages X25)



L'adressage interstation se décompose en 2 niveau

NIVEAU	RESEAU
NIVEAU	STATION

ADRESSAGE INTERSTATIONS

L'adressage interstation identifie les stations sur les réseaux :

FIPWAY
TELWAY
ETHWAY
MAPWAY

Un adresse intersation est constituée du numéro de réseau et numéro de station.

{Réseau.Station}

{Réseau.ALL}

Diffusion vers toutes les stations

ADRESSAGE INTRASTATION

L'adressage intrastation identifie les entités appartenant à la station

SYS

Systeme (UNI-TE)

MOD ou module.voie.entité

Coupleurs

module.voie.N° esclave

Adresses sur bus

\mod.voie.pt\mod.voie.entité

Entrées sorties déportées (FIPIO)

APP ou APP.TXT (vers bloc txt)

Programme application

Exemple

ADR#{2.4}0.1.7 L'adresse unitelway 7 sur la voie 1 du coupleur intégré (module 0) de la station 4 sur le réseau 2

FONCTIONS DE COMMUNICATION

Description	Fonction	Commentaires
Lecture d'objets standard	"READ_VAR"	UTW, FIP, MODBUS, etc.
Ecriture d'objets standard	"WRITE_VAR"	UTW, FIP, MODBUS, ect.
Emission de requêtes UNI-TE	"SEND_REQ"	UNI-TE ex: 16#14 écriture d'un mot 16#25 Mise en RUN
Echange de données de type texte	"DATA_EXCH"	Appli <-> Appli
Ecriture de chaîne de caractères	"PRINT_CHAR"	Chaîne de caractères
Lecture de chaîne de caractères	"INPUT_CHAR"	Chaîne de caractères
Emission d'une chaîne de caractères (Il faut configurer la voie avec un caractère d'arrêt)	"OUT_IN_CHAR"	Chaîne de caractères, peut contenir des caractères spéciaux ex : \$N = CR+LF
Décalage à droite d'un octet d'un tableau	"ROR1_ARB"	Repositionne les octets après certaines requêtes
Emission d'un Télégramme	"SEND_TLG"	Uniquement sur FIPWAY pour les 16 premières stations
Réception d'un Télégramme	"RCV_TLG"	Uniquement sur FIPWAY pour les 16 premières stations
Arrêt d'un échange	"CANCEL"	Faire référence au N° d'échange contenu dans le 1 ^{er} mot des paramètres de gestion

Toutes les fonctions de communication ont un paramètre de gestion constitué de 4 mots

Numéro du mot	Octet poids fort	Octet poids faible
%MWi	Numéro d'échange mis à jour par le système TXTi	Bit d'activité à "1" pendant l'échange TXTi,D
%MWi+1	Compte-rendu de l'opération Spécifique à la fonction ex: CR de requête TXTi,V	Compte-rendu de communication commun à toutes les fonctions TXTi,S
%MWi+2	Time-out x 100 ms OFB,timeout	
%MWi+3	Longueur Nombre d'octets, maxi 254 TXTi,L	

Toujours mettre à jour le paramètre longueur avant chaque lancement d'une fonction de communication

EXEMPLE

(* écriture des mots %MW10 à %MW19, la table de gestion est en %MW20:4 *)
(* le destinataire est sur le réseau 2, station 4, module 0, voie 0, équipement 6*)

```
%MW22:=50;      (* Time Out 5 sec. *)
%MW23:=20       (* 20 Octets à transmettre *)
WRITE_VAR(ADR#{2.4}0.0.6,'%MW',10,20,%MW0:10,%MW20:4);
```

COMPTES RENDUS DE COMMUNICATION

Valeur	Compte-rendu communication: octet poids faible
16#00	Echange correct
16#01	Arrêt de l'échange sur time-out
16#02	Arrêt de l'échange sur demande utilisateur (CANCEL)
16#03	Format d'adresse incorrect
16#04	Adresse destinataire incorrecte
16#05	Paramètres de gestion incorrects
16#06	Paramètres spécifiques incorrects
16#07	Problème d'émission vers le destinataire
16#08	Réservé
16#09	Taille du buffer de réception insuffisante
16#0A	Taille du buffer d'émission insuffisante
16#0B	Absence de ressource système
16#0C	Numéro d'échange incorrect
16#0D	Aucun télégramme reçu
16#0E	Longueur incorrecte
16#0F	Service télégramme non configuré
16#10	Coupleur réseau absent
16#FF	Message refusé

Si le compte rendu est message refusé 16#FF, voir les détails dans le compte rendu d'opération

Compte-rendu d'opération Octet de poids fort

Valeurs du compte-rendu sur message refusé :

Code refus Valeur	Signification
16#01	Manque de ressource processeur
16#02	Manque de ressource ligne
16#04	Erreur ligne
16#05	Erreur de longueur
16#06	Voie de communication en défaut
16#07	Erreur d'adressage
16#08	Erreur applicatif
16#0B	Absence de ressource système
16#0D	Destinataire absent
16#0F	Problème de routage intra-station
16#11	Format d'adresse non géré
16#12	Manque de ressource destinataire

AIDE A LA SAISIE D'UNE FONCTION DE COMMUNICATION

EN LADDER SELECTIONNEZ 

EN LISTE D'INSTRUCTION OU EN LITTERAL STRUCTURE SELECTIONNEZ
SERVICE → SAISIR L'APPEL D'UNE FONCTION

Appel de fonction

Informations Fonctions : Paramètres

Famille	V.Bib	V.App	Nom	Commentaire
Chaînes de caractères	1.01	-	ROR1_ARB	Décalage de 1 octet vers la droite d'u>
Commande de Mouvement	1.00	-	SEND_REQ	Emission/Réception de requêtes UNI-TE
Communication	1.10	1.10	SEND_TLG	Emission d'un télégramme
Conversions numériques	1.01	-	WRITE_VAR	Ecriture d'objets standards

Format d'appel

Paramètres de la PROCÉDURE :

Nom	Type	Nature	Commentaire	Zone de saisie
ADR	AR_W	IN	Adresse: ADR#{r.s}] m.v.e ou SYS	
OBJ	STRING	IN	Objet à écrire: '%MW', '%M', '%KW',...	
NUM	DWORD	IN	Adresse du premier objet à écrire	

Visualisation de l'appel

WRITE_VAR ()

OK Annuler **Détail...**

READ_VAR

Paramètres

Adresse :

Type d'objet à lire :

Adresse du premier objet à lire :

Nombre d'objets consécutifs à lire :

Zone de réception : :

Compte rendu : :

Types acceptés : %KWi:n, %MWi:n, (n=6)
Adresse immédiate (ADR#)

OK Annuler

Saisie de
l'adresse

SAISIE DE L'ADRESSE

Réseau 2, Station 4
Module 0, Voie 100 (intégré TSX 7)
Esclave 6

The screenshot shows the 'Aide à la saisie de l'adresse' dialog box. The 'Mode' is set to 'Distant'. The 'Adresse générée' field contains 'ADR#{2.4}0.100.6', which is circled in red. Under 'Niveau réseau', 'Choix du Réseau' is set to '2' and 'Choix de la station' is set to '4' with 'Type' as 'TSX47-107'. Under 'Niveau station', 'Station destinataire' is 'TSX47-107'. In the 'Atteindre' section, 'Module' is checked. In the 'Module' section, 'UC' is checked, 'Rack' is '0', and 'Module' is '0'. 'Voie 0' and 'Voie 1' are unselected. In the 'Type' section, 'Maître' is selected. Under 'Niveau équipement', 'Protocole' is 'UNI_TELWAY'. In the 'Type' section, 'Esclave' is selected. 'Diffusion' is unselected and 'Esclave N°' is '6'. 'OK' and 'Annuler' buttons are at the bottom.

Diffusion sur le réseau

The screenshot shows the 'Aide à la saisie de l'adresse' dialog box. The 'Mode' is set to 'Distant'. The 'Adresse générée' field contains 'ADR#{2.ALL}SYS'. Under 'Niveau réseau', 'Choix du Réseau' is set to '2' and 'Choix de la station' is set to '4' with 'Type' as 'Diffusion'. The 'Type' dropdown is highlighted in blue.

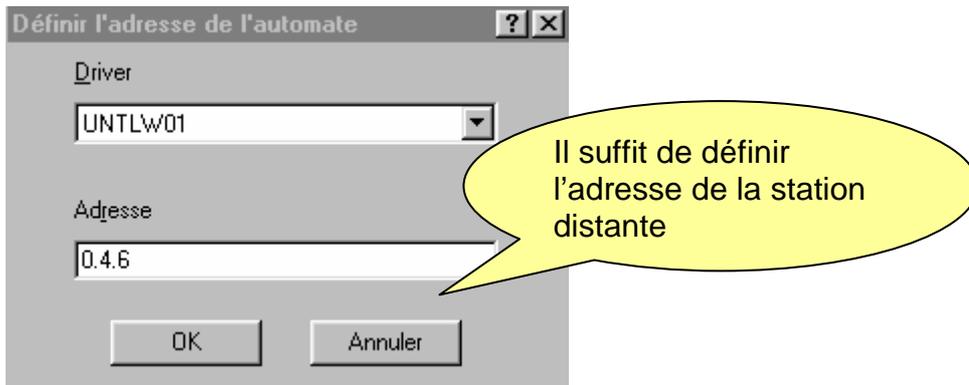
DIAGNOSTIQUE D'UNE LIAISON

On peut visualiser l'état d'une voie en ligne, exemple d'une voie unitelway maître

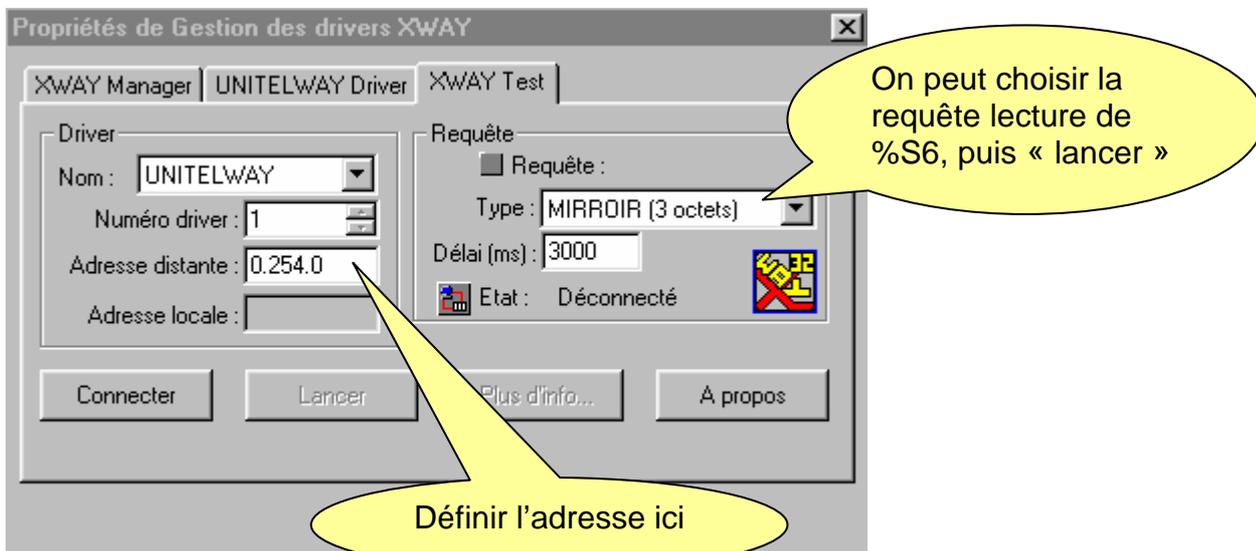
The screenshot displays the 'TSX 57102 [RACK 0 POSITION 0]' diagnostic window. At the top, it shows 'Mise au point' and 'Désignation : PROCESSEUR TSX P 57102 Version : 3.5'. Below this, 'VOIE 1' is set to 'TSX SCP 114 CARTE PCMCIA RS485 MP' with a 'LIAISON UNI-TELWAY' and 'MAST' configuration. A 'DIAG...' button is visible. The 'Type' section is set to 'Maître' and includes 'Compteurs Erreurs Locaux' with four counters (Messages émis non acquittés, Messages émis refusés, Messages reçus non acquittés, Messages reçus refusés) and an 'R.A.Z. Compteurs' button. The 'Test Voie' section has a list of addresses (5, 6, 7, 8) with '6' selected, and buttons for 'Identification', 'Compteurs', 'R.A.Z. Compteurs', 'Saisie Requête', and 'Miroir'. A 'Réponse Requête' field shows '..0.TSX 3722...0....' and radio buttons for 'Ascii' and 'Hexa'. The 'Esclave' section is currently empty.

Les adresses des esclaves qui répondent

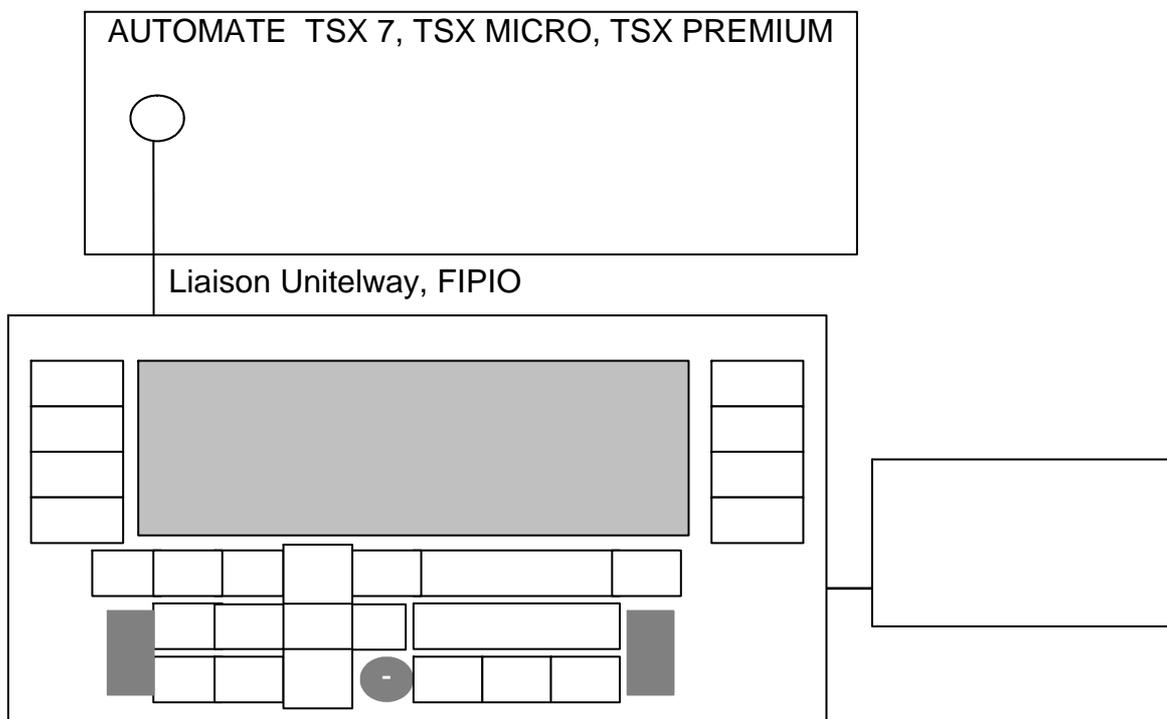
MISE EN LIGNE AVEC UNE STATION



On peut tester la liaison avec XWAY DRIVER MANAGER



DIALOGUE OPERATEUR CCX 17



AFFICHAGE DE TEXTES

300 Messages d'états

300 Messages d'alarmes regroupés dans 50 groupes maxi

SAISIES DE VALEURS

SORTIE T.O.R. ET LEDS

COMMUNICATION

Uni-telway

Fipio

Imprimante

CONCEPTION

Avec le logiciel PL7-MMI sous OS2 ou sous WINDOWS

Avec des fonctions automate

ECHANGES AVEC L'AUTOMATE

Par boîte aux lettres

échanges automatiques

Par fonction PL7

FONCTIONS DOP

PAS D'APPLICATION DANS LE CCX17

SEND_MSG	Envoi un message contenu dans la mémoire API avec ou sans variable
ASK_MSG	Envoi un message contenu dans la mémoire API , le CCX17 attend une réponse de l'utilisateur, la réponse est rangée dans une zone de mots %MW
GET_MSG	Identique à ASK_MSG, mais la saisie n'est pas obligatoire.
SEND_ALARM	Envoi un message d'alarme qui doit être acquitté par l'opérateur.

APPLICATION CCX17 CREE PAR LE LOGICIEL MMI17

DISPLAY_MSG	Affiche un message contenu dans le CCX17
DISPLAY_GRP	Affiche un groupe de message contenu dans le CCX17
ASK_VALUE	Affiche un message contenu dans le CCX17, le CCX17 attend une réponse de l'opérateur
GET_VALUE	Identique à ASK_VALUE mais la réponse n'est pas obligatoire.
DISPLAY_ALRM	Affiche un message d'alarme contenu dans le CCX17
PID_MMI	Affiche la face avant d'un régulateur PID intégré

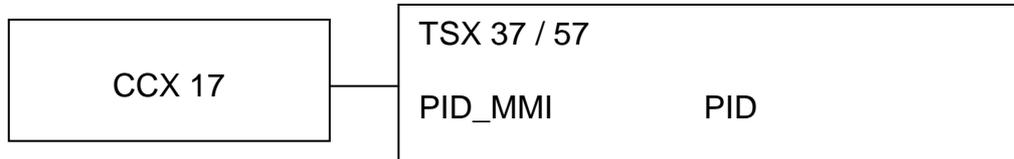
PARAMETRAGE ET CONTROLE DU CCX17

CONTROL_LED	Commande des leds et du relais
ASSIGN_KEYS	Affectations des touches du CCX17
PANEL_CMD	Commandes du CCX17, Effacement de l'écran, effacement d'une ligne, impression, acquittement d'une alarme depuis l'API

La fonction PID_MMI

Cette fonction permet d'afficher la face avant d'un régulateur sur le CCX 17, le paramètre EN permet de valider l'affichage

La fonction PID_MMI établit un dialogue entre le TSX 37/57 et un CCX 17.



Les fonctions PID_MMI sont exécutées à chaque cycle (appel non conditionné). Il y a un seul appel à la fonction PID_MMI pour gérer les PID de l'application. Il y a un appel de la fonction PID_MMI par CCX 17 connecté à l'automate.

La détection des PID de l'application par la fonction PID_MMI est automatique, y compris dans le cas d'ajout ou retrait de PID en mode RUN.

Le repérage du correcteur PID désiré est réalisé par le paramètre TAG de la fonction PID. Sa sélection dépend de la valeur du paramètre DEVAL_MMI de la fonction. Seuls sont pris en compte, par la fonction PID_MMI, les PID dont le paramètre DEVAL_MMI est = 0.

Limites

Le nombre maximum de PID exploités par les CCX est de 9, quel que soit le nombre de CCX connectés. Il y a pas de limitation du nombre de PID dans l'application.

PARAMETRES DES FONCTIONS DOP

ADRESSE DU CCX17

Le CCX17 peut être sur un bus unitelway ou FIPIO, il est adressable depuis l'automate local (adresse intrastation) ou depuis une station en réseau (adresse interstation)

sur bus Unitelway

ADR#{<Réseau>.<Station>}<rack.module>.<voie>.<adresse unitelway>

sur bus FIPIO

ADR#{<Réseau>.<Station>}<rack.module>.<voie>.<point de connexion>**\SYS**,

Remarque : pour un adressage local, Réseau et station sont facultatifs.

L'adresse peut être contenu dans une zone de mots (6 mots)

	Poids fort	Poids faible
%MWi / %KWi	6 (UNITELWAY) 7 (FIPIO)	0
%MWi+1 / %KWi+1	station (254 en local)	réseau (0 en local)
%MWi+2 / %KWi+2	module	rack
%MWi+3 / %KWi+3	@ Unitelway	Voie
	Point de connexion pour FIPIO	
%MWi+4 / %KWi+4	0	FIPIO = 254 = 16#FE Unitelway = 0
%MWi+5 / %KWi+5	0	0

Remarque : L'adressage par une zone de mot, n'est intéressante que s'il y a plusieurs CCX17 sur un même bus, sinon utilisez plutôt un adressage immédiat.

DONNEES A EMMETRE AU CCX17

Dans le cas d'une application créée par MMI17 la donnée à emmettre est contenu dans un mot, et correspond au numéro de message ou au numéro de groupe à afficher.

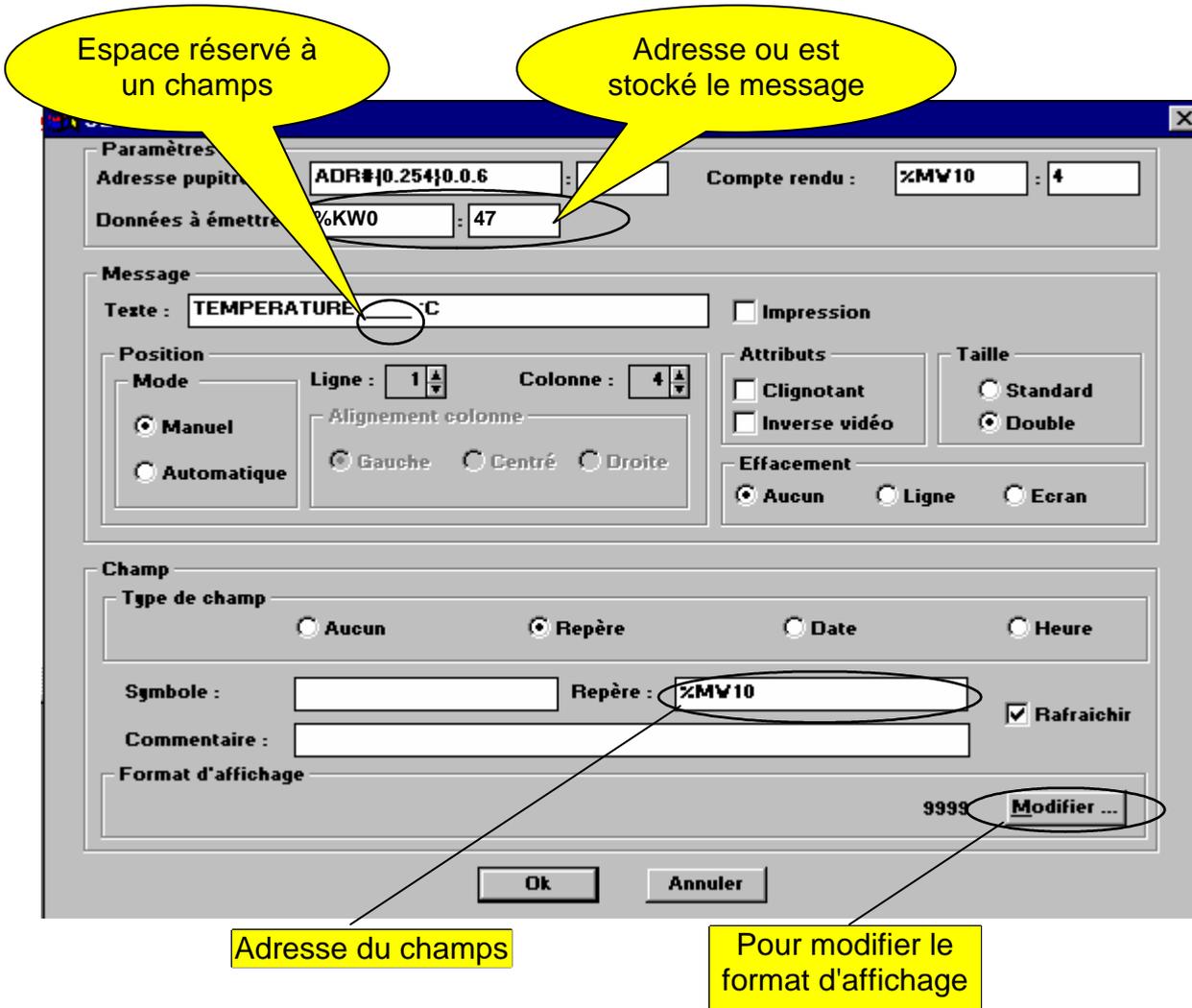
Dans le cas d'un message contenu dans la mémoire automate, la donnée doit respecter la structure suivante.

1^{er} MOT 16#CC17
 2^{em} MOT Numéro de commande

FONCTION	Valeur du mot
SEND_MSG	0
SEND_ALARM	0
ASK_MSG	33
GET_MSG	6
CONTROL_LEDS	Bit 0 à 3 Etat du voyant vert Bit 4 à 7 Etat du voyant jaune Bit 8 à 11 Etat du voyant rouge Bit 12 à 15 Etat du relais Etats 0 inchangé 1 éteint / ouvert 2 allumé / fermé F Clignotant (sauf relais)
ASSIGN_KEYS	N°du bit = N°de touche à configurer - 1 Les deux mots suivants contiennent les modes de marche des touches, chaque touche est codé sur 2 bits 00 Touche RAZ 01 Sur front 10 Bascule 11 aucune action Les 12 mots suivants contiennent les numéro de bits internes affectés à chaque touche (-1 si une touche n'est pas affectée)
PANEL_CMD	1: effacement écran , 2: effacement d'une ligne, 9: impression de l'historique des messages, 10: effacement de l'historique des messages, 11: impression de l'historique des alarmes, 13: effacement de l'historique des alarmes, 29: effacement d'une alarme (de 1 à 300 pour DISPLAY_ALARM) destinée au CCX17, 30: effacement d'une alarme (de 900 à 999 pour SEND_ALARM) destinée à l'automate,

Le mot suivant contient le numéro de ligne ou d'alarme.

AIDE A LA SAISIE DES FONCTIONS DOP



SEND_MSG(ADR#0.0.6,%KW0:47,%MW50:4);
La zone de mots %KW0:47 est automatique affectée.

Attention : le message occupe 27 mots plus la longueur du message, maxi 47 mots, mini 27 mots.

CONSEILS D'UTILISATION

- 1 Utilisez l'aide à la saisie des fonctions DOP
- 2 Réservez un espace de 50 mots entre chaque messages
- 3 Définissez l'adresse des données à émettre en mots constants

ERGONOMIE

SELECTIONNER - COPIER - COUPER - COLLER

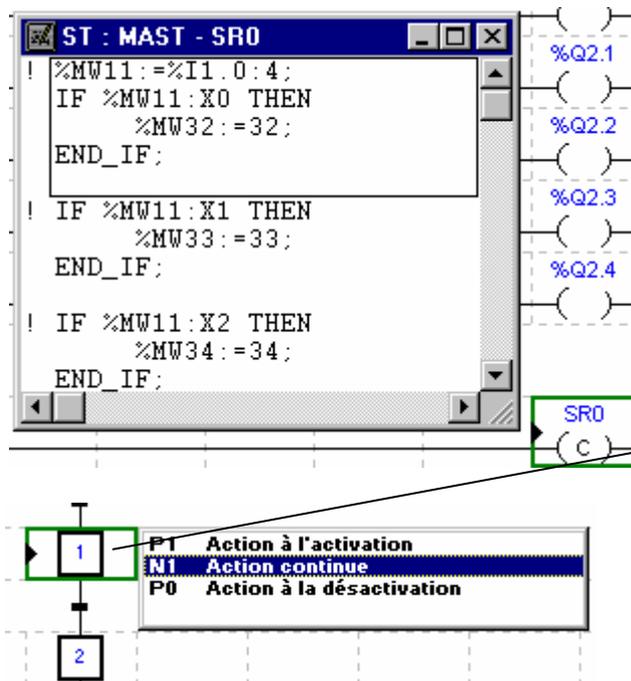
Cliquez sur la zone grise pour sélectionner le RUNG ou la phrase littérale, maintenez la touche SHIFT enfoncée et glissez pour sélectionner plusieurs RUNGS



Vous pouvez COPIER (CTRL+ C) - COUPER (CTRL + X) - COLLER (CTRL + V)

Astuce : Une fois copié vous pouvez le coller dans le bloc notes, vous pouvez alors modifier dans le bloc note et ensuite vous copiez le bloc note et le collez dans votre application, par exemple vous pouvez changer tous les %MW110 en %MW111.

ZOOM - INFORMATIONS - DETAILS



Double cliquez sur le bouton droit pour voir le sous programme

INITIALISER UNE TABLE DE SYMBOLES

Sélectionnez un ou plusieurs RUNG ou phrases en maintenant la touche SHIFT enfoncée.