
Instructions de base

4

Présentation

Contenu de ce chapitre

Ce chapitre décrit les instructions de base du langage PL7.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sujet	Page
Instructions de chargement	15
Instructions d'affectation	15
Instruction ET logique	16
Instruction OU logique	16
Bloc fonction temporisateur %Tmi	17
Instruction de comparaison	20

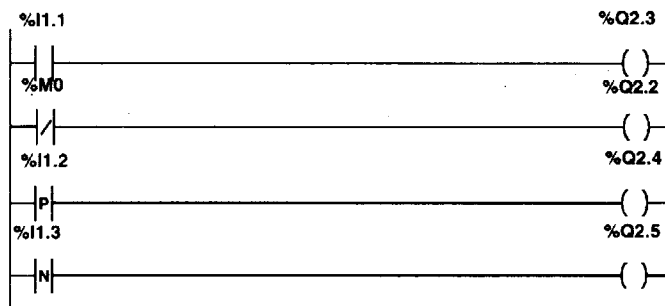
Instructions de chargement

Rôle Le tableau suivant décrit le rôle de chacune des instructions.

Langage à contacts	Liste d'instructions	Littéral structuré	Description	Chronogramme
	LD	:=	Contacts à fermeture: contact passant (résultat à 1) quand l'objet bit qui le pilote est à l'état 1.	Opérande Résultat
	LDN	:=NOT	Contacts à ouverture: contact passant (résultat à 1) quand l'objet bit qui le pilote est à l'état 0.	Opérande Résultat
	LDR	:=RE	Contacts à front montant: détection du passage de 0 à 1 de l'objet bit qui le pilote. La mise à 1 du résultat s'effectue pendant 1 cycle.	Opérande Résultat
	LDF	:=FE	Contacts à front descendant: détection du passage de 1 à 0 de l'objet bit qui le pilote. La mise à 1 du résultat s'effectue pendant 1 cycle.	Opérande Résultat

Exemple en langage à contacts

L'exemple suivant montre la programmation des instructions de chargement en langage à contacts.



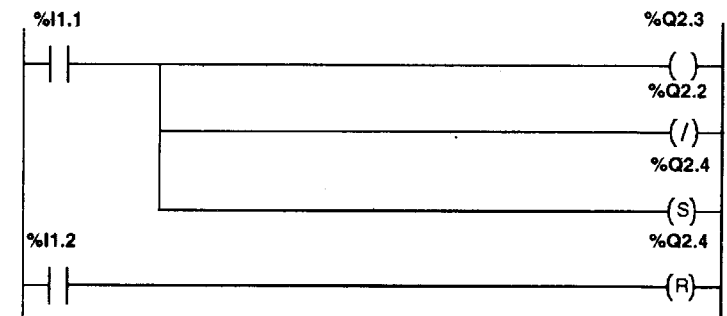
Instructions d'affectation

Rôle Le tableau suivant décrit le rôle de chacune des instructions.

Langage à contacts	Liste d'instructions	Littéral structuré	Description	Chronogramme
	ST	:=	aux bobines directes: l'objet bit associé prend la valeur du résultat de l'équation.	Opérande Résultat
	STN	:=NOT	aux bobines inverses: l'objet bit associé prend la valeur de l'inverse du résultat de l'équation.	Opérande Résultat
	S	SET	aux bobines à enclenchement: l'objet bit associé est mis à 1 lorsque le résultat de l'équation est à 1.	Opérande Résultat
	R	RESET	aux bobines à déclenchement: l'objet bit associé est mis à 0 lorsque le résultat de l'équation est à 1.	Opérande Résultat

Exemple en langage à contacts

L'exemple suivant montre la programmation des instructions d'affectation en langage à contacts.



Instruction ET Logique

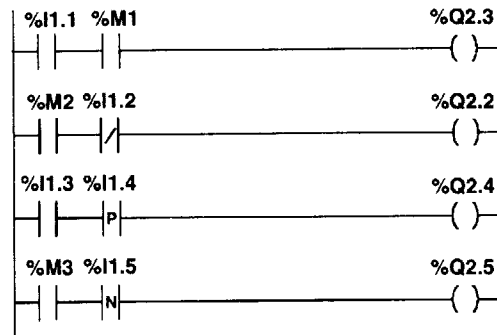
Rôle

Le tableau suivant décrit le rôle de chacune des instructions

Langage à contacts	Liste d'instructions	Littéral structuré	Description	Chronogramme
	AND	AND	ET logique entre l'opérande et le résultat booléen de l'instruction précédente	
	ANDN	AND (NOT...)	ET logique entre l'inverse de l'opérande et le résultat booléen de l'instruction précédente	
	ANDR	AND (RE...)	ET logique entre le front montant de l'opérande et le résultat booléen de l'instruction précédente (2) Mise à 1 pendant 1 cycle	
	ANDF	AND (FE...)	ET logique entre le front descendant de l'opérande et le résultat booléen de l'instruction précédente. (2) Mise à 1 pendant 1 cycle	

Exemple en langage à contact

L'exemple suivant montre la programmation des instructions ET Logique en langage à contacts.



Instruction OU Logique

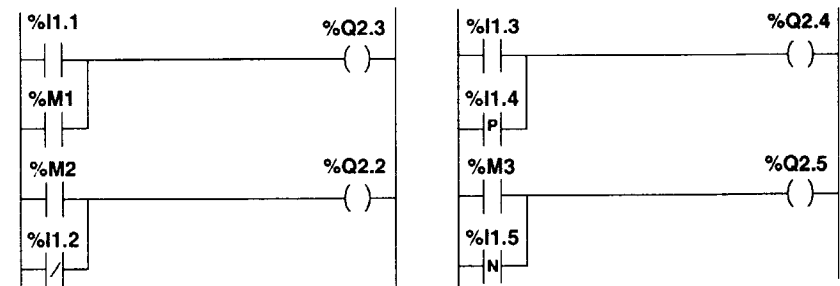
Rôle

Le tableau suivant décrit le rôle de chacune des instructions

Langage à contacts	Liste d'instructions	Littéral structuré	Description	Chronogramme
	OR	OR	OU logique entre l'opérande et le résultat booléen de l'instruction précédente	
	ORN	OR (NOT...)	OU logique entre l'inverse de l'opérande et le résultat booléen de l'instruction précédente	
	ORR	OR (RE...)	OU logique entre le front montant de l'opérande et le résultat booléen de l'instruction précédente	
	ORF	OR (FE...)	OU logique entre le front descendant de l'opérande et le résultat booléen de l'instruction précédente.	

Exemple en langage à contact

L'exemple suivant montre la programmation des instructions OU Logique en langage à contacts.



Présentation du bloc fonction temporisateur %TMI

Généralités

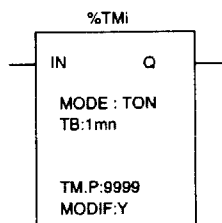
Le temporisateur a 3 modes de fonctionnement :

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

Les retards ou durées d'impulsion sont programmables et peuvent être modifiables ou non par terminal.

Illustration

La représentation graphique du bloc fonction temporisateur est la suivante :



Caractéristiques

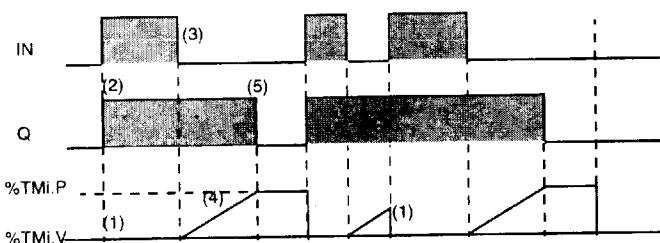
Le temporisateur possède les caractéristiques suivante :

Caractéristique	Repère	Valeur
Numéro temporisateur	%TMI	0 à 63 pour un TSX 37, 0 à 254 pour un TSX 57
Mode	TON	• retard à l'enclenchement (par défaut)
	TOF	• retard au déclenchement
	TP	• monostable
Base de temps	TB	1mn (par défaut), 1s, 100ms, 10ms (16 temporisateurs maxi à 10ms). Plus la base de temps est faible, plus la précision du temporisateur sera grande.
Valeur courante	%TMI.V	Mot qui croît de 0 à %TMI.P sur écoulement du temporisateur. Peut être lu, testé, mais non écrit par programme (%TMI.V peut être modifiée par terminal).
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 (instruction) "Armement"	IN	Sur front montant (mode TON ou TP) ou front "Armement" 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.

Fonctionnement du bloc fonction temporisateur %TMI en mode TOF

Généralités Le fonctionnement en mode TOF du temporisateur permet de gérer des retards au déclenchement.

Illustration Le chronogramme illustre le fonctionnement du temporisateur en mode TOF.



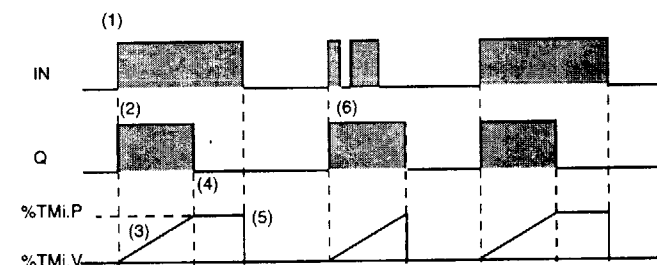
Fonctionnement Le tableau suivant décrit le fonctionnement du temporisateur en mode TOF.

Phase	Description
1	La valeur courante %TMI.V prend la valeur 0, sur un front montant de l'entrée IN (même si le temporisateur est en cours d'évolution)
2	Le bit de sortie %TMI.Q passe à 1.
3	Lors du front descendant sur l'entrée IN, le temporisateur est lancé.
4	La valeur courante croît vers %TMI.P d'une unité à chaque impulsion de la base de temps TB.
5	Le bit de sortie %TMI.Q retombe à 0 quand la valeur courante a atteint %TMI.P

Fonctionnement du bloc fonction temporisateur %TMI en mode TP

Généralités Le fonctionnement en mode TP du temporisateur permet de gérer d'élaborer une impulsion de durée précise (fonction monostable).

Illustration Le chronogramme illustre le fonctionnement du temporisateur en mode TP.



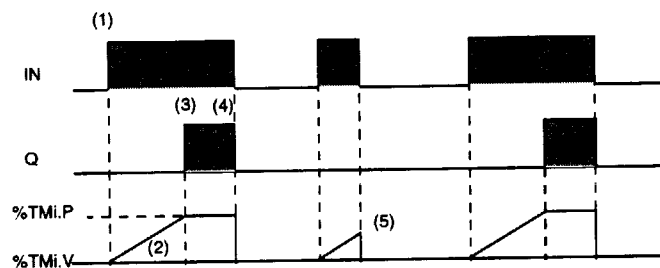
Fonctionnement Le tableau suivant décrit le fonctionnement du temporisateur en mode TP.

Phase	Description
1	Lors d'un front montant sur l'entrée IN, le temporisateur est lancé
2	Le bit de sortie %TMI.Q passe à 1
3	La valeur courante %TMI.V du temporisateur croît de 0 vers %TMI.P d'une unité à chaque impulsion de la base de temps TB
4	Le bit de sortie %TMI.Q retombe à 0 quand la valeur courante a atteint %TMI.P.
5	Quand l'entrée IN et la sortie %TMI.Q sont à 0, %TMI.V prend la valeur 0.
6	Ce monostable n'est pas réarmable.

Fonctionnement du bloc fonction temporisateur %TMI en mode TON

Généralités Le fonctionnement en mode TON du temporisateur permet de gérer des retards à l'encenchement.

Illustration Le chronogramme illustre le fonctionnement du temporisateur en mode TON.



Fonctionnement Le tableau suivant décrit le fonctionnement du temporisateur en mode TON.

Phase	Description
1	Lors d'un fron. montant sur l'entrée IN, le temporisateur est lancé
2	La valeur courante %TMI.V du temporisateur croît de 0 vers %TMI.P d'une unité à chaque impulsion de la base de temps TB
3	Le bit de sortie %TMI.Q passe à 1 dès que la valeur courante a atteint %TMI.P
4	Le bit de sortie %TMI.Q reste à 1 tant que l'entrée IN est à 1.
5	Quand l'entrée IN est à 0, le temporisateur est arrêté même s'il était en cours d'évolution : %TMI.V prend la valeur 0.

Programmation et configuration des blocs fonction temporisateur

Généralités La programmation des blocs fonction temporisateur est identique quel que soit le mode d'utilisation sélectionné.

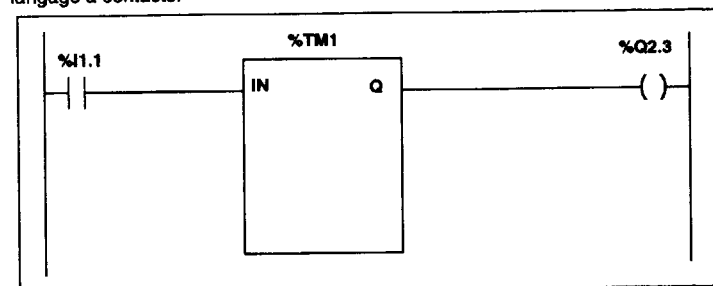
Le choix du fonctionnement TON, TOF ou TP s'effectue dans l'éditeur de variables.

Configuration Elle consiste à déterminer les paramètres suivants :

Paramètre	Valeurs
Mode	TON, TOF ou TP.
TB	1min, 1s, 100ms ou 10ms
%TMI.P	0 à 9999
MODIF	Y ou N

Programmation en langage à contacts

Le programme ci-après illustre l'utilisation d'un bloc fonction temporisateur en langage à contacts.



Instructions de comparaison

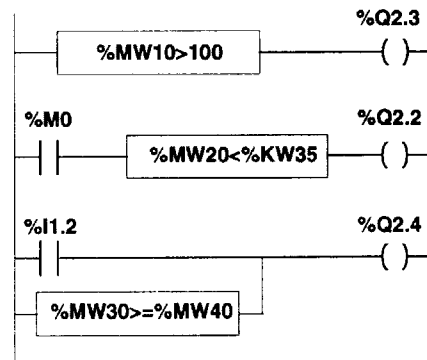
Généralités

Les instructions de comparaison permettent de comparer deux opérandes.

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

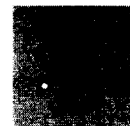
Structure

Langage à contacts



Note : Les blocs comparaison se programment en zone de test.

Adressage des objets langage associés aux métiers



Présentation

Objet de ce sous-chapitre Ce sous-chapitre présente l'adressage des objets langage associés aux modules métier.

Contenu de ce chapitre Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Adressage des objets de modules entrées/sorties en rack	22
Syntaxe	22
Exemples	23

Adressage des objets de modules d'entrées/sorties en rack

Présentation L'adressage des principaux objets bit et mot de modules d'entrées/sorties est de type géographique. C'est à dire qu'il dépend :

- du numéro (adresse) du rack,
- de la position physique du module dans le rack,
- du numéro de la voie du module.

Illustration L'adressage est défini de la manière suivante :



Syntaxe

Le tableau ci-dessous décrit les différents éléments constituant l'adressage.

Famille	Elément	Valeurs	Description
Symbole	%	-	-
Type d'objet	I	-	Image de l'entrée physique du module,
	Q	-	Image de la sortie physique du module, Ces informations sont échangées de manière automatique à chaque cycle de la tâche à laquelle elles sont attachées.
	M	-	Variable interne Ces informations de lecture ou d'écriture sont échangées à la demande de l'application.
	K	-	Constante interne Ces informations de configuration sont disponibles en lecture seulement.
Format (taille)	X	-	Booléen Pour les objets de type booléen, cet élément peut être omis.
	W	16 bits	Simple longueur.
	D	32 bits	Double longueur.
	F	32 bits	Flottant. Le format flottant utilisé est celui de la norme IEEE Std 754-1985 (équivalent IEC 559).
Adresse rack	x	0 ou 1 0 à 7	TSX 5710/102/103/153, PMX 57102, PCX 571012). Autres processeurs.
Position module	y	00 à 14 (1)	Numéro de position dans le rack. Lorsque le numéro de rack (x) est différent de 0, la position (y) est codée sur 2 digits : 00 à 14 ; par contre si le numéro de rack (x) = 0, on élimine les zéros non significatifs (élimination par la gauche) de "y" ("x" n'apparaît pas et "y" est sur 1 digit pour les valeurs inférieures à 9).
N° voie	i	0 à 127 ou MOD	MOD : voie réservée à la gestion du module et des paramètres communs à toutes les voies.
Rang	r	0 à 127 ou ERR	Position du bit dans le mot. ERR : indique un défaut module ou voie.
(1) : le nombre d'emplacements maximum nécessite l'utilisation de 2 racks à la même adresse.			

Objets associés aux métiers

Exemples

Le tableau ci-dessous présente quelques exemples d'adressage d'objets.

Objet	Description	Illustration
%MW2.0.3	Mot d'état de rang 3 de la voie 0 du module d'entrées TOR situé à la position 2 du rack 0.	<p>The diagram illustrates three racks, labeled 0, 1, and 2 on the left. Each rack contains five slots, numbered 0 to 4 at the top. Rack 0: Slot 0 contains a 'PULSE' module; Slot 1 contains a 'TOR' module; Slot 2 contains a 'Loop' module; Slot 3 contains a 'COMM' module. Rack 1: Slot 3 contains an 'ANALOG' module; Slot 4 contains a 'DIGITAL' module. Rack 2: Slot 4 contains an 'ANALOG' module.</p>
%MW103.0.3	Mot d'état de rang 3 de la voie 0 du module de sorties TOR situé à la position 3 du rack 1.	
%I102.MOD.ERR	Information de défaut du module d'entrées analogiques situé à la position 2 du rack 1.	
%I204.3.ERR	Information de défaut de la voie 3 du module de sorties analogiques situé à la position 4 du rack 2.	