INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

ESISAR

Ecole Supérieure d'Ingénieurs en Système Industriels Avancés Rhône-Alpes

BANC DE TRAVAUX PRATIQUES

TRAITEMENT DE SURFACE A TRANSLATEURS

utilisation du Logiciel

PL7-micro

ESISAR, 50 rue B. de laffemas - BP 54 - 26902 VALENCE CEDEX 9 - France
Tél. 04 75 75 94 00 - Fax 04 75 43 56 42

Introduction

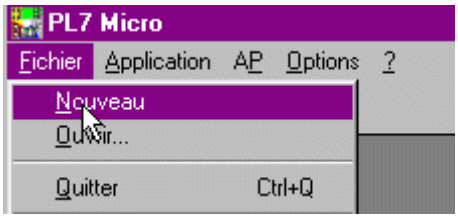
Ce manuel est fait pour permettre aux élèves d'utiliser le logiciel PL7-micro sur PC, et plus particulièrement pour leur permettre de réaliser des programmes pour l'automate TSX 37 connecté à la chaîne de traitement de surface à translateurs. Il ne se veut ni un cours de GRAFCET, ni une description du fonctionnement des automates TSX de télémécanique, ni un manuel complet d'utilisation du logiciel PL7-micro. Il doit permettre aux élèves de transcrire seul des GRAFCET décrivant le fonctionnement de l'automatisme en un fichier programme (.STX) compréhensible par l'automate.

Les renseignements qui y sont portés permettent la réalisation de programmes en GRAFCET, prêt à être testés. Le transfert des programmes dans l'automate, leur mise en route, et le déboguage ne sont pas traités ici, mais expliqués en salle de travaux pratiques. L'intérêt de ce manuel est de permettre aux élèves de travailler seul, préalablement à une séance de travaux pratiques au cours de laquelle il pourront tester leur GRAFCET sur la chaîne de traitement de surface, en présence d'un enseignant et sous sa responsabilité.

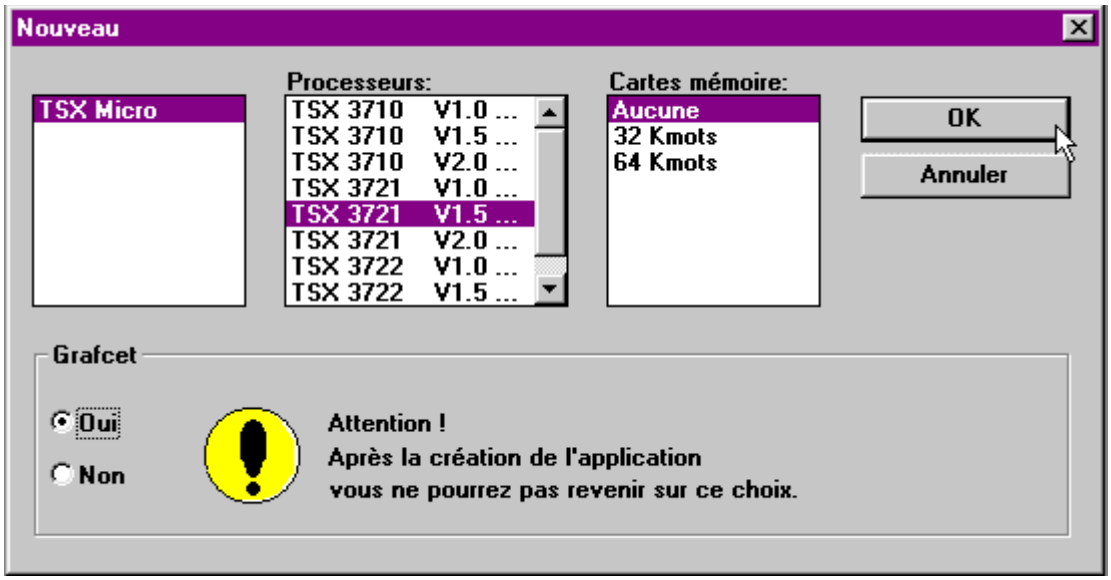
De nombreux renseignements contenus dans ce manuel sont extraits de l'aide du logiciel PL7-micro, et pour tous renseignements complémentaires les élèves sont invités à se référer à l'aide en ligne du logiciel.

Créer un nouveau GRAFCET pour la chaîne de traitement de surface

- Lancer PL7 micro (groupe de programme modicon télémécanique)
- Créer un nouveau fichier

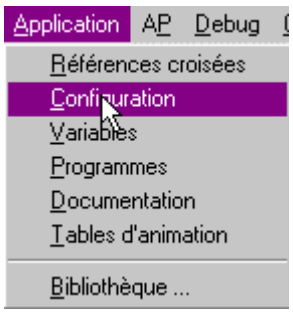


- choisir l'automate, et le langage de programmation .
  Nous disposons d'un TSX 3721 V1.5, sans carte mémoire.

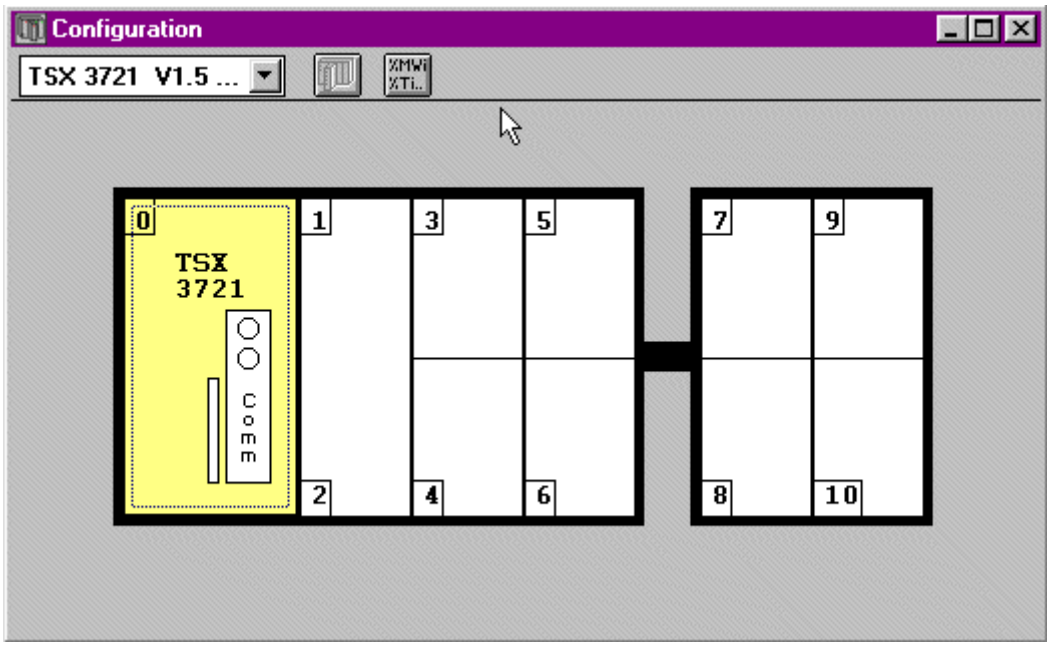


Configurer l'automate

il s'agit d'indiquer au logiciel quels modules d'Entrées Sorties sont connectés sur l'automate :

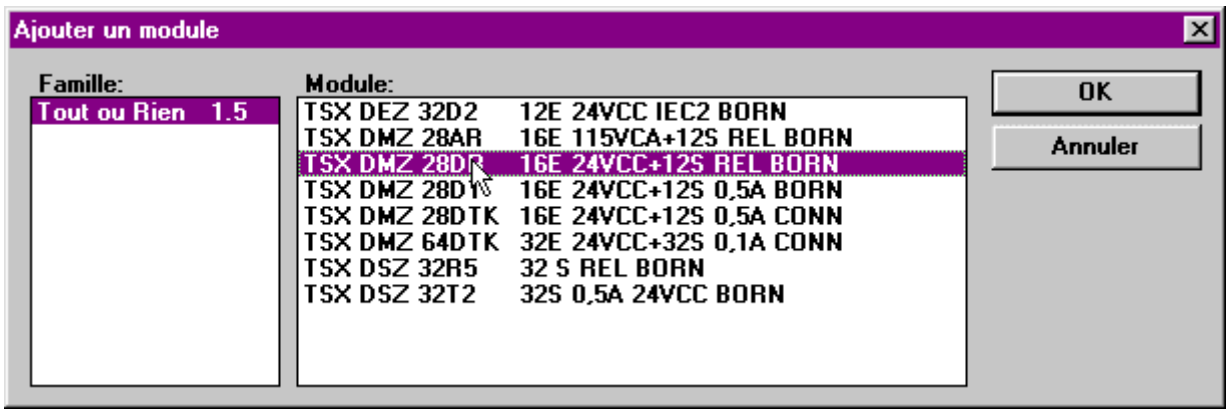


- Dans la fenêtre de configuration, double cliquer sur le bac à configurer,

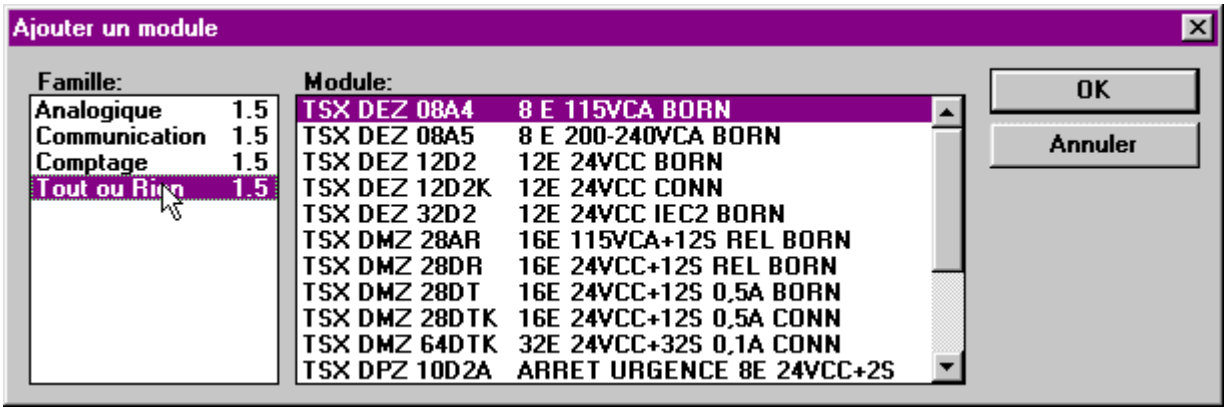


- Une liste de modules apparaît, choisir celui qui est physiquement connecté dans le rack. Pour le rack 1-2, seul quelques modules TOR apparaissent, pour les autres racks, vous devez choisir entre TOR, analogique, comptage, ou communication.

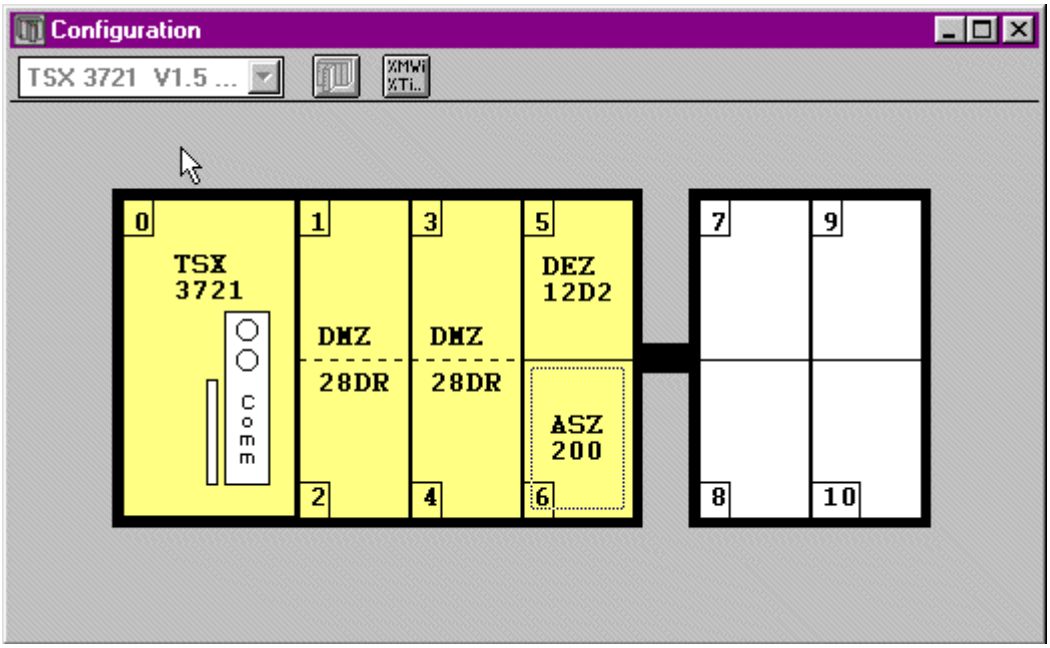
Modules du rack 1



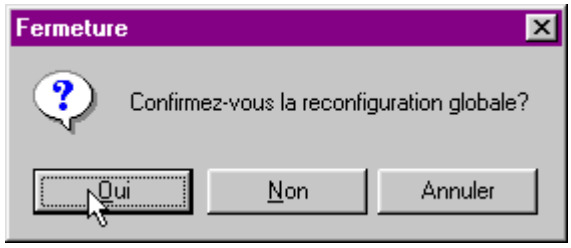
Modules des rack 3-4 à 9-10



Dans le cas de la chaîne de traitement de surface à translateurs, les modules sont les suivants :



- Quand vous avez fini, fermez la fenêtre. Le logiciel vous demande alors



Répondre OUI.

Suite à cette étape, le logiciel connaît les modules d'E/S disponibles dans l'automate. Il nous reste alors à programmer le système.

Les objets : entrées, sorties, mémoire, configuration

L'adressage des principaux objets bits et mots de modules d'entrées/sorties est défini par les caractères suivants :

%	I ,Q,M ou K	X, W, D ou F	x.i	r
Symbole	**Type d'objet** I = Entrée Q =Sortie M = information en lecture écriture K = Information de configuration	**Format** X = booléens W = mots D = double mots F = flottants	Position (x) et numéro de voie (i) du module TSX Micro	Rang r=0 à 127 ou ERR

Exemples :

I1.12 est l'entrée 12 du module format complet situé sur le rack 1

Q2.0 est la sortie 0 du module situé sur le rack 2 (ici un seul module 16 E, 8 S, en 1-2).

Adressage des bits : Valeur immédiate des bits: 0 ou 1 (False ou True)

Type	Adresse (ou valeur)	Nombre maxi	Accès en écriture (1)
Bits d'entrées	%Ixi ou %IXx.i	TSX 37-10 : 264 TSX 37-20 : 328 TSX 57-10 : 512 TSX 57-20 : 1024	oui
Bits de sorties	%Qx.i ou %QXx.i	TSX 37-10: 264 TSX 37-20: 328 TSX 57-10: 512 TSX 57-20: 1024	oui (2)
Bits internes	%Mi ou %MXi	TSX 37-10: 264 TSX 37-20: 328 TSX 57-10: 4096 TSX 57-20: 4096	oui
Bits système	%Si	128	
Bits de blocs fonction	ex : %TMi.Q %DRi.F		non
Bits extraits de mots	ex : %MW10:X5		selon type de mot

Bits système

Bit	Fonction	Etat initial	Gestion
%S0	1 = démarrage à froid (reprise secteur avec perte des données)	0	S ou U->S
%S1	1 = reprise à chaud (reprise secteur sans perte de données)	0	S ou U->S
%S4	Base de temps 10 ms	-	S
%S5	Base de temps 100 ms		
%S6	Base de temps 1 s		
%S7	Base de temps 1 mn		

Bit	Fonction	Etat initial	Gestion
%S8	Test du câblage(Utilisable sur automate non configuré)	1	U
%S9	1 =passage en repli des sorties	0	U
%S10	0 =défaut entrées/sorties	1	S
%S11	1 =débordement chien de garde	0	S
%S13	1 = premier cycle après mise en RUN	-	S
%S15	1 =défaut chaîne de caractères	0	S->U
%S16	0 =défaut E/S tâche	1	S->U
%S17	état du bit sorti, lors d'une opération de décalage	0	S->U
%S18	1 =débordement ou erreur arithmétique	0	S->U
%S19	1 =débordement de période tâche	0	S->U
%S20	1 =débordement d'index	0	S->U
%S21	1=initialisation Grafcet	0	S
%S22	1= désactivation Grafcet	0	S
%S23	1= Grafcet figé	0	S
%S26	1= Dépassement de capacité de la table des étapes actives avec STOP automate - (voir %SW20 et %SW21)	0	S
%S30	1 = activation de la tâche maître	1	U
%S31	1 = activation de la tâche rapide	1	U
%S38	1 = validation des événements	1	U
%S39	1 = saturation dans le traitement des événements	0	U
%S40 à %S47	0= Défaut des racks 0 à 7 (%S40 rack 0). défaut rack = OU logique des défauts modules du rack	1	S
%S49	1= réarmement des sorties statiques disjonctées	0	
%S50	1 = mise à l'heure de l'horodateur	0	U
%S59	1 = validation du réglage de la date courante	0	U
%S66	1 = voyant batterie toujours éteint	0	U
%S67	0 = pile carte mémoire en service	-	S
%S68	0 = pile de sauvegarde (processeur)en service	-	S
%S69	1 = validation du mode visualisation mémoire "WORD" sur les afficheurs	0	U
%S70	1 = rafraîchissement données sur bus AS-i ou liaison TSX Nano	0	U->S
%S73 (1)	Passage en mode protégé sur bus AS-i	0	U
%S74 (1)	Sauvegarde configuration présente sur bus AS-i	0	U
%S80	1 = mise à zéro des compteurs de messages	0	U->S
%S90	1 = rafraîchissement des mots communs	0	S->U
%S96 (1)	Validité de la sauvegarde du programme application	-	S
%S97 (1)	Validité de la sauvegarde des %MW	-	S
%S98 (1)	Déport du bouton poussoir du coupleur TSX SAZ 10	0	U
%S99 (1)	Déport du bouton poussoir du bloc de visualisation	0	U
%S100	Protocole sur prise terminal	-	S

S = géré par le système, U = géré par l'utilisateur, U->S = mis à 1 par l'utilisateur, remis à 0 par le système, S->U = mis à 1 par le système, remis à 0 par l'utilisateur.
(1) uniquement sur TSX 37 de version égale ou supérieure à V2.0

Les langages de programmation des automates TSX - télémécanique

Les automates TSX de télémécanique utilisent plusieurs langages différents :

Le langage à contact ou Ladder, langage de base des automates TSX, vous les trouverez sur tous les automates TSX.

Le langage PL7, langage proche du grafcet, il utilise le langage à contact pour programmer les actions associées aux étapes, et les réceptivités des transitions. Il n'est pas disponible sur les automates bas de gamme.

Le langage liste d'instruction est un langage de type « assembleur ».
Le langage littéral structuré est plus proche d'un langage évolué de programmation, mais n'est pas disponible pour tous les automates ou tous les logiciels. Nous n'en disposons pas ici.

Le langage à contact

Généralités

La programmation se fait à l'aide de programmes graphiques, représentant des « schémas électriques ».

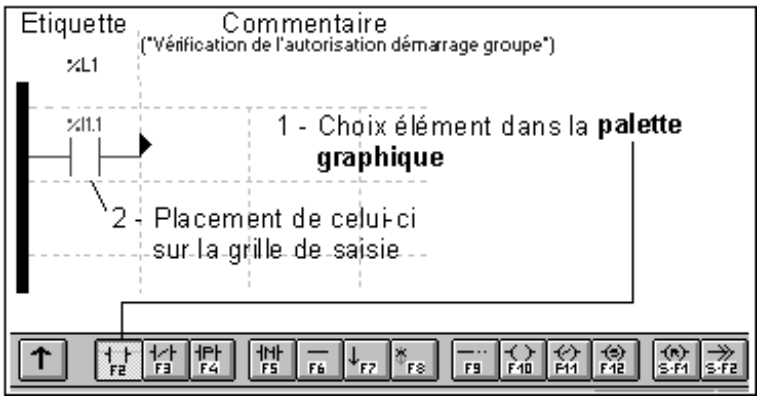
Un programme écrit en langage à contacts (LD) se compose d' une suite de réseaux exécutés séquentiellement par l'automate.
Dessiné entre deux barres de potentiel, un réseau est un ensemble d' éléments graphiques représentant :

. 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.

Ces éléments graphiques sont reliés entre eux par des connexions horizontales et verticales.

Chaque réseau (nommé Rung) comporte:

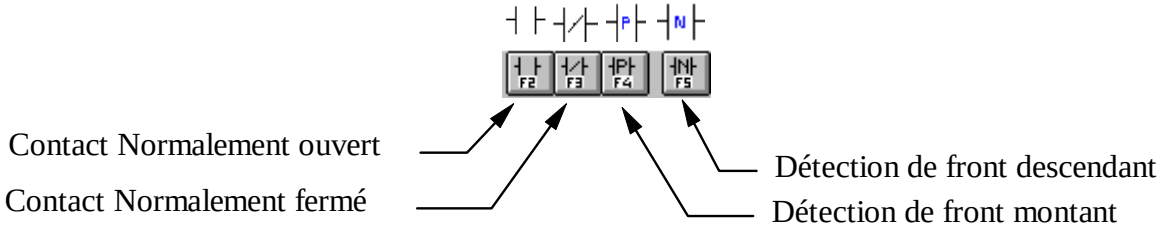
. une étiquette facultative(%Li:, cellule haut - gauche du rung),
. un commentaire facultatif (1ère ligne à droite de l'étiquette),
. 7 lignes et 11 colonnes (taille maximum)



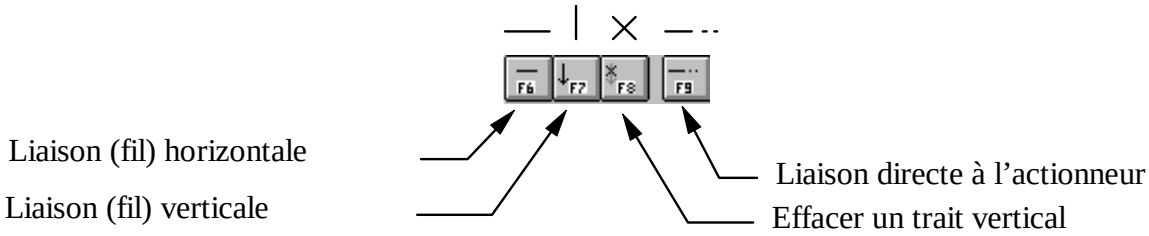
Remarque: Une réceptivité est composé d'un seul rung.

Eléments graphiques du langage à contact

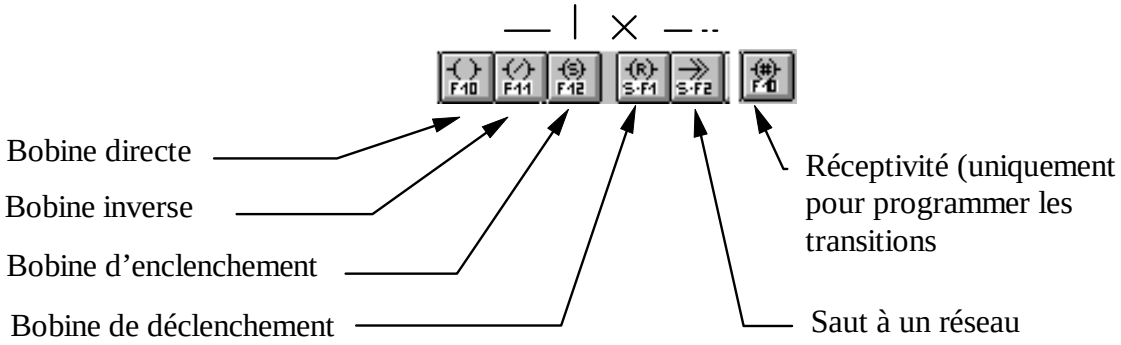
Instructions booléennes



Eléments de liaison



Eléments d'action



Eléments d'action programmés

Bobine appel à un sous-programme (CALL).

Permet un branchement en début de sous-programme local. L'exécution d'un appel à un sous-programme provoque : - l'arrêt de la « *»scrutation»* » du réseau en cours, - l'exécution du sous-programme,- la reprise de la « *»scrutation»* » du réseau interrompu. Un sous-programme peut en appeler un autre.

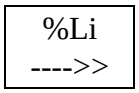
Retour de sous - programme (RETURN)

Cet élément est réservée au sous-programme SR. Elle permet le retour au module appelant.

Arrêt programme (HALT)

Cet élément provoque l'arrêt de l'exécution du programme.

Eléments de saut



Saut à un autre réseau

SHIFT+F2: permet un branchement à un réseau étiqueté, amont ou aval. Les sauts (JMP) ne sont effectifs qu'au sein d'un même module de programmation. Un ? précède l'étiquette %Li lorsque celle-ci n'a pas été préalablement définie.

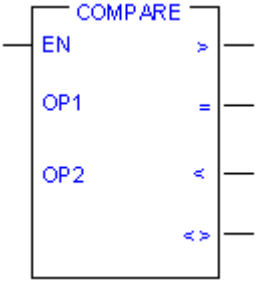
L'exécution d'un saut provoque :

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.

Blocs comparaison

Bloc comparaison verticale



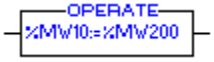
SHIFT+F4: permet la comparaison de 2 opérandes suivant le résultat la sortie correspondante passe à 1.Si EN=0, les sorties sont mises à 0.Dimension : 2 colonnes/4 lignes.

Bloc comparaison horizontale

SHIFT+F5: permet la comparaison de 2 opérandes la sortie passe à 1 lorsque le résultat est vérifié. Dimension : 2 colonnes/1 ligne.

Bloc opération

Bloc opération

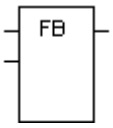


SHIFT+F3: réalisent les opérations arithmétiques, logiques font appel au langage littéral structuré. Dimension : 4 colonnes/1 ligne.

Appel de fonction : accès à la saisie assistée

Blocs fonctions

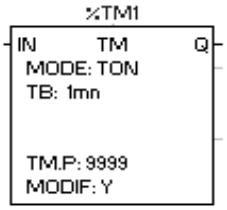
Temporisateur
Compteur
Monostable
Registre
Programmateur
Cyclique (Drum)



SHIFT+F7:Chacun des blocs fonctions prédéfinis utilise des entrées et des sorties permettant de les relier aux autres éléments graphiques.

Bloc fonction prédéfinis

Temporisateur



Le temporisateur dispose de 3 modes de fonctionnement :.

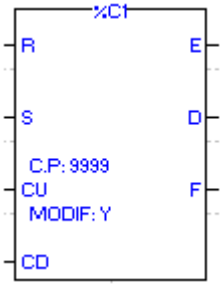
- TON: ce mode permet de gérer des retards à l'enclenchement. Ce retard est programmable et peut être modifiable ou non par terminal..
- TOF : ce mode permet de gérer des retards au déclenchement. Ce retard est programmable et peut être modifiable ou non par terminal..
- TP : ce mode permet d'élaborer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

Caractéristiques

Numéro temporisateur	%TMi	0 à 63
Mode	TON TOF TP	. retard à l'enclenchement (par défaut) . retard au déclenchement . monostable
Base de temps	TB	1mn (par défaut), 1s, 100ms, 10ms, Plus la base de temps est faible, plus la précision du temporisateur est grande. 16 Tempo. maxi avec TB = 10 ms
Valeur courante	%TMi.V	Mot qui croît de 0 à %TMi.P sur écoulement du temporisateur. Peut être lu, testé, mais non écrit par programme (1).
Valeur de présélection	%TMi.P	0£%TMi.P£9999. Mot qui peut être lu, testé, et écrit par programme. Est mis à la valeur 9999 par défaut. La durée ou retard élaboré est égal à %TMi.P x TB.
Réglage par terminal (MODIF)	Y/N	Y : possibilité de modification de la valeur de présélection %TMi.P en réglage. N : pas d'accès en réglage.
Entrée "Armement"	IN	Sur front montant (mode TON ou TP) ou front descendant (mode TOF), démarre le temporisateur.
Sortie "Temporisateur"	Q	Bit associé %TMi.Q, sa mise à 1 dépend de la fonction réalisée TON, TOF ou TP.

(1) %TMi.V peut être modifiée par terminal.

Compteur



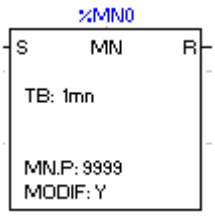
Le bloc fonction compteur/décompteur permet d'effectuer le comptage ou le décomptage d'événements, ces deux opérations peuvent être simultanées.

Caractéristiques

Numéro de compteur	**%Ci**	**0 à 31**
Valeur courante	%Ci.V	Mot incrémenté ou décrémenté en fonction des entrées CU et CD. Peut être lu, testé mais non écrit par programme (1).
Valeur de préselection	%Ci.P	0£%Ci.P£9999. Mot pouvant être lu, testé, écrit. (Mis à 9999 par défaut)
Réglage par terminal (MODIF)	Y/N	Y: possibilité de modification de la valeur de présélection en réglage. N : pas d'accès en réglage.
Entrée remise à zéro	R	Sur état 1 : %Ci.V = 0.
Entrée présélection	S	Sur état 1: %Ci.V = %Ci.P.
Entrée comptage	CU	Incrémente %Ci.V sur front montant.
Entrée décomptage	CD	Décrémente %Ci.V sur front montant.
Sortie débordement	E (Empty)	Le bit associé %Ci.E=1(mis à 1 quand %Ci.V devient égal à 9999, est remis à 0 si le compteur continue de décompter), lorsque le décomptage déborde %Ci.V passe de 0 à 9999, %S18=1. Lorsque le comptage déborde (%Ci.V passe de 9999 à 0) %S18=1.
Sortie préselection atteinte	D (Done)	Le bit associé %Ci.D=1, lorsque %Ci.V=%Ci.P.
Sortie débordement	F (Full)	Le bit associé %Ci.F =1 lorsque %Ci.V passe de 9999 à 0 (mis à 1 quand %Ci.V devient égal à 0, est remis à 0 si le compteur continue de compter).

(1) %Ci.V peut être modifiée par terminal.

Monostable



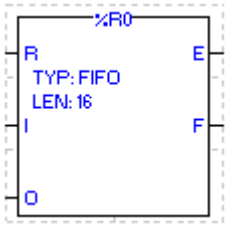
Le bloc fonction monostable permet d'élaborer une impulsion de durée précise.Cette durée est programmable et peut être modifiable ou non par terminal.

Caractéristiques

Numéro	%MNi	0 à 7
Base de temps	TB	1mn, 1s, 100ms, 10ms (1mn par défaut)
Valeur courante	%MNi.V	Mot qui décroît de %MNi.P vers 0 sur écoulement du temporisateur. Peut être lu, testé, mais non écrit
Valeur de présélection	%MNi.P	0 < %MNi.P £ 9999. Mot pouvant être lu, testé, écrit. La durée de l'impulsion (PRESET) est égale à : %MNi.P x TB

Modification MODIF	Y/N	Y : possibilité de modification de la valeur de présélection en réglage. N : pas d'accès en réglage.
Entrée "Départ"	S(Start)	Sur front montant %MNi.V = %MNi.P puis %MNi.V décroît vers 0
Sortie "Monostable"	R(Running)	Le bit associé %MNi.R est à 1 si %MNi.V > 0 (écoulement "en cours" monostable) %MNi.R = 0 si %MNi.V = 0

Registre

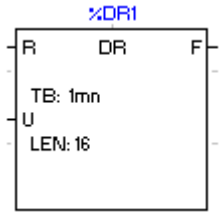


Un registre est un bloc mémoire permettant de stocker jusqu'à 255 mots de 16 bits de deux manières différentes :- file d'attente (premier entré, premier sorti) appelée pile FIFO (First In, First Out),- pile (dernier entré, premier sorti) appelée pile LIFO (Last In, First Out).

Numéro Registre	%Ri	0 à 3
Mode	FIFO LIFO	File d'attente (choix par défaut). Pile
Longueur	LEN	Nombre de mots de 16 bits (1 £ LEN £ 255) composant le bloc mémoire registre.
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 (ou instruction) "Stockage"	I (In)	Sur front montant provoque le stockage du contenu du mot %Ri.I dans le registre.
Entrée (ou instruction) "Déstockage"	O (Out)	Sur front montant provoque le rangement d'un mot d'information dans le mot %Ri.O.
Entrée (ou instruction) "Remise à zéro"	R(Reset)	Sur état 1 initialise le registre.
Sortie "Vide"	E(Empty)	Le bit %Ri.E associé indique que le registre est vide. Peut être testé.
Sortie "Plein"	F (Full)	Le bit %Ri.F associé indique que le registre est plein. Peut être testé.

Nota : lorsque les deux entrées I et O sont activées simultanément, le stockage est réalisé avant le déstockage.

Programmateur cyclique (Drums)



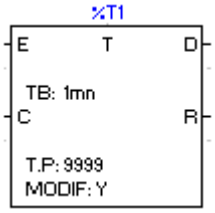
D'un principe de fonctionnement similaire au programmateur à cames, le programmateur cyclique (Drum) change de pas en fonction d'événements extérieurs. A chaque pas, le point haut d'une came donne un ordre exploité par l'automatisme. Dans le cas du programmateur cyclique, ces points hauts sont symbolisés 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.

Caractéristiques

Numéro	%DRi	0 à 7
Nombre de pas	LEN	1 à 16 (16 par défaut)

Base de temps	TB	1mn, 1s, 100ms, 10ms (1mn par défaut)
Temps enveloppe ou durée du pas en cours	%DRi.V	0£%DRi.V£9999. Mot remis à zéro à chaque changement de pas. Peut être lu, testé mais non écrit. La durée est égale à %DRi.V x TB
Numéro du pas en cours	%DRi.S	0£%DRi.S£15. Mot pouvant être lu et testé. Ne peut être écrit qu'à partir d'une valeur immédiate.%S18 = 1 si on tente d'écrire une valeur de pas non configuré.
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'ordre.
Sortie	F (FULL)	Indique que le dernier pas défini est en cours. Le bit %DRi.F associé peut être testé (%DRi.F=1 si %DRi.S = nombre de pas configurés - 1).
Etat d'un pas	%DRi.Wj	Mot de 16 bits définissant les états du pas j du programmateur i. Peut être lu, testé mais non écrit.
Bits d'ordre		Sorties ou bits internes associés au pas (16 bits d'ordre).

Temporisateur série 7



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.

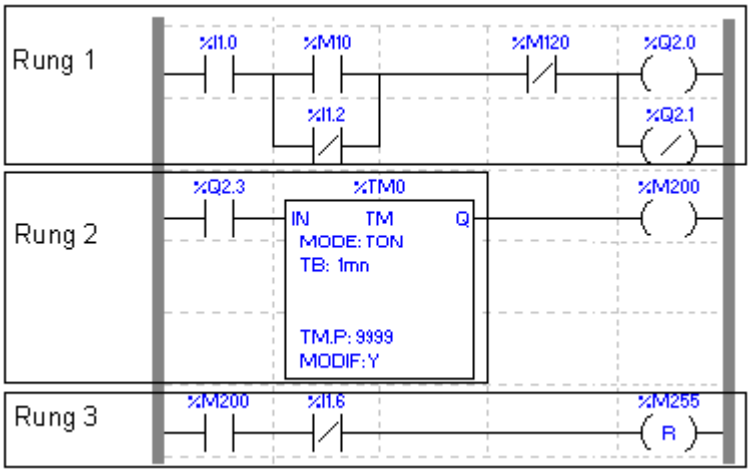
Caractéristiques

Numéro	%Ti	0 à 63
Base de temps	TB	1mn, 1s, 100ms, 10ms (1mn par défaut)
Valeur courante	%Ti.V	Mot qui décroît de %Ti.P vers 0 sur écoulement du temporisateur. Peut être lu, testé, mais non écrit.
Valeur de présélection	%Ti.P	0 < %Ti.P £ 9999. Mot qui peut être lu, testé, écrit. Est mis à la valeur 9999 par défaut.
Modification MODIF	Y/N	Y : possibilité de modification de la valeur de présélection en réglage N : pas d'accès en 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
Sortie "Temporisateur écoulée"	D(Done)	Le bit associé %Ti.D = 1, si temporisateur écoulé%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.

Règles d'exécution d'un réseau de contacts

Principe d'exécutionOrdre d'exécution

L'exécution d'un réseau s'effectue réseau connexe par réseau connexe, puis au sein d'un réseau connexe, dans le sens de gauche à droite



Un réseau connexe contient des éléments graphiques tous reliés entre eux par des éléments de liaison (hors barre de potentiel), mais indépendants des autres éléments graphiques du réseau (pas de liaisons verticales vers le haut ou vers le bas en limite de réseau connexe).

Le premier réseau connexe évalué est celui dont le coin gauche est situé le plus en haut à gauche.

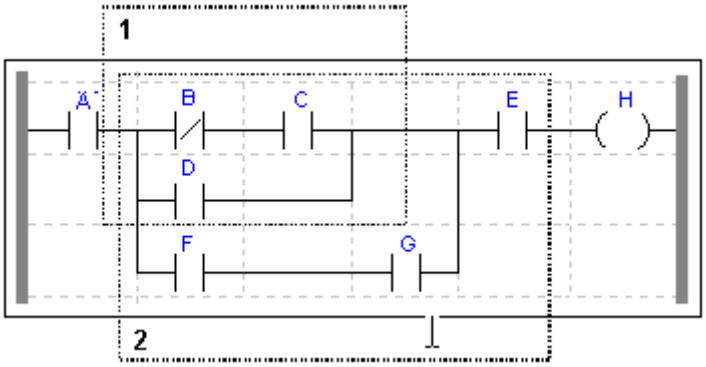
Un réseau connexe est évalué dans le sens de l'équation : évaluation du réseau de haut en bas, ligne par ligne, et dans chaque ligne de gauche à droite. Dans le cas où une liaison verticale de convergence est rencontrée, le sous réseau qui lui est associé est évalué (selon la même logique) avant de continuer l'évaluation du réseau qui l'englobe.

En respectant cet ordre d'exécution, le système :

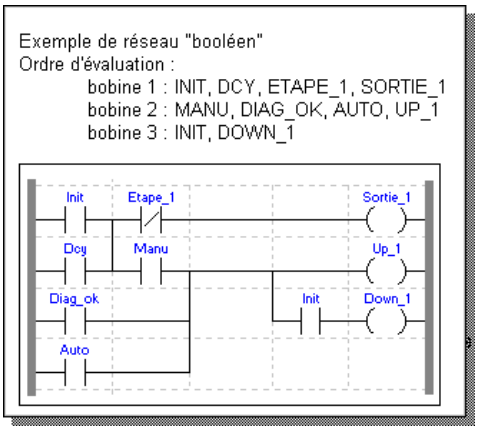
- évalue l'état logique de chaque contact, en fonction de la valeur courante des objets internes de l'application ou de l'état des entrées des modules d'entrées/sorties acquis en début de cycle,
- exécute les traitements associés aux fonctions, aux blocs fonctions et aux sous programmes,
- met à jour les objets bits associés aux bobines (la mise à jour des sorties des modules d'entrées/sorties s'effectue en fin de cycle),
- débranche vers un autre réseau étiqueté du même module programme (saut JMP), ou retourne au module appelant (RETURN) ou arrêt du programme (HALT).

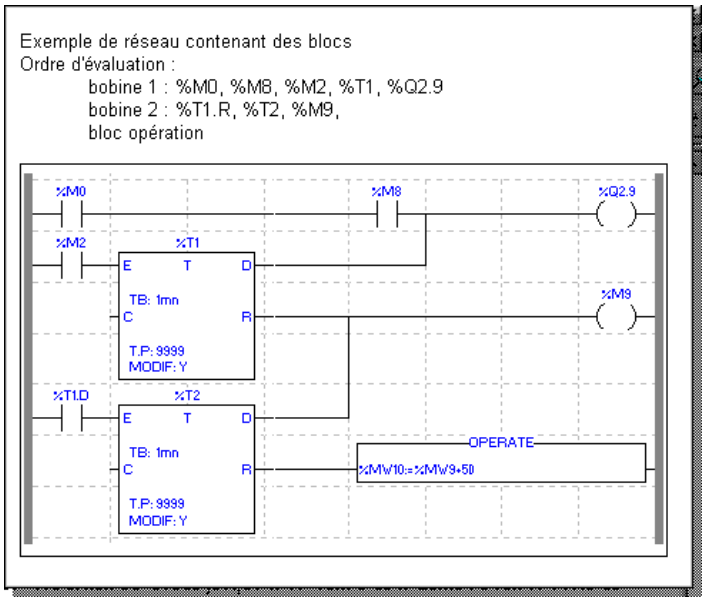
Nota : un réseau de contacts ne doit pas contenir des réseaux connexes imbriqués.

Ordres d'exécution d'un réseau



- évaluation du réseau jusqu'à rencontre de la 1ère liaison verticale de convergence : contacts A, B,
- - évaluation du 1er sous réseau : contact D
- - poursuite de l'évaluation du réseau jusqu'à la rencontre de la 2ème liaison verticale de convergence: contact E
- - évaluation du 2ème sous réseau : contacts F et G
- - évaluation de la bobine H.





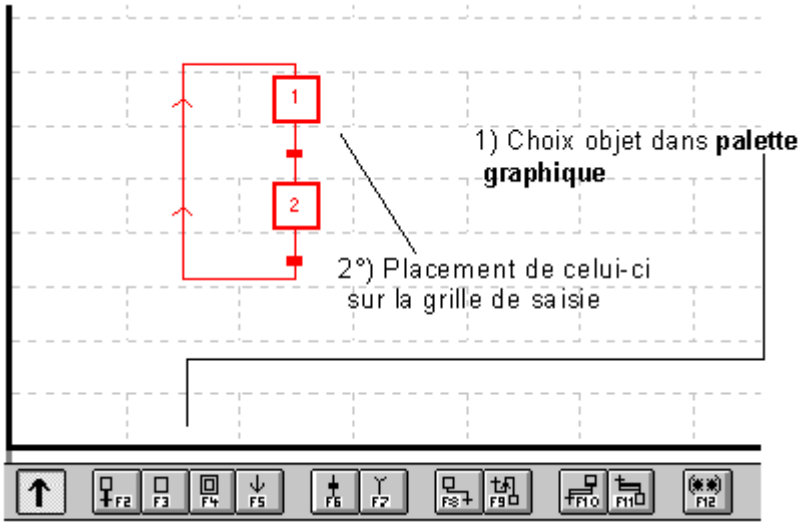
Constitution d'un programme en langage Grafcet

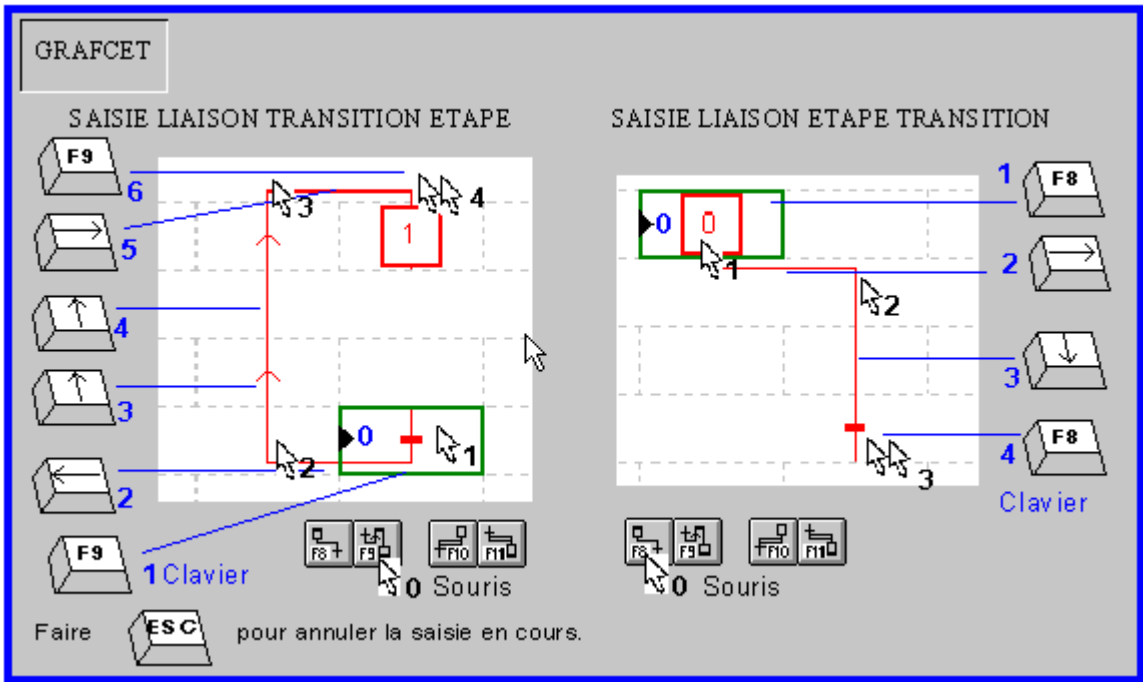
Le langage Grafcet (GR7) permet de représenter graphiquement et de façon structurée le fonctionnement d'un automatisme séquentiel.

Cette description s'effectue à l'aide d'objets graphiques simples représentant:

- . Les étapes auxquelles peuvent être associées des actions.
- . Les transitions auxquelles sont associées des réceptivités
- . Les liaisons orientée reliant une étape à une transition ou une transition à une étape

La saisie du graphe s'effectue par pages Grafcet repérées de 0 à 7 dans la barre d'état.

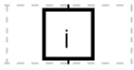




Objets graphiques Grafcet

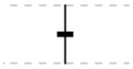
Etapes  Maximum: 128 dont 64 actives simultanément

Initiale **F4**  i = 0 à 63

Simple **F3**  i = 0 à 127

Etape + Transition 

Transitions  Maximum: 1024 dont 96 valides simultanément

Simple **F6** 

Divergence en ET **F11**  Aiguillage vers 11 étapes maximum

Convergence en ET **F10** 

Liaison Etape ->Transition **F8**  Activation de 11 étapes simultanément maximum

Liaison Transition -> Etape **F9** 

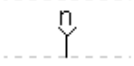
Renvois 

De destination **F5**  n = numéro de l'étape destination

D'origine **F7**  n = numéro de l'étape origine

Renvois 

De destination **F5**  n = numéro de l'étape destination

D'origine **F7**  n = numéro de l'étape origine

Liaisons orientées 

Vers le haut

Vers le bas

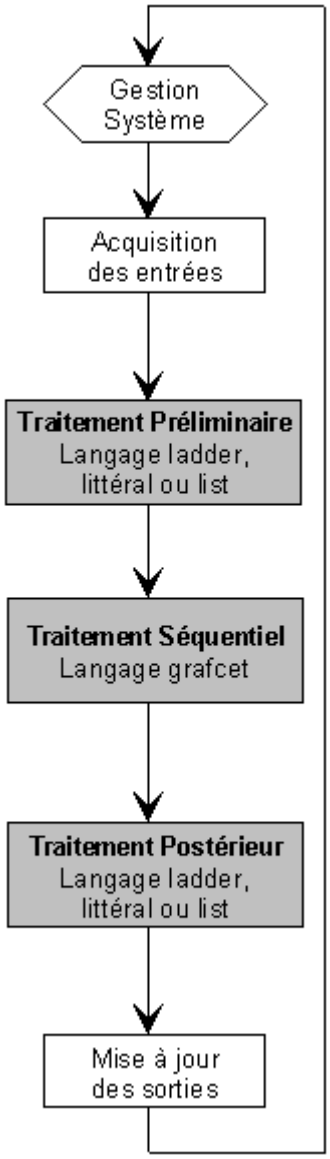
Vers la gauche ou la droite  Permet le rebouclage d'un graphe (Utiliser les touches flèches)

Commentaires 

La tâche GRAFCET

Un programme écrit en langage grafcet comporte 3 traitements successifs. Leur scrutation s'effectue selon le cycle de base suivant:

Seule la tâche maître (MAST) supporte le langage Grafcet.



Gestion système :

Le système assure implicitement:
. la mise à jour des bits et mots système,
. la surveillance de l'automate,
. le traitement des requêtes du terminal,
. le routage de la messagerie.

Acquisition des entrées

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

Traitement Préliminaire

Saisi en langage Ladder, Littéral ou List, il permet de traiter:
. le prépositionnement du graphe,
. les modes de marches de l'application,
. la logique d'entrée.

Traitement Séquentiel

Saisi en langage Grafcet, Il permet de traiter l'ossature séquentielle de l'application et donne accès au traitement des réceptivités et des actions associées aux étapes. Nous ne programmerons pas les actions associées au étapes dans le traitement séquentiel, mais dans le traitement postérieur.

Traitement Postérieur

Saisi en langage Ladder, Littéral ou List,il permet de traiter:
. la logique de sortie,
. la surveillance et les sécurités spécifiques aux sorties,
. la gestion des fonctions d'automatisme (temporisateur, compteur,...)
Nous programmerons les sorties associées aux étapes dans cette phase du traitement.

.Mise à jour des sorties

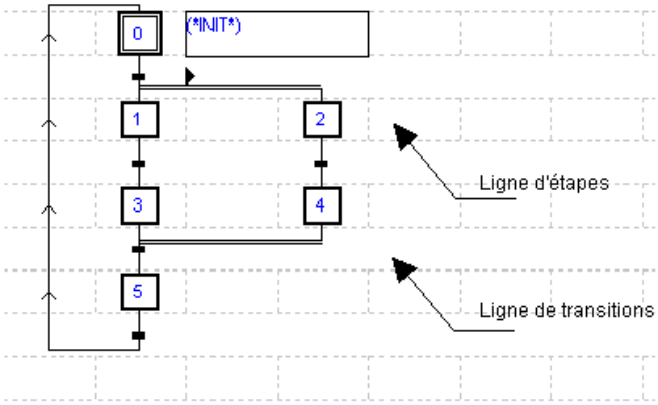
Mise à jour de l'état physique des sorties (valeurs figées pendant le traitement).

La page Grafcet

La page Grafcet est affichée sous forme d'une matrice composée de 14 lignes et 11 colonnes définissant 154 cellules. Chaque cellule peut recevoir un objet graphique.

Les lignes sont de 2 types:

- - Lignes d'étapes où sont saisies les étapes et les renvois de destination,
- - Lignes de transitions où sont saisies les transitions et les renvois d'origine.



Des commentaires peuvent être saisis. Ce sont des objets graphiques "indépendants" qui ne sont rattachés ni à une étape ni à une transition.

Un module programme est constitué au maximum de 8 pages Grafcet.

Pour accéder à une page Grafcet, exécuter la commande Edition/Atteindre ou cliquer sur l'icône .

Règles à utiliser pour la création d'un GRAFCET

Règle A : Numérotation des étapes : Toute étape doit être repérée. Le repère (nombre compris entre 0 et 127) s'inscrit à l'intérieur du cadre symbolisant l'étape.

Règle B : Unicité des repères dans un module : Deux repères identiques ne peuvent pas être programmés dans un même module de programmation.

Règle C : Représentation des liaisons Etape -> Transition : Une liaison Etape -> Transition peut se tracer de la droite vers la gauche ou de la gauche vers la droite.

Règle D : Représentation des divergences en ET : Une divergence en ET se représente toujours de la gauche vers la droite.

Règle E : Liaisons orientées : Une liaison verticale est toujours orientée du haut vers le bas. Dans le cas contraire, il faut le spécifier par une flèche directionnelle vers le haut.

Règle F : Croisement de liaisons : Les liaisons orientées horizontales ou verticales peuvent se croiser ou se rencontrer. Le croisement d'une liaison orientée avec le prolongement d'une divergence ou convergence en ET est impossible.

Règle G : Utilisation des renvois : A un renvoi d'origine doit toujours correspondre un renvoi de destination.

Objets spécifiques Grafcet

Type	Adresse	Commentaires
Bits associés aux étapes	%Xi	Etat de l'étape i du graphe principal
Bits système associés au GRAFCET	%S21%S22%S23%S26	
Mots temps associés au GRAFCET	%Xi.T	Temps de l'activité de l'étape i (Chart)
Mots système	%SW20	Nombre d'étapes actives (maximum: 64)
%SW21	Nombre de transitions valides (maximum: 96)	
Instructions	SET RESET	Activation ou désactivation d'une étape

Bit %S21

Normalement à l'état 0, la mise à 1 du bit %S21 provoque l'initialisation du grafcet.

Mise à 1
- Par mise à l'état 1 de %S0 (reprise à froid)
- Par le programme utilisateur (dans le PRL)
- Par le terminal (en mode données ou réglage)

Mise à 0
- Par le système en début de traitement séquentiel
- Par le programme utilisateur (dans le PRL)
- Par le terminal (en mode données ou réglage)

Bit %S22

Normalement à l'état 0, la mise à 1 du bit %S22 provoque la désactivation des étapes actives.

Mis à 1
- Par le programme utilisateur (dans le PRL)

Mis à 0
- Par le système dans le traitement séquentiel

Bit %S23

Normalement à l'état 0, la mise à 1 du bit %S23 provoque le maintien en l'état de la situation courante du GRAFCET.

Mis à 1
- Par le programme utilisateur (dans le PRL)

Mis à 0
- Par le programme utilisateur (dans le PRL)

Bit %S26

Normalement à l'état 0, la mise à 1 du bit %S23 signale un dépassement de capacité de la table des étapes actives et/ou de la table des transitions valides (capacités configurables).

Mots temps associés au Grafcet: %Xi.T

i : Numéro d'étape compris entre 0 et 127.
T: Valeur comprise entre 0 et 9999 .
Base de temps: 100ms.

A l'activation de l'étape, le contenu est remis à zéro puis incrémenté,
A la désactivation de l'étape, le contenu est figé.
Le nombre de mots temps d'activité n'est pas configurable, un mot est réservé pour chaque étape.
ces mots ne sont pas indexables.

Instructions SET et RESET

Exemples:

- Demande d'activation de l'étape 0

LITTERAL	LADDER	LIST
SET %X0	%X0	LD1
	-(S)-	S %X0

- Demande de désactivation de l'étape 2

LITTERAL	LADDER	LIST
RESET %X2	%X2	LD1
	-(R)-	R %X2

Nota: La programmation de ces instructions est réservée au traitement préliminaire.

Créer un GRAFCET

1. Créer le module programme correspondant au langage Grafcet (GR7).

1.1 Sélectionner la commande Application/Programmes. 

1.2 Sélectionner la tâche puis le module programme. La tâche MAST est créée par défaut.

- si l'application comporte du Grafcet, sélectionner:
  - PRL ,
  - CHART puis la page,
  - POST.
- si la tâche est événementielle, sélectionner l'événement correspondant.
- pour créer un sous programme, sélectionner Créer SR et indiquer le n° de SR.

1.3. Accéder au module sélectionné par Ouvrir.

1.4. Sélectionner le type de langage

- - Langage à contacts: LD
- - Liste d'instructions: IL
- - Littéral structuré: ST

et valider par OK.

2. Saisir les éléments graphiques

⇒ Avec la souris:

1. Cliquer sur l'objet graphique situé dans la palette graphique.
2. Cliquer sur la grille à l'endroit désiré.
   Dans le cas d'une étape, un numéro d'étape est proposé par défaut, le modifier éventuellement puis valider par ENTER.
3. Procéder de la même manière pour les autres objets graphiques.
4. Valider le graphe par la touche ENTER, par les touches <CONTROL>+W, par la commande Edition/Valider graphe ou cliquer sur l'icône

⇒ Avec le clavier:

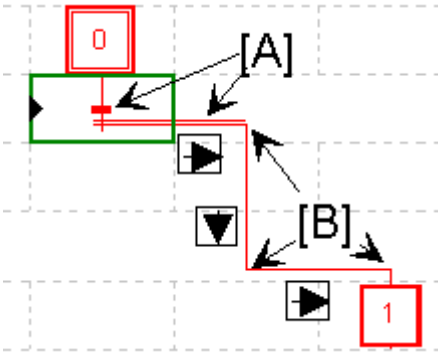
1. Positionner le curseur sur la grille à l'endroit désiré (à l'aide des touches flèches).
2. Appuyer sur la touche fonction F2 à F11 correspondant à l'objet graphique situé dans la palette graphique à insérer.

Dans le cas d'une étape, un numéro d'étape est proposé par défaut, le modifier éventuellement.

3. Valider par ENTER. Procéder de la même manière pour les autres objets graphiques.
4. Valider le graphe par la touche ENTER, par la commande Edition/Valider graphe ou par <CONTROL>+W. Lorsqu'un graphe est validé, l'affichage change:
   - les objets graphiques passent de la couleur rouge à la couleur noire,
   - Les bords de la page deviennent gris.

Réalisation d'une divergence en ET

Une divergence en ET débute sur une transition pour aller sur une étape.



Pour réaliser cette liaison:

⇒ Avec la souris

1. Cliquer sur l'objet graphique situé dans la palette graphique (F11).
2. Cliquer sur la transition de départ ou sur le segment préexistant de la divergence [A] (cas d'une divergence multiple).

   L'éditeur passe dans un mode de tracé: aucune autre action n'est disponible tant que la liaison n'est pas terminée ou abandonnée.
3. Dessiner la liaison en cliquant aux points de rupture du tracé [B] (changement de direction) dans les lignes de transitions.
4. Effectuer un double clic au dernier point de rupture ou cliquer de nouveau sur l'objet graphique situé dans la palette graphique (F11).
   Si la cellule de destination est vide, une étape est créée automatiquement.
5. Modifier éventuellement le numéro d'étape puis valider par Enter.

⇒ Avec le clavier

1. Se positionner sur la transition de départ ou sur le segment préexistant de la divergence [A] (cas d'une divergence multiple) à l'aide des flèches.
2. Appuyer sur F11.

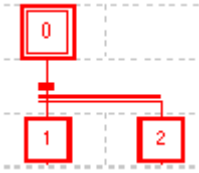
L'éditeur passe dans un mode de tracé: aucune autre action n'est disponible tant que la liaison n'est pas terminée ou abandonnée.

3. Dessiner la liaison à l'aide des touches flèches.
4. Appuyer de nouveau sur F11 au dernier point de rupture.
   Si la cellule de destination est vide, une étape est créée automatiquement.

5 Modifier éventuellement le numéro d'étape puis valider par Enter.

Nota: En cours de saisie, dessiner en "revenant sur ses pas" efface le tracé courant.
Pour annuler la saisie en cours, utiliser la touche ESC.

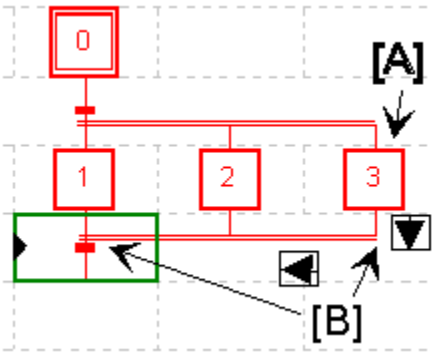
Nota : Une divergence en ET se représente toujours de la gauche vers la droite.



Limitations : Le segment représenté par un trait double ne peut être coupé par une autre liaison.

Réalisation d'une convergence en ET

Une convergence en ET débute sur une étape pour aller sur une transition.



Pour réaliser cette liaison:

⇒ Avec la souris

1. Cliquer sur l'objet graphique situé dans la palette graphique (F10).
2. Cliquer sur l'étape de départ [A].
   L'éditeur passe dans un mode de tracé: aucune autre action n'est disponible tant que la liaison n'est pas terminée ou abandonnée.
3. Dessiner la liaison en cliquant aux points de rupture du tracé [B] (changement de direction) dans les lignes de transitions.
   Si la cellule de destination est vide :
4. Effectuer un double clic au dernier point de rupture ou cliquer de nouveau sur l'objet graphique situé dans la palette graphique pour créer la transition.

⇒ Avec le clavier

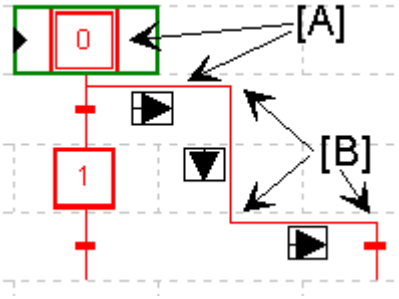
1. Se positionner sur l'étape de départ à l'aide des flèches.
2. Appuyer sur F10.
   L'éditeur passe dans un mode de tracé: aucune autre action n'est disponible tant que la liaison n'est pas terminée ou abandonnée.
3. Dessiner la liaison à l'aide des touches flèches
   Si la cellule de destination est vide :

4. Appuyer de nouveau sur F10 au dernier point de rupture pour créer la transition.

Nota : En cours de saisie, dessiner en "revenant sur ses pas" efface le tracé correspondant.
Pour annuler la saisie en cours, utiliser la touche ESC.
Limitations : Le segment représenté par un trait double ne peut être coupé par une autre liaison.

Réalisation d'une divergence en OU (liaison Etape -> Transition)

Une liaison étape -> transition débute sur une étape pour aller sur une transition.



Pour réaliser cette liaison:

⇒ Avec la souris

1. Cliquer sur l'objet graphique situé dans la palette graphique (F8).
2. Cliquer sur l'étape de départ ou sur le segment préexistant de la liaison étape -> transition [A] (cas d'une liaison multiple). L'éditeur passe dans un mode de tracé: aucune autre action n'est disponible tant que la liaison n'est pas terminée ou abandonnée.
3. Dessiner la liaison en cliquant aux points de rupture du tracé [B] (changement de direction) dans les lignes de transitions.
   Si la cellule de destination est vide :
4. Effectuer un double clic au dernier point de rupture ou cliquer de nouveau sur l'objet graphique situé dans la palette graphique pour créer la transition.

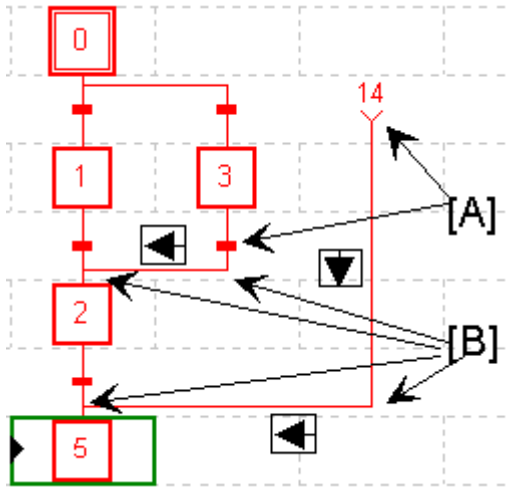
⇒ Avec le clavier

1. Se positionner sur l'étape de départ ou sur le segment préexistant de la liaison étape -> transition [A] (cas d'une liaison multiple)à l'aide des flèches.
2. Appuyer sur F8. L'éditeur passe dans un mode de tracé: aucune autre action n'est disponible tant que la liaison n'est pas terminée ou abandonnée.
3. Dessiner le tracé à l'aide des touches flèches.
   Si la cellule de destination est vide :
4. Appuyer de nouveau sur F8 au dernier point de rupture pour créer la transition.

Nota: En cours de saisie, dessiner en "revenant sur ses pas" efface le tracé correspondant.
Pour annuler la saisie en cours, utiliser la touche ESC.

Réalisation d'une convergence en OU (liaison Transition -> Etape)

Réalisation d'une liaison Transition -> Etape



Une liaison transition -> étape débute sur une transition pour aller sur une étape.

Pour réaliser cette liaison:

⇒ Avec la souris

1. Cliquer sur l'objet graphique situé dans la palette graphique (F9).
2. Cliquer sur la transition de départ ou le renvoi d'origine [A]. L'éditeur passe dans un mode de tracé: aucune autre action n'est disponible tant

que la liaison n'est pas terminée ou abandonnée.

3. Dessiner la liaison en cliquant aux points de rupture du tracé [B] (changement de direction) dans les lignes de transitions.

   Si la cellule de destination est vide :

4. Effectuer un double clic au dernier point de rupture ou cliquer de nouveau sur l'objet graphique situé dans la palette graphique pour créer l'étape.

Modifier éventuellement le numéro d'étape puis valider par Enter.

⇒ Avec le clavier

1. Se positionner sur la transition de départ ou le renvoi d'origine à l'aide des flèches.
2. Appuyer sur F9. L'éditeur passe dans un mode de tracé: aucune autre action n'est disponible tant que la liaison n'est pas terminée ou abandonnée.
3. Dessiner la liaison à l'aide des touches flèches jusqu'à l'étape ou le renvoi de destination.

   Si la cellule de destination est vide :

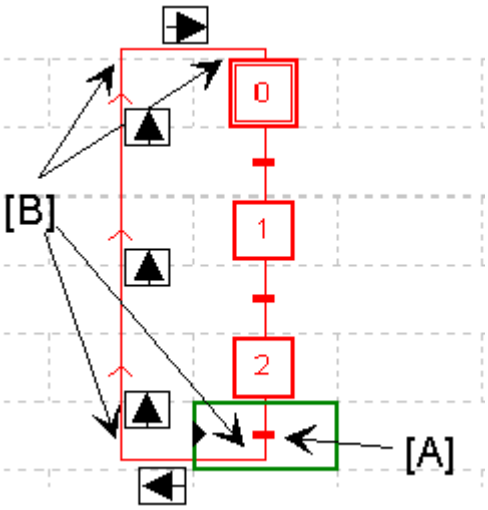
4. Appuyer de nouveau sur F9 au dernier point de rupture pour créer l'étape.

5. Modifier éventuellement le numéro d'étape puis valider par Enter.

Nota: En cours de saisie, dessiner en "revenant sur ses pas" efface le tracé correspondant.
Pour annuler la saisie en cours, utiliser la touche ESC.

Réalisation d'un rebouclage

2 types de rebouclage sont disponibles pour achever un graphe:



Par liaisons orientées

⇒ Avec la souris

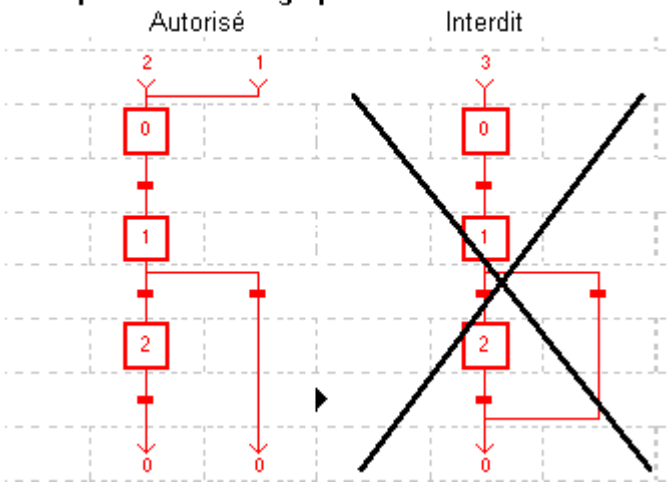
1. Cliquer sur l'objet graphique situé dans la palette graphique (F9).
2. Cliquer sur la transition de fin de graphe [A]. L'éditeur passe dans un mode de tracé: aucune autre action n'est disponible tant que la liaison n'est pas terminée ou abandonnée.
3. Dessiner la liaison en cliquant aux points de rupture du tracé [B] (changement de direction) dans les lignes de transitions.
4. Effectuer un double clic sur l'étape à relier ou cliquer de nouveau sur l'objet graphique situé dans la palette graphique pour valider l'objet.

⇒ Avec le clavier

1. Se positionner sur la transition de départ [A] à l'aide des flèches.
2. Appuyer sur F9. L'éditeur passe dans un mode de tracé: aucune autre action n'est disponible tant que la liaison n'est pas terminée ou abandonnée.
3. Dessiner la liaison à l'aide des touches flèches jusqu'à l'étape ou le renvoi de destination.

Par renvois

Exemple de rebouclage par renvois :



⇒ Avec la souris

1. Cliquer sur l'objet graphique situé dans la palette graphique (F5 ou F7)
2. Cliquer sur la grille à l'endroit désiré.
3. Renseigner le numéro d'étape d'origine ou de destination puis valider par ENTER.

⇒ Avec le clavier

1. Se positionner à l'endroit désiré.
2. Sélectionner l'objet graphique désiré en appuyant sur F5 ouF7.
3. Renseigner le numéro d'étape d'origine ou de destination puis valider par ENTER.

Nota: En cours de saisie, dessiner en "revenant sur ses pas" efface le tracé correspondant.
Pour annuler la saisie en cours, utiliser la touche ESC.

3. Programmer les actions

NOTE : Télémécanique propose cette méthode pour programmer les actions, directement dans les étapes. Les actions continues ne sont pas désactivées à la désactivation de l'étape.
NE PROGRAMMEZ PAS VOS ACTIONS DE CETTE MANIERE, MAIS PROGRAMMEZ LES DANS LE POST. Et, dans le post, ne mettez qu'une seule bobine par action.

Pour programmer une action:

⇒ Avec la souris:

1. Sélectionner l'étape en effectuant un double clic droit.

2. Sélectionner le type d'action associée à l'étape (action à l'activation, continue ou à la désactivation) par un double clic gauche.

3. Sélectionner le type de langage:

   - Langage à contacts: LD
   - Liste d'instructions: IL
   - Littéral structuré: ST

et valider par OK.

Avant validation, il est possible d'annuler la programmation en appuyant sur ESC.

4. Réaliser la programmation.

⇒ Avec le clavier:

⇒

1. Positionner le curseur sur l'étape à l'aide des touches flèches puis valider la sélection en utilisant la commande Service/Accès au code.

2. Sélectionner le type d'action associée à l'étape à l'aide des touches flèches puis valider par ENTER.
3. Sélectionner le type de langage à l'aide des touches flèches:
   - Langage à contacts: LD
   - Liste d'instructions: IL
   - Littéral structuré: STet valider par ENTER.

Avant validation, il est possible d'annuler la programmation en appuyant sur ESC.

4. Réaliser la programmation.

Programmation des réceptivités

Pour programmer une réceptivité:

⇒Avec la souris:
1. Sélectionner la transition en effectuant un double clic droit.
2. Sélectionner le type de langage:
   - Langage à contacts: LD
   - Liste d'instructions: IL

et valider par OK. Avant validation, il est possible d'annuler la programmation en appuyant sur ESC.

3. Réaliser la programmation.

⇒ Avec le clavier:

1. Positionner le curseur sur la transition à l'aide des touches flèches puis valider la sélection en utilisant la commande Service/Accès au code.

2. Sélectionner le type de langage à l'aide des touches flèches:
   - Langage à contacts: LD
   - Liste d'instructions: IL

et valider par ENTER. Avant validation, il est possible d'annuler la programmation en appuyant sur ESC.

3. Réaliser la programmation.

Règles de programmation en langage à contacts (LD)
Seuls les éléments suivants peuvent être utilisés :

- Eléments graphiques de test: contacts (Bi, I/O, Ti,D...), blocs comparaisons,

- Eléments graphiques d'action: bobine "dièse" uniquement (les autres bobines n'étant pas significatives dans ce cas).

Règles de programmation en langage liste d'instructions (IL)

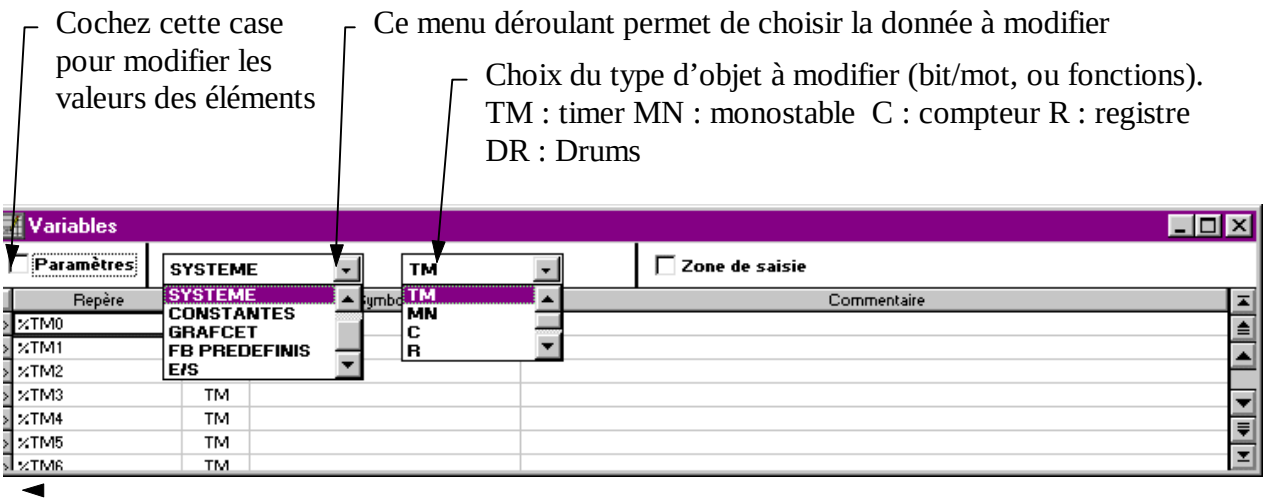
La liste d'instructions admise pour l'écriture d'une réceptivité diffère d'une liste d'instructions classique par:

- la structure générale:
- pas d'étiquette (%L).
- la liste des instructions:
- pas d'instructions d'actions (objetsbits, mots ou blocs fonctions),
- pas de saut, d'appel de sous-programme.

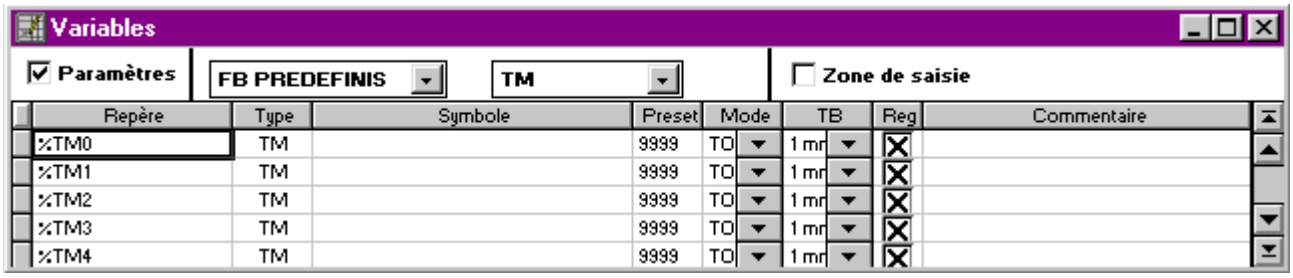
Nota: Pour annuler la saisie en cours, utiliser la touche ESC.

Réglage des bits, mots et blocs fonction :

Un programme, qu'il soit en LADDER ou en GRAFCET, utilise différentes variables, en particulier celles liées aux blocs fonction (temporisateur, compteurs,...). Le réglage de ces différentes variables s'effectue dans un tableau. Pour y accéder, cliquez sur l'icône suivant :  . Cela vous donne accès au tableau suivant :



Après avoir coché « Paramètres », Cliquez dans les cases ou sur les menus déroulants pour modifier les paramètres et autres valeurs prédéfinies.



Sommaire

Nouveau GRAFCET 3

Configuration de l'automate 3
menu application - configurer 3
Choix des modules 4
Valider la configuration 5

Les objets : entrées, sorties, mémoire, configuration 6
L'adressage des principaux objets 6
Adressage des bits : 6
Bits système 6

Les langages des automates TSX - télémécanique 8

Le langage à contacts (ladder) 8
Généralités 8
Les éléments graphiques du langage à contact 9
Instructions booléennes 9
Eléments de liaison 9
Eléments d'action 9
Eléments d'action programmées 9
Eléments de saut 10
Blocs comparaison 10
Blocs fonction 11
Fonctions prédéfinies 11
Temporisateurs 11
Compteurs 12
Monostables 12
Registres 13
Programmateur cyclique 13
Temporisateur série 7 14
Règles d'exécution d'un réseau de contacts 15
Principe d'exécution 15
ordres d'exécution 16

Constitution d'un programme en langage GRAFCET 17
Les objets graphiques Graphique GRAFCET 18
La tâche GRAFCET 19
La page GRAFCET 20
Règles à utiliser pour la création d'un GRAFCET 20
Objets spécifiques GRAFCET 21
Créer un GRAFCET 22
Divergence et convergence en ET 23
Divergence et convergence en OU 25
Rebouclage 26
Programmer les actions 27
Programmer les réceptivités 28
Réglage des mots, bits et bloc fonctions 30