

## **Ladder et instructions automates**

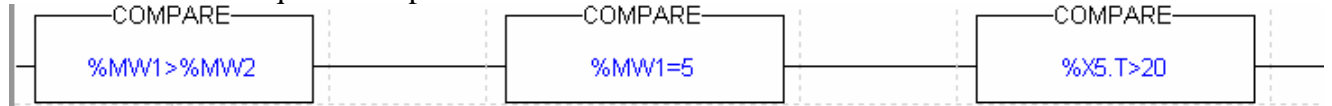
Ladder et instructions automates.....	1
Les instructions automates .....	2
Comparaison.....	2
Opérations sur Mots .....	3
Opérations arithmétiques.....	3
Opérations sur chaînes de bits .....	3
Décalage circulaire ou non circulaire logique.....	4
Opérations logiques.....	6
Transcodage .....	6
Les fonctions standards à une seule variable numérique (réel).....	6
Exercices associés .....	7
Le Ladder .....	8
L'algorithme.....	8
Structure itérative .....	9
Les tests .....	10
Exercice d'application du ladder N°1 .....	12
Exercice ladder N°2 .....	13
Exercice Ladder : Positionnement à l'aide d'un codeur incrémental.....	14
Exercice .....	14
Programmation ladder d'un conversion Gray binaire .....	15
Acquisition de 4 roues codeuses branchés sur les mêmes entrées .....	16
Extrait BTS sujet 2000 .....	17
Pile Fifo Lifo .....	18
Communication réseaux entre 2 TSX 47 .....	19
Performance du réseau automate : Unitelway sur TSX .....	21

# Les instructions automatées

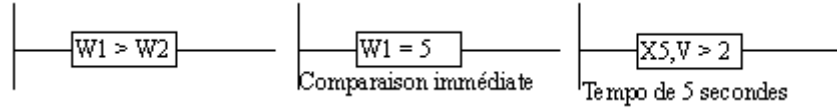
## Comparaison

La comparaison est située dans la zone de test. (Comme le contact ouvert et fermé)

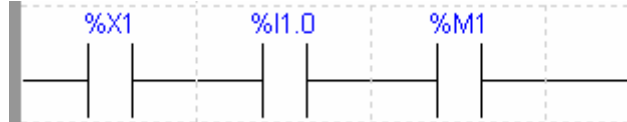
La sortie est à 1 lorsque la comparaison est vraie.



sur TSX17 : (PL7-2)



Pour rappel la comparaison sur un bit, c'est forcément un contact. (Même un bit extrait d'un mot)



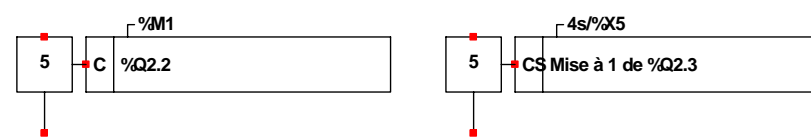
Equivalence Tsx17 : X1, I1,0 et B1

Equivalence Mitsubishi : ..., X1, Y1, M1

### Exercice :

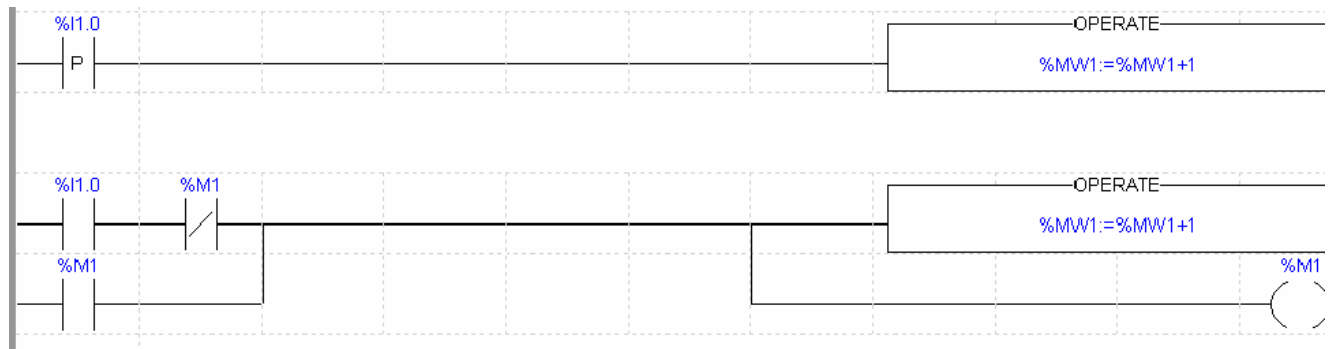
1) Action conditionnelle

Ecrire en ladder l'action suivante :



2) Ecriture d'un front montant sur TSX Micro et TSX 17

Ecrire en ladder l'incrémenter du mot %MW1 lors de l'apparition de %I1.0



Dessiner le chronogramme des variables %I1.0, %M1 et donner l'évolution du mot %MW1

## Opérations sur Mots

Mot ou mot indexé ou valeur immédiate vers mot ou mot indexé

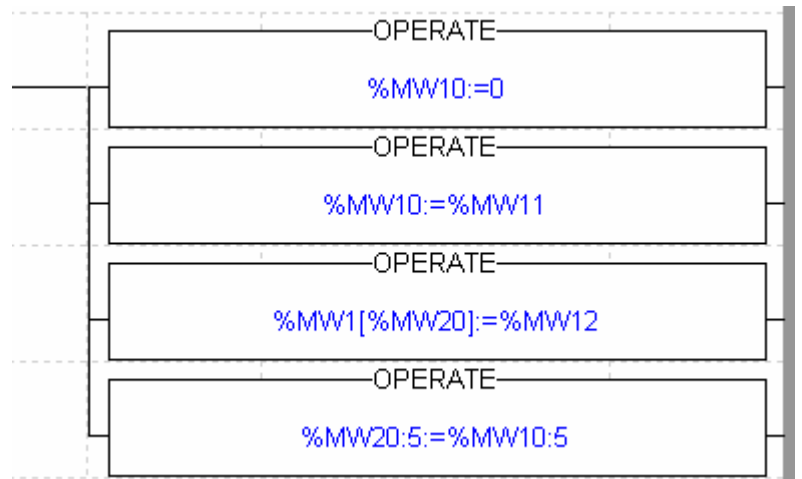
Table de mots vers table de mots

%Mw10 est mis à 0

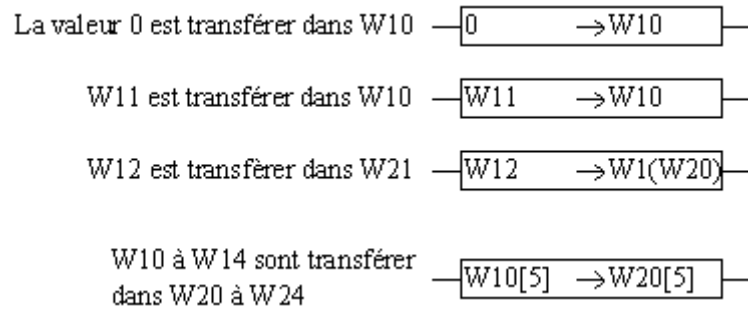
%MW10 prend la valeur de %MW11

Le mot %MWxx prend la valeur de %MW12 avec  $xx=1+\%Mw20$

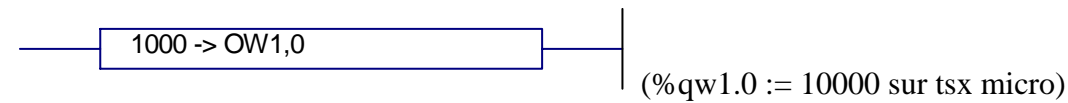
Les mots de %MW20 à %MW24 prennent la valeur des mots de %MW10 à %MW15



L'écriture sur TSX17 est la suivante,



Rappel : Pour la sortie analogique N°0 du module en position 1, que l'on veut mettre à 10 volt on a :



## Opérations arithmétiques

Ces instructions sont: +, -, \*, /, REM (pour le reste d'une division)..

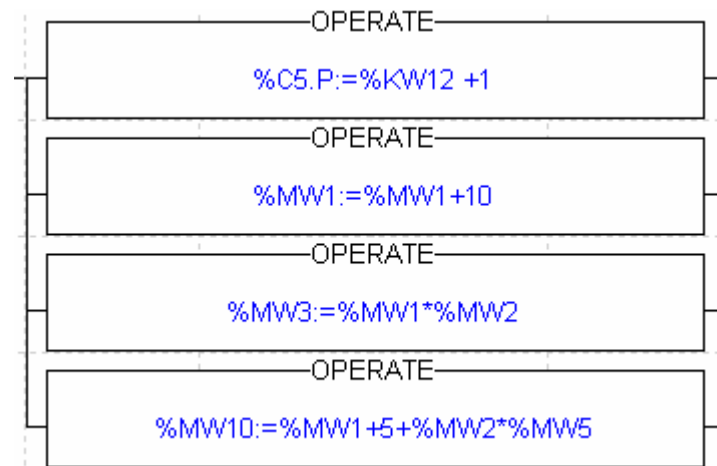
L'opération est réalisée dans un bloc OPERAT.

La valeur a atteindre du compteur 5 sera de %KW12 +1,

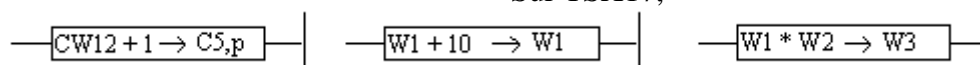
Addition,

Multiplication

Formule complexe.

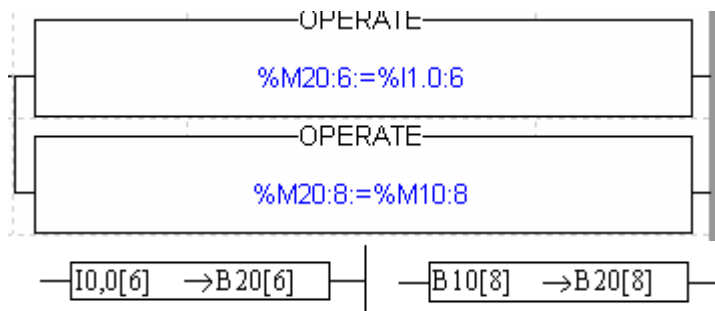


Sur TSX17,

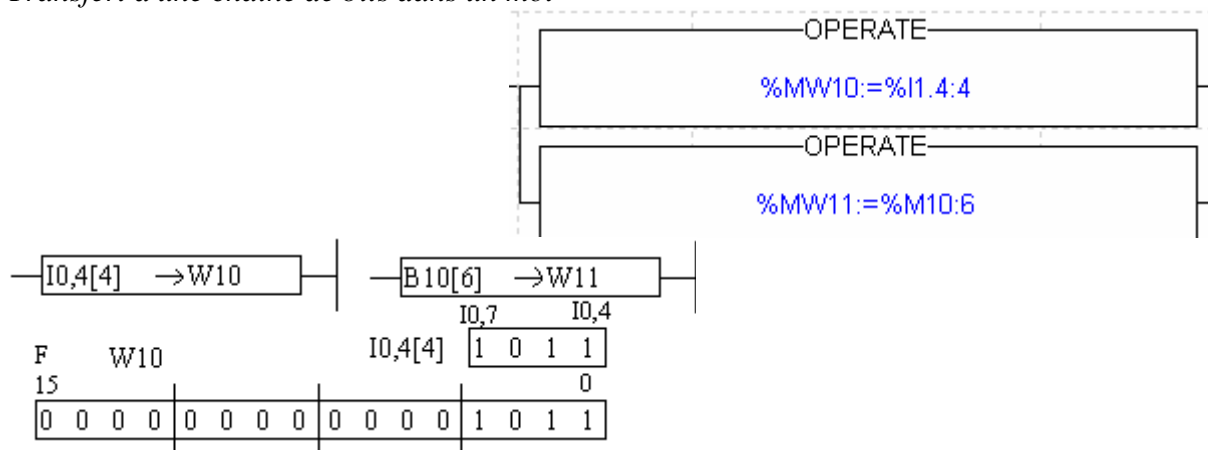


## Opérations sur chaînes de bits

Transfert d'une chaîne de bits dans une autre



Transfert d'une chaîne de bits dans un mot

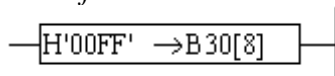


Transfert d'un mot dans une chaîne de bits



Les 16 bits du mot W1 sont transférés dans les bits B0 à B16 (Le bit de poids faible de W1 dans B0)

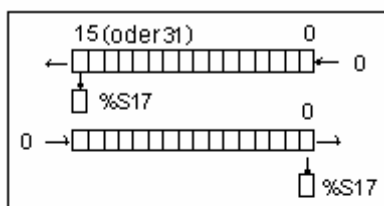
Transfert d'une valeur immédiate dans une chaîne de bits



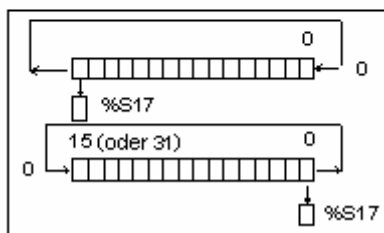
Règle: Le nombre de bits à transférer identiques aux nombres de bits destinataires.

## Décalage circulaire ou non circulaire logique

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

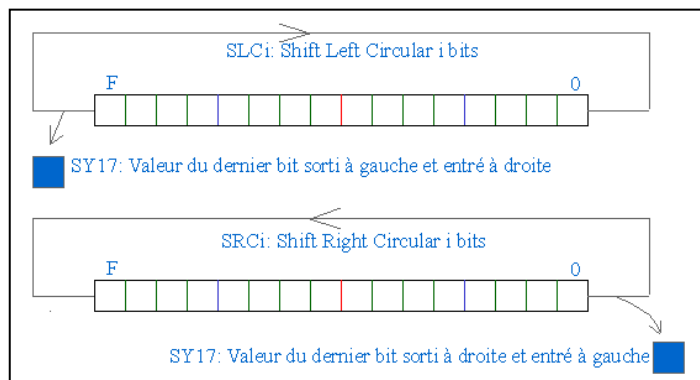


Tsx Micro (ci contre) **le décalage logique** :  
 SHL(op2,i) décalage logique à gauche de i positions.  
 SHR(op2,i) décalage logique à droite de i positions.



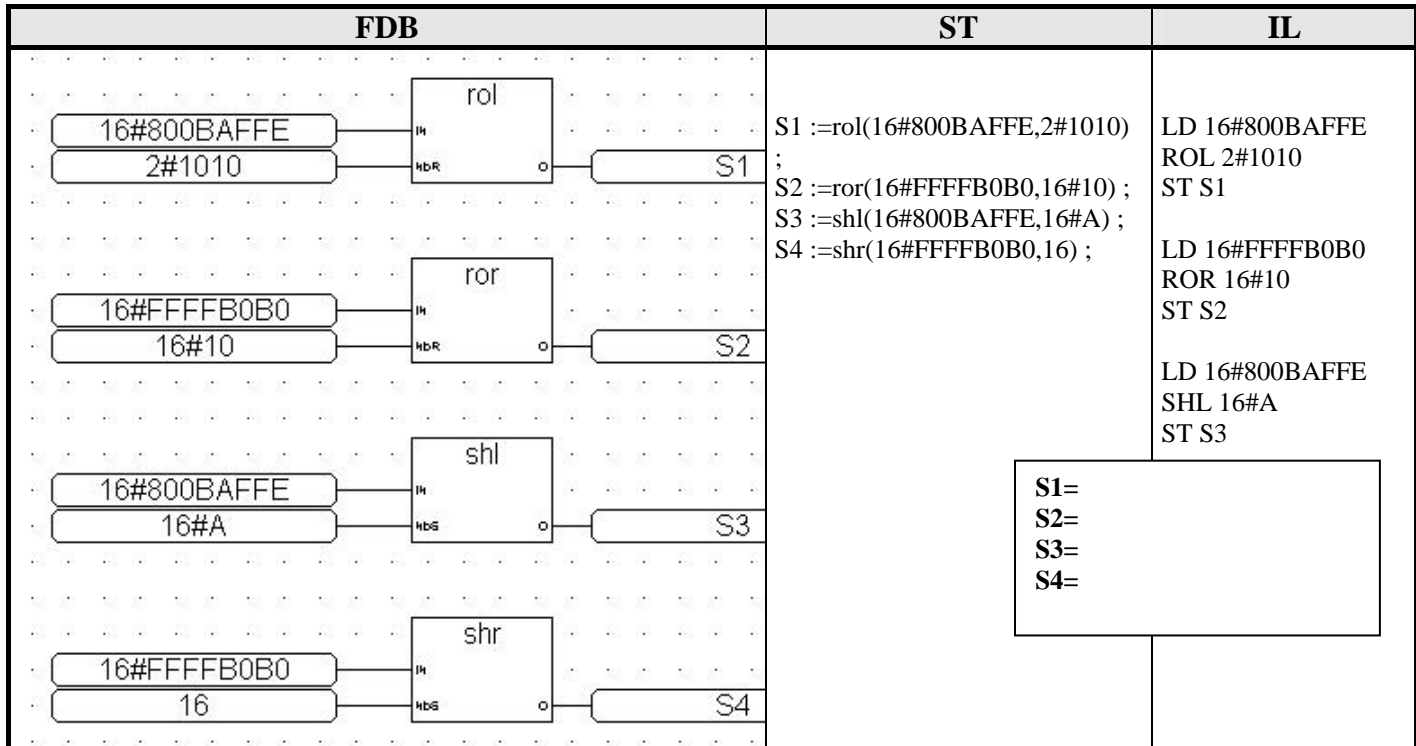
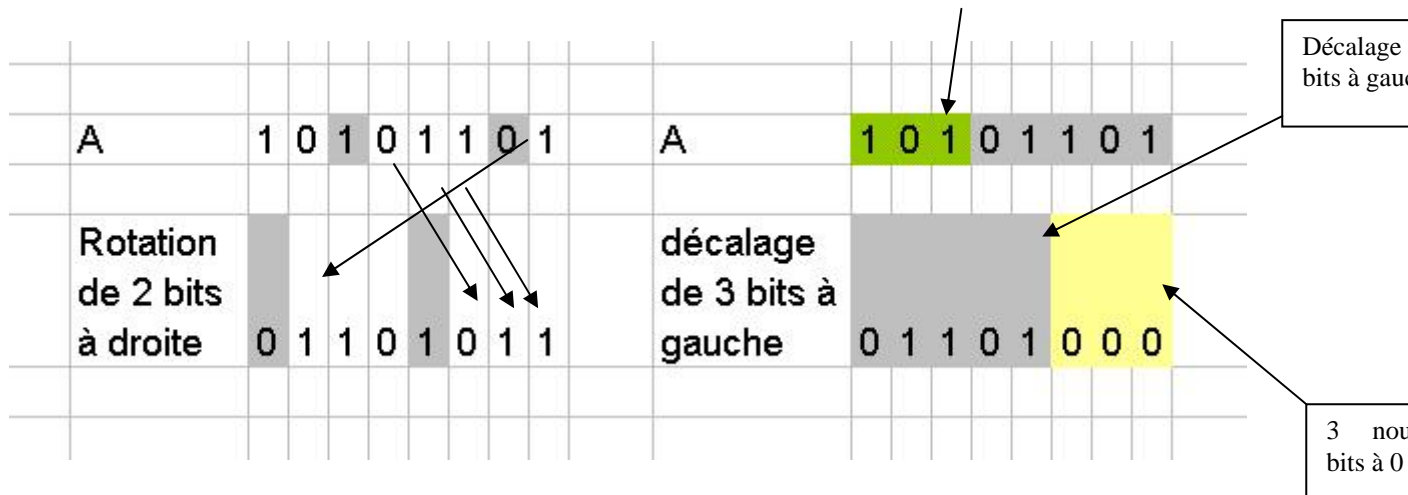
### Le décalage circulaire

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



Exemple sur un registre 8 bits

Ces 3 bits sont perdus



## Opérations logiques

Les opérations logiques sont réalisés bits par bits.  
AND, OR, XOR (ou exclusif), CPL (Complément)

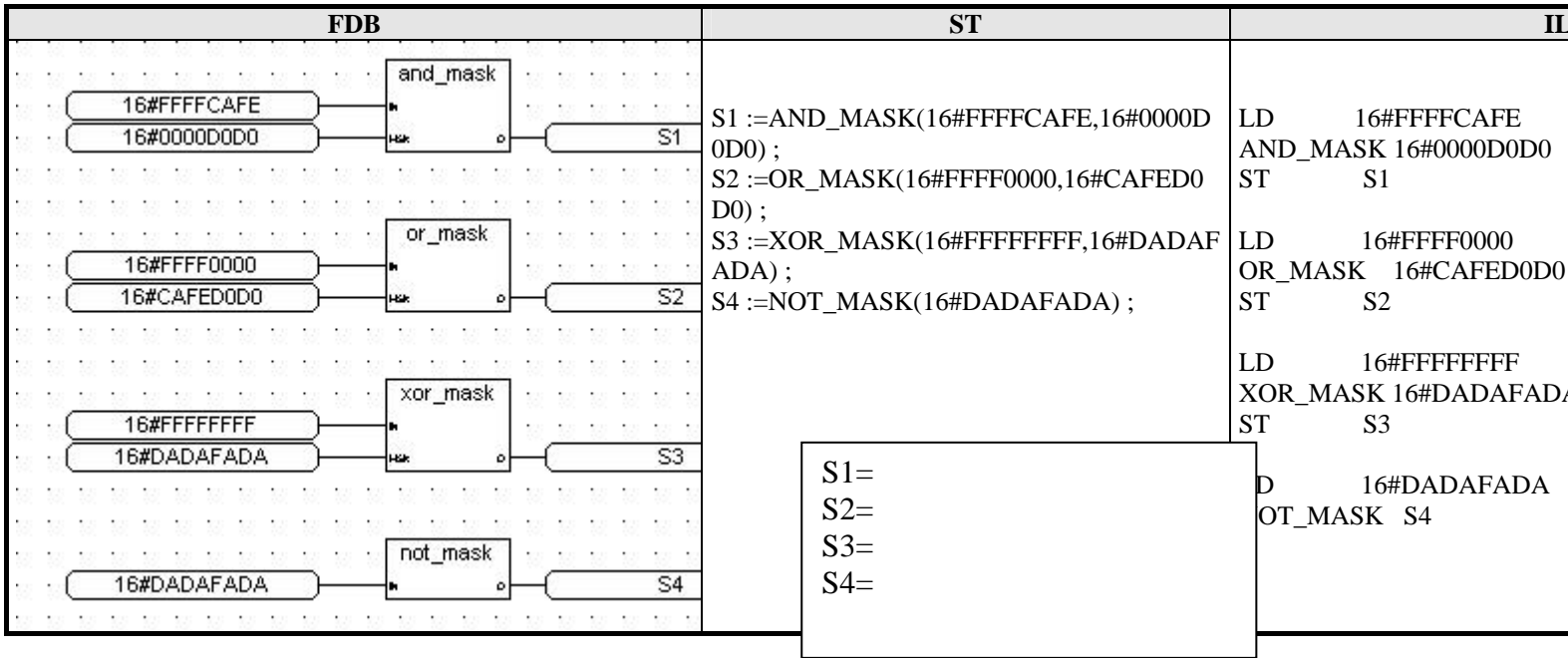
A	1 0 1 0 1 1 1 1	A	1 0 1 0 1 1 1 1
B	0 1 1 0 1 1 0 0	B	0 1 1 0 1 1 0 0
<b>A OR B</b>	<b>1 1 1 0 1 1 1 1</b>	<b>A XOR B</b>	<b>1 1 0 0 0 0 1 1</b>
A	1 0 1 0 1 1 1 1	A	1 0 1 0 1 1 1 1
B	0 1 1 0 1 1 0 0	<b>NOT A</b>	<b>0 1 0 1 0 0 0 0</b>
<b>A AND B</b>	<b>0 0 1 0 1 1 0 0</b>		

Ce sont des opérations de masquages bit à bit entre 2 entiers.

### Utilisation Principale Le MASQUE d'une partie d'un mot.

W1 (1100 1100 1101 1011) AND W2 (0000 0000 1111 1111) → W3 (0000 0000 1101 1011)

A retenir: ET Forçage à 1 (pour le bit 0, masque '0000 0000 0000 0001)  
OU Forçage à 0 (pour le bit 0, masque '1111 1111 1111 1110)



## Transcodage

BIN: Transcodage BCD vers binaire  
BCD: Transcodage binaire vers BCD  
ATB: Transcodage ASCII vers binaire  
BTA: Transcodage binaire vers ASCII

## Les fonctions standards à une seule variable numérique (réel)

**Fonctions mathématiques**

**Fonctions trigonométriques**

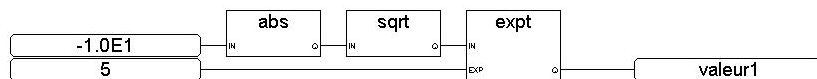
ABS Valeur absolue  $|x|$   
 SQRT Racine carrée  $\sqrt{x}$   
 EXPT Exponentiation  $y^x$  et  $x$  est entier !  
 LOG Logarithme  $\log(x)$

Sortie dans l'intervalle [-1.0 ,+1.0]  
 COS Cosinus  
 SIN Sinus  
 TAN Tangente  
 Entrée dans l'intervalle [-1.0 ,+1.0]  
 Sortie dans l'intervalle [0.0, PI]  
 ACOS Arc cosinus  
 ASIN Arc sinus  
 ATAN Arc tangente

### FBD

### ST

### IL



Valeur1:=expt(sqrt(abs(-1.0E1)),5);

LD -1.0E1  
 ABS  
 SQRT  
 EXPT 5  
 ST Valeur1

Valeur1 =

## Exercices associés

Masque

Multiplication et division par multiple de 2

Décalage sur 32 bits

Addition sur 32 bits avec SY17 débordement (65536) (SY18 bit de signe)

Roues codeuses

## Passage du Binaire à l'Hexadécimal, au code Gray

Le gray est utilisé pour les capteurs absolus, pour passer d'une ligne à l'autre on ne peut changer qu'un bit à la fois

Déc	Hex	Bin	Gray	Déc	Hex	Bin	Gray
0	0	0000	0000	8	8	1000	1100
1	1	0001	0001	9	9	1001	1101
2	2	0010	0011	10	A	1010	1111
3	3	0011	0010	11	B	1011	1110
4	4	0100	0110	12	C	1100	1010
5	5	0101	0111	13	D	1101	1011
6	6	0110	0101	14	E	1110	1001
7	7	0111	0100	15	F	1111	1000

Un chiffre Hexadécimal équivaut à 4 bits binaires, pour convertir du binaire en hexadécimal il faut associer les bits 4 par 4

Soit: 0101 0011 0001 1000

Donne: 5 3 1 8

# Le Ladder

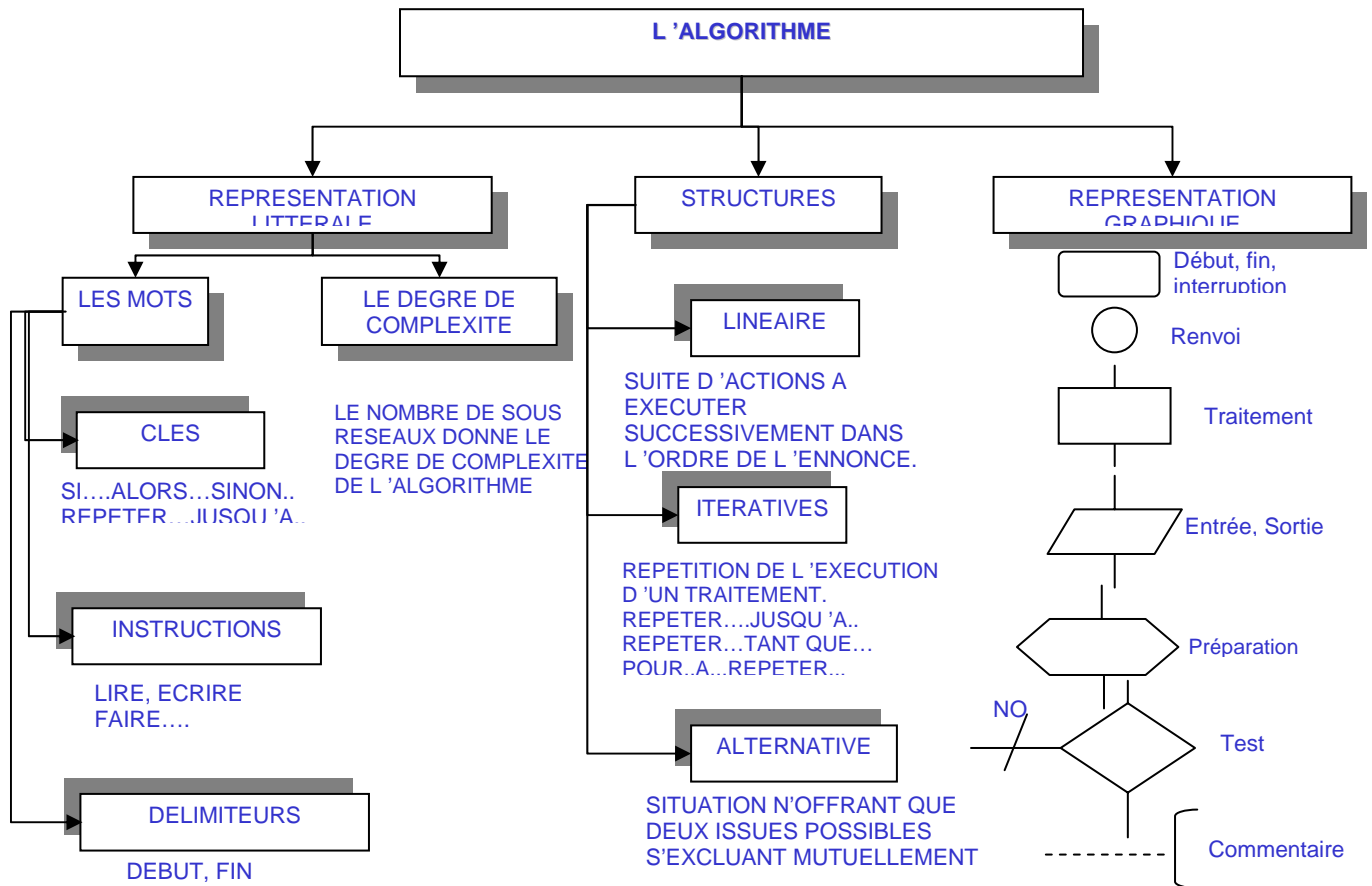
## L'algorithme

### Qu'est-ce que c'est ?

Un algorithme est un ensemble de règles opératoires rigoureuses ordonnant à un processeur particulier d'exécuter, dans un ordre déterminé, un nombre fini d'opérations élémentaires pour résoudre tous les problèmes d'un type donné (NF Z 61-100)

Un algorithme peut être **représenté** :

- soit littéralement grâce au **langage algorithmique**
- soit graphiquement à l'aide de l'**algorithme** (ou **organigramme**)



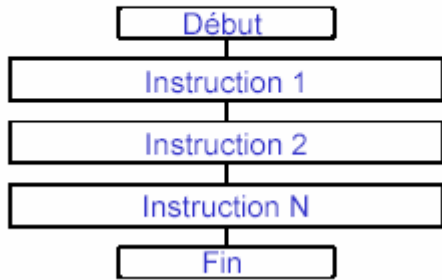
### La structure linéaire



REPRESENTATION LITTERALE

- début
- instruction 1
- instruction 2
- .....
- instruction N
- fin

REPRESENTATION GRAPHIQUE

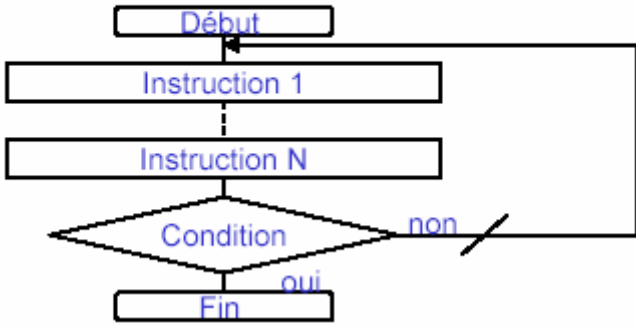


Structure itérative

REPRESENTATION LITTERALE

- début
- **répéter**
- instruction 1
- .....
- instruction N
- **jusqu 'à « condition vraie »**
- fin

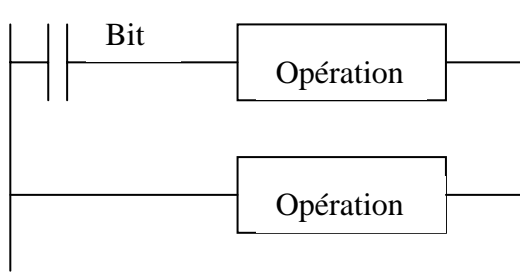
REPRESENTATION GRAPHIQUE



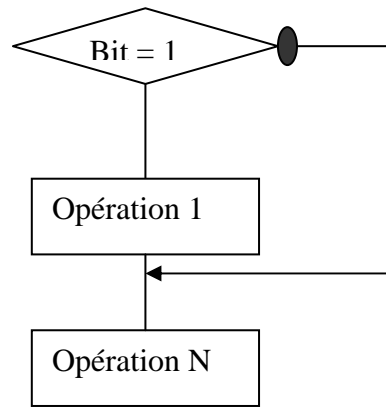
## Les tests

### Test sur 1 bit, en Pascal :

Si Bit = 1 alors Opération 1 finsi  
Opération N



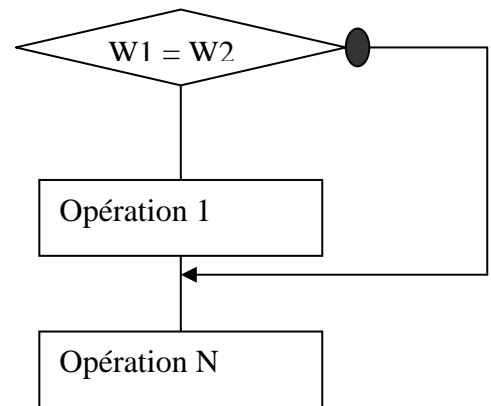
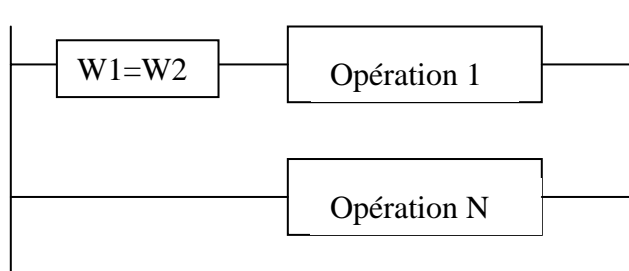
en Ladder :



### Test sur des mots, en Pascal :

Si W1=W2 alors Opération 1 finsi  
Opération N

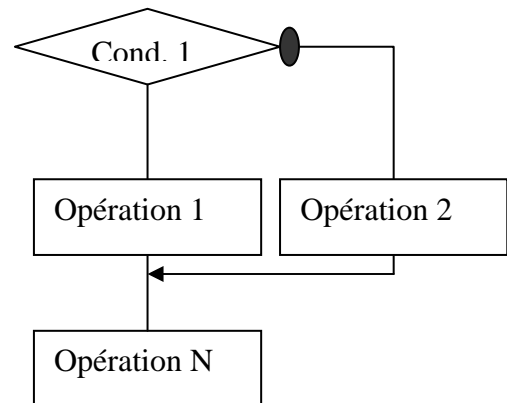
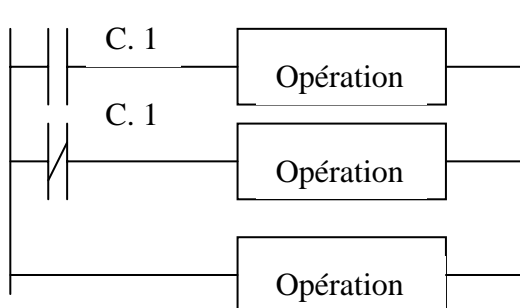
en Ladder :



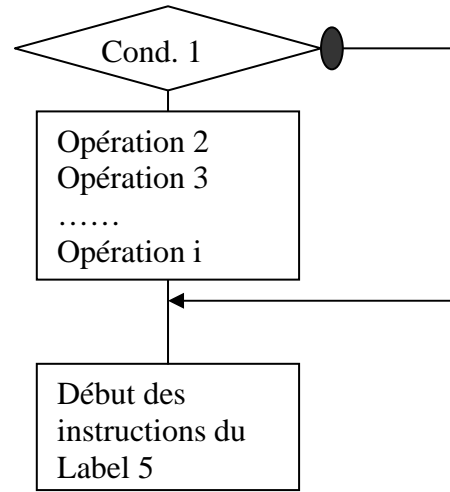
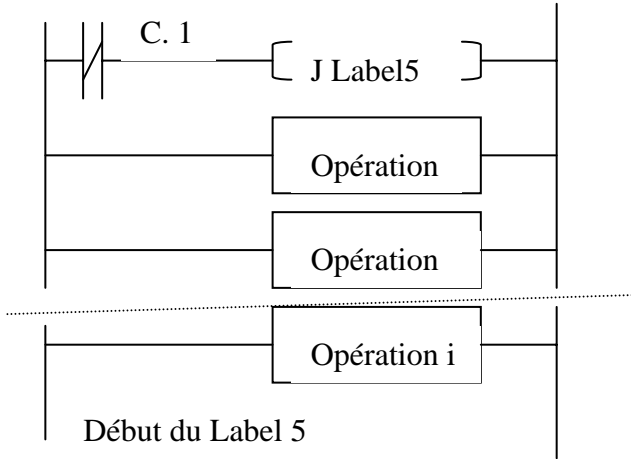
### Test avec 2 opérations, en Pascal :

Si Condition 1 alors Opération 1 sinon Opération 2 finsi

En Ladder :



**Test avec une série d'opération en pascal:**  
 Si Condition 1 alors Opération 2 à Opération i finis

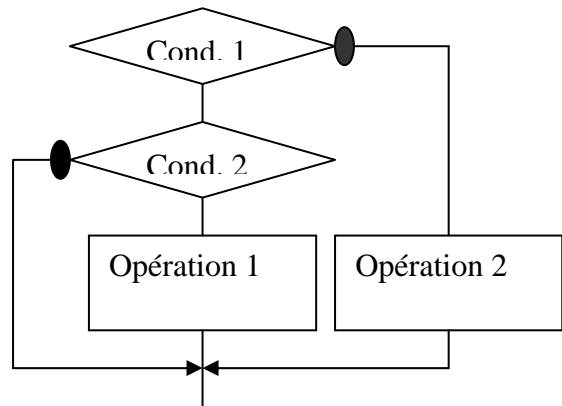
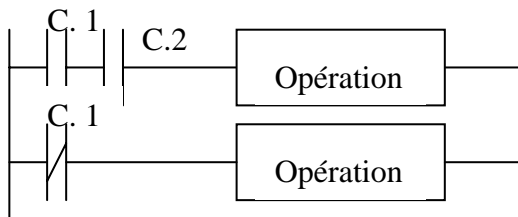


**En ladder:**

**Plusieurs Test les uns à la suite des autres en pascal :**

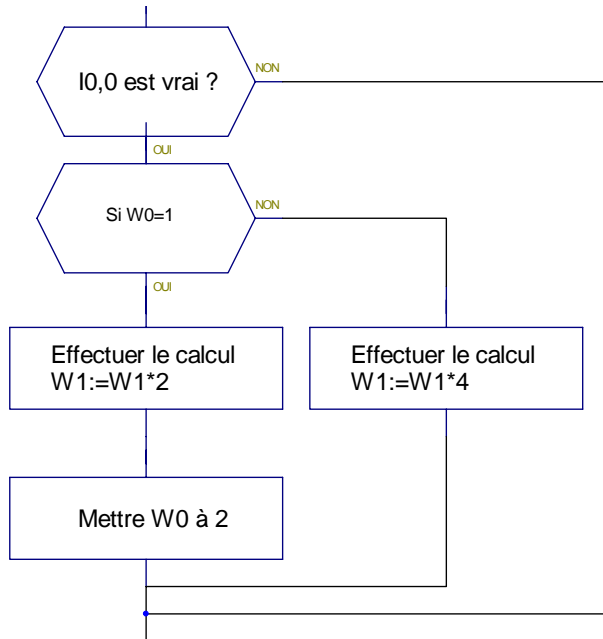
Si Condition 1 alors  
     Si Condition 2 alors Opération 1 finis  
     Sinon  
     Opération 2  
     Finsi

**En Ladder :**

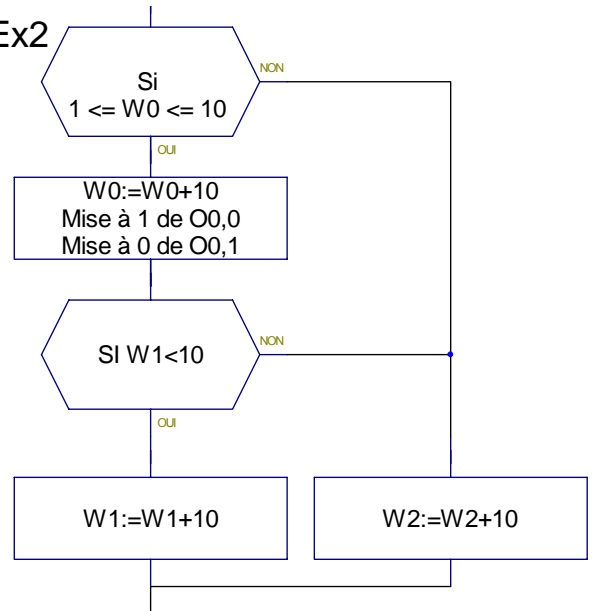


## Exercice d'application du ladder N°1

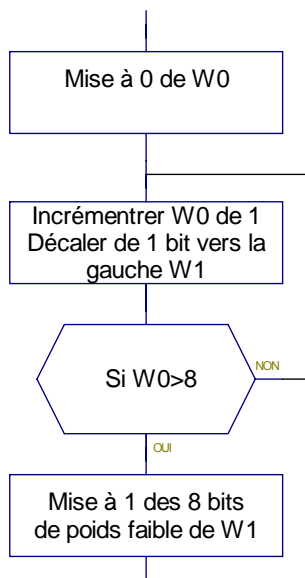
Ex1



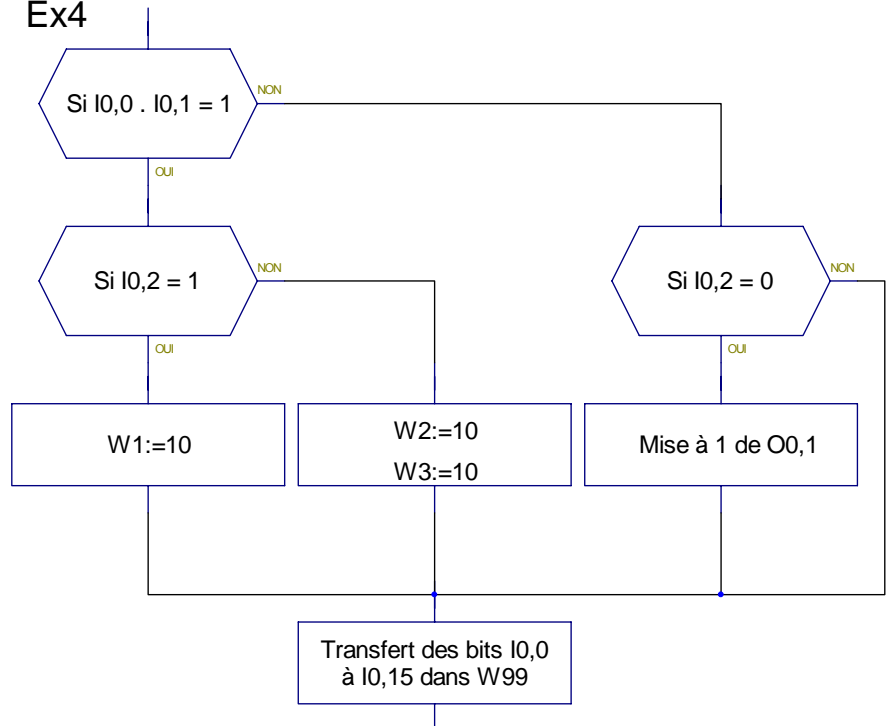
Ex2



Ex3



Ex4

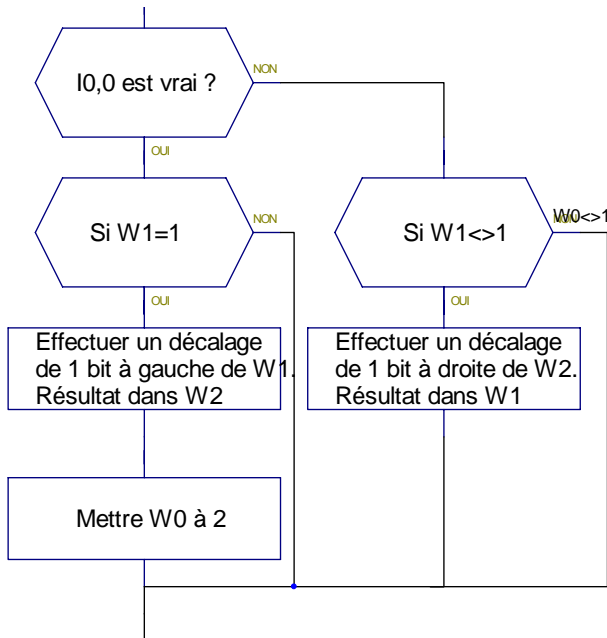


Ex5

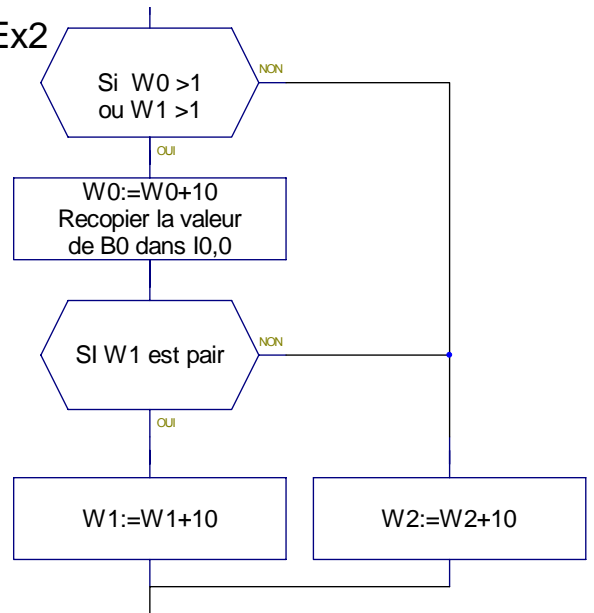
Réaliser l'algorithme, ainsi que le programme ladder qui met à 0 la zone mémoire allant de W10 à W49

## Exercice ladder N°2

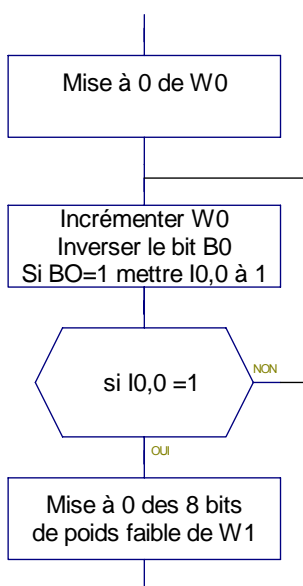
Ex1



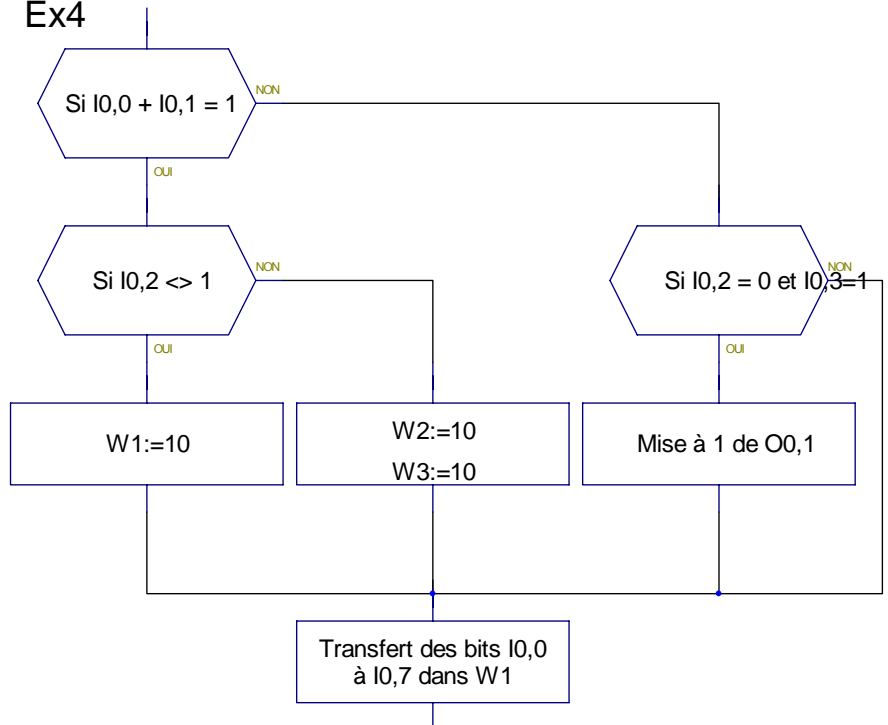
Ex2



Ex3



Ex4



Ex5

Réaliser le programme ladder qui met les mots pair à 0 et les mots impairs à 1, la zone de mot ira de W0 à W99.

donc W0=0, W1=1, W2=0, W3=1

## Exercice Ladder : Positionnement à l'aide d'un codeur incrémental

On désire réaliser le programme ladder de calcul de positionnement.

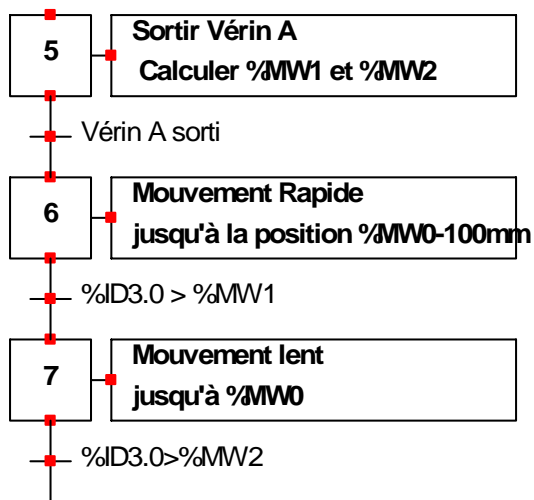
Nous avons un codeur incrémental 1024 points monté sur l'axe d'une poulie dont la développée (périmètre) est de 150mm.

Cette poulie entraîne un chariot qui se déplace sur une glissière.

L'opérateur a entré la position à obtenir dans le mot `%MW0` qui est en mm.

La position du chariot est contenu dans le mot double `%ID3.0` qui correspond à la carte de comptage, sur laquelle on a branché le codeur.

Réaliser le programme permettant de calculer `%MW1` et `%MW2`



## Exercice

Voir feuille correction TP pl7 micro

Calcul de base. Si `%M1=1` alors `%MW1:=1` et `%MW2:=%MW1*%MW1`

Calcul avec un test pour utiliser le théorème de Morgan Si `%M2=1` ou `%M3=1` alors `%MW3:=%MW2` sinon `%MW3:=-%MW2`

SI `%M4=1` alors mettre la table mémoire `%MW50` à `%MW59` à la valeur 5, avec comme mot index `%MW5`

Exercice sur l'hystérésis

## ***Programmation ladder d'un conversion Gray binaire***

**Objectif:** Convertir la valeur Gray contenue dans le mot MGB en binaire dans le mot MBB.

A partir du texte réaliser le programme automate pour TSX Micro.

Légendes

- MGB: Mots 16 bits en Gray (donnée entrante)
- MBB: Mots binaire (donnée sortante, résultat)
- AUX G: Mot permettant d'isoler un bit Gray
- AUX B: Mot destiné à calculer le bit binaire

Initialisation Mettre à 0 les variables MBB, AUX G, AUX B, I.

### **Corps principale**

Tant que I <> 16 alors

- Décaler MBB de 1 bit à gauche
- Recopier le bit MGB de poids fort(bit F) dans le bit de poids faible de AUX G (bit 0)
- Décaler MGB d'un bit vers la gauche
- $MBB := AUX\ B\ XOR\ AUX\ B$
- $AUX\ G := 0$
- $I := I + 1$

Fin de tant que.

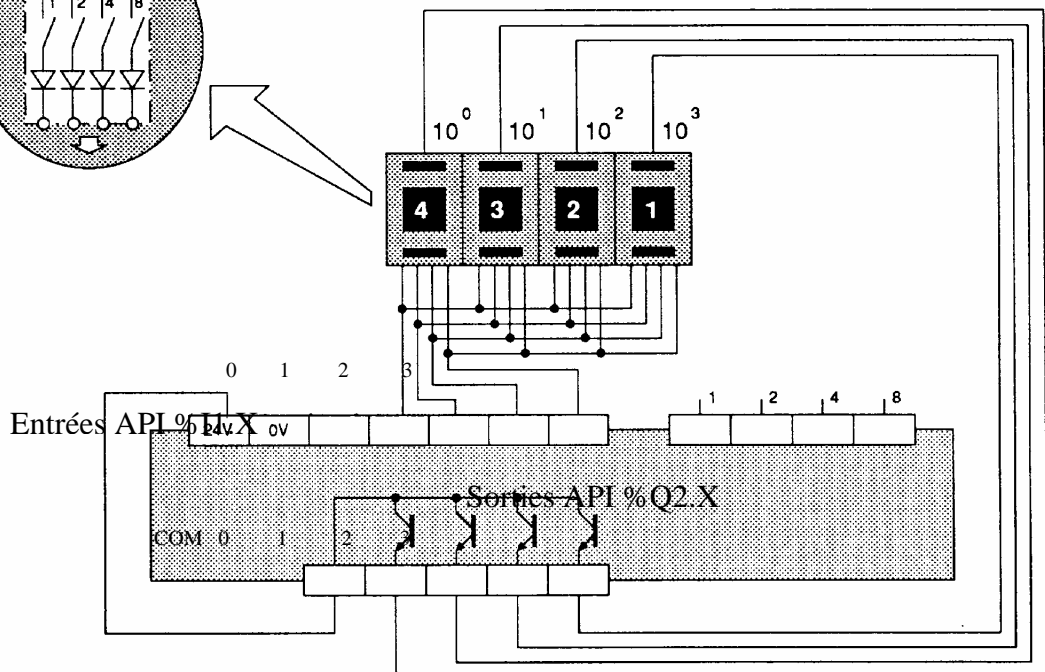
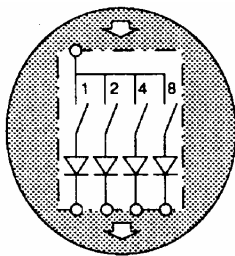
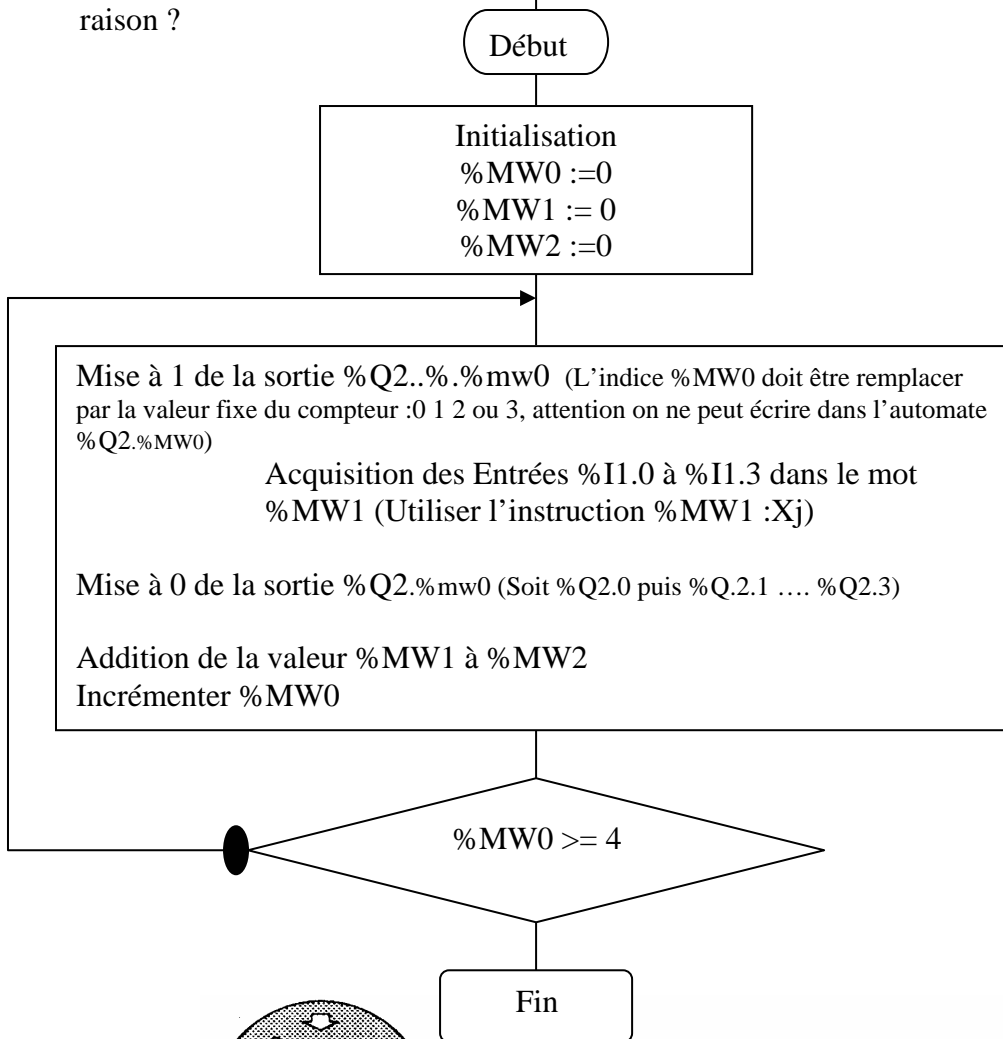
**Résultat :** MBB contient la valeur binaire.

## Acquisition de 4 roues codeuses branchés sur les mêmes entrées

Réaliser le programme ladder correspondant à l'organigramme ci dessous.

Pour information : %MW0 correspond à un compteur, %MW2 à la somme des 4 roues codeuses.

Ce programme de la façon ou il est écrit ne peut fonctionner, pouvez vous en donner la raison ?





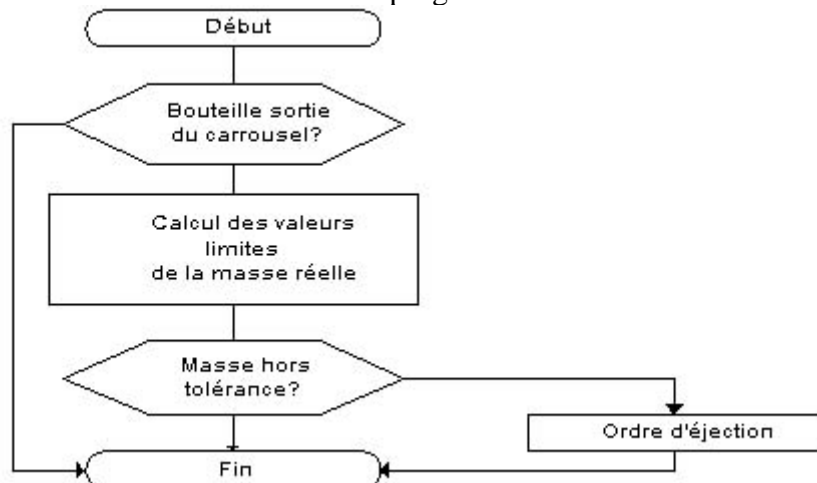
## Extrait BTS sujet 2000

Le poste permet de contrôler la conformité de la masse de gaz après le remplissage.

On tolère  $\pm 50$  kg par rapport à la masse totale théorique (PTT)

La valeur traitée par l'automate correspond à la masse en kg multipliée par 100.

L'algorithme suivant décrit ce sous programme de contrôle de masse:



Question: En fonction de l'adressage des entrées, des sorties et des variables internes proposé écrire le sous programme correspondant à l'algorithme, en langage ladder normalisé (norme CEI 1131-3).

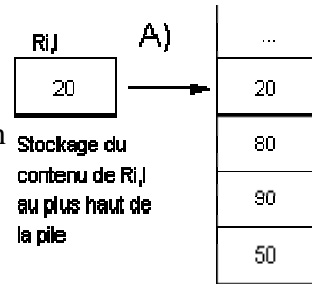
Adresse des entrées	
Détection bouteille sortie carrousel	%I1.3
Détection bouteille sur contrôle de masse	%I1.4
Adresse des sorties	
Commande d'éjection après contrôle de masse	%Q2.1
Adresse des variables internes	
Adresse de la valeur du PTT	%MW0
Adresse de la valeur de la masse réelle mesurée	%MW10
Adresse de la valeur supérieure admissible	%MW1
Adresse de la valeur inférieure admissible	%MW2

Complément: Le & quitte %I1.4 a été ajouté au sujet d'origine.

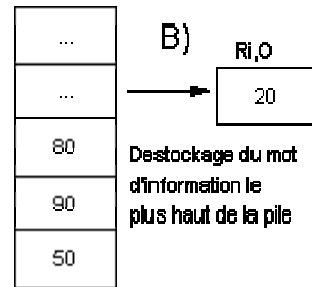
# Pile Fifo Lifo

## File d'attente (premier entré, premier sorti) ou ou pile FIFO (First in, First out)

FIFO, La première information entrée est la première à être déstockée. Lorsqu'une demande de stockage est prise en compte le contenu du mot d'entrée  $R_{i,I}$  préalablement chargé est stocké au plus haut de la file. Lorsqu'une demande de destockage est prise en compte le mot d'information le plus bas de la file est rangé dans le mot de sortie  $R_{i,O}$ .



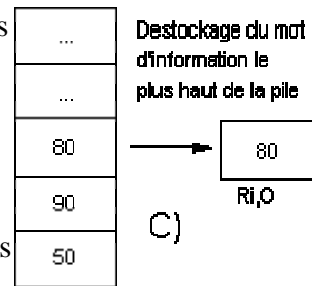
LIFO, la dernière information entrée est la première à être déstockée. Lorsqu'une demande de stockage est prise en compte (front montant sur l'entrée I), le contenu du mot d'entrée  $R_{i,I}$  préalablement chargé, est stocké au plus haut de la pile (fig A).



Lorsqu'une demande de destockage est prise en compte (front sur l'entre O) le mot d'information le plus haut (dernière information entrée) est rangée dans le mot  $R_{i,O}$  (fig B puis fig C).

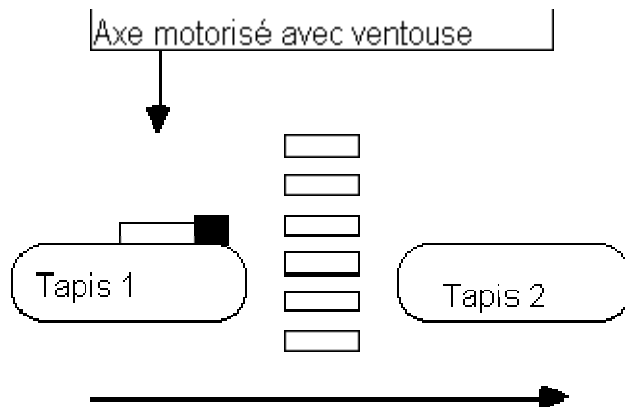
## Application concrète

Un tapis 1 amène des pièces dont le type est codé 16 bits, les pièces arrivent suivant la cadence de la machine qui diffère suivant les pièces (présence d'une nouvelle pièce lorsque pp est actif). Idem pour le tapis 2, qui récupère ces pièces à la condition que la machine 2 soit disponible (information: demande).



Un bras manipulateur assure le stockage des pièces en les empilant

les unes sur les autres.



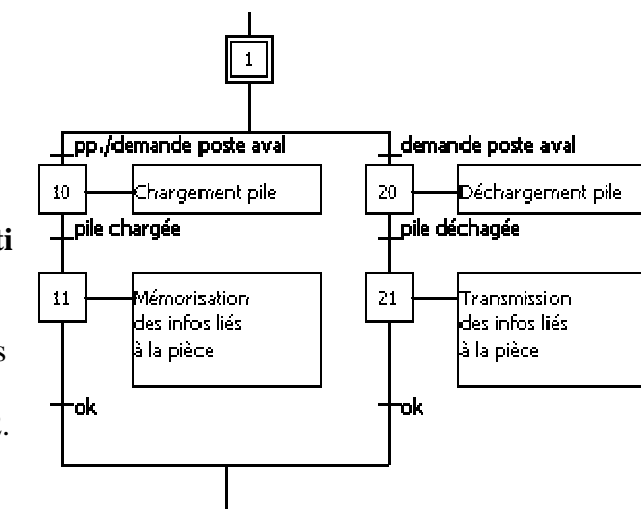
Il s'agit ici de réaliser le programme qui va stocker les informations.

on:

- W0 contient le type de la pièce arrivant sur le tapis 1.
- W1 contient le type de la pièce allant sur le tapis 2.

Réaliser le traitement post des étapes X11 et X21.

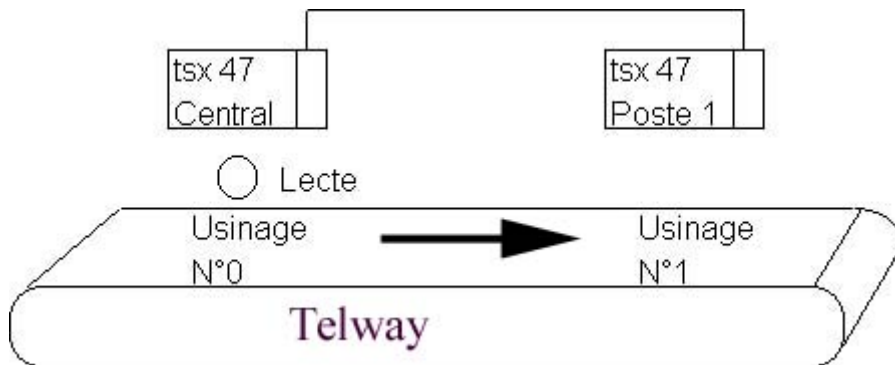
Affectati



## Communication réseaux entre 2 TSX 47

Utilisation des mots Comi,0

Le tapis tourne en continu, une butée est présente à chaque poste qui est commandé par chacun des automates. Lorsque le poste 1 a fini son usinage il autorise le poste central à lui faire parvenir la prochaine pièce ainsi que les informations qui lui sont nécessaires.



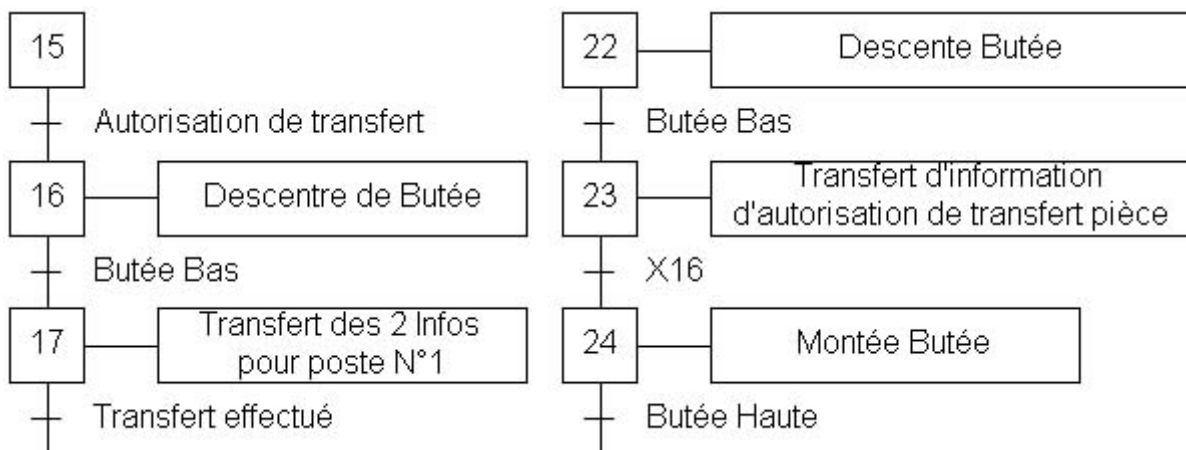
### Poste central: (à gauche sur le schéma)

- Lecture des informations de la pièce.
- Usinage de Type 0
- Transmissions de 2 informations: Type de pièce (5 pièces possible) et de la matière de la pièce à usiner (Aluminium, Plastique, acier)

### Poste 1:

- Usinage de type 1, correspondant aux informations reçu du poste central en fonction du type de pièce et de la matière pour le choix des conditions de coupe.
- Transmissions d'1 information de pièce évacuer vers le poste central..

Ecire en ladder pour chacun des postes le programme concernant le réseau.



**Rappel:** Chaque automate dispose de 4 mots 16 bits COM i,0 à COM i,3. (i étant le numéro de l'automate sur le réseau).

**Note:** Attention X16 n'appartient au grafcet N2.

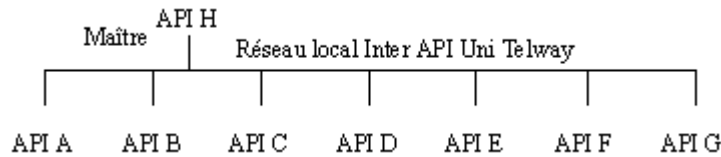
L'autorisation de transfert de pièce doit être inhibé une fois la pièce transférée.

<b>Les affectations pour le poste 0</b>	<b>Les affectations pour le poste 1</b>
W1: Type de pièce	W1: Type de pièce.
W2: Matière	W2: Matière
O1,0: Descente Butée (monostable)	O1,0: Descente butée (mono)
I1,0: Butée Bas	I1,0: Butée Bas
Com0,x : A définir.	Com1,x : A définir.

## Performance du réseau automate : Unitelway sur TSX

On se propose de calculer le temps maximum de transmission des informations sur le réseau de communication de type maître esclave à huit stations.

Le bus proposé supporte le service Uni-TE qui est le langage commun entre les constituants. Il permet l'accès aux variables et paramètres des divers équipements. La taille des messages est limitée à 32 Octets.



Sachant que les temps de cycle moyens de traitement (asynchrone) des divers automates sont respectivement:

- pour l'automate (ou équipement) maître H:  $TC_8=80ms$
- pour les automates (équipements) esclaves A et B:  $TC_1=50ms$  et  $TC_2=35ms$

Calculer les valeurs mini et maxi du temps de réponse T ( Temps de réaction à un évènement "e") pour les événements suivants:

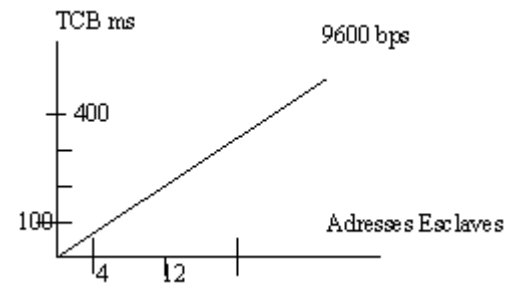
- Un événement apparu sur la PO de l'automate A et inuisant une action "s" sur la PO associé à l'API B (T=durée entre l'apparition de "e" et l'émission de "s"). On appuiera son estimation à partir d'un schéma.
- Un événement, qui, apparu sur la PO de l'automate A, nécessite une coordination et un traitement sur l'API H, avant d'autoriser une action "s" sur la PO de A. Là aussi on appuiera le calcul d'un schéma.

### FICHE Constructeur

Le tps de cycle du bus UNI TELWAY est lié au nbr d'équipement scrutés (adresses liaison), au débit binaire, au temps de retournement de chaque équipement, au nombre, à la longueur et à la nature des messages.

TCB: Tps de cycle bus, correspond au temps séparant deux interrogations d'un même équipement.

La droite donne le tps de cycle en fonction du nbr d'esclaves à 9600 bps.

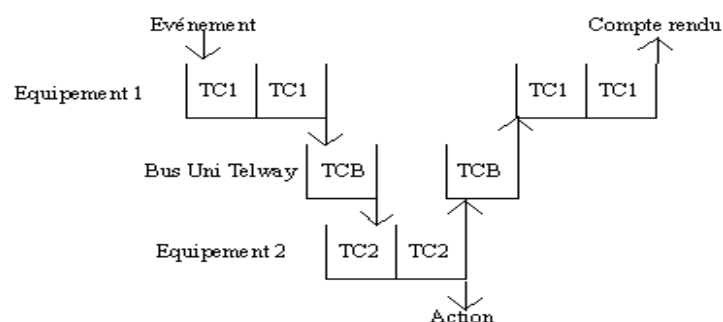


Le tableau suivant indique les temps à ajouter (en ms) pour connaître la véritable valeur du TCB en fonction du trafic (N=nbr de caractères utiles):

Maître vers esclave	$24 + 1,2 N$
Esclave vers maître	$19 + 1,2N$
Esclave vers esclave	$44 + 2,3N$

Le tps de réponse d'application dans une architecture d'automatisme distribuée ne dépend pas que du système de communication, il dépend aussi:

du tps de traitement des équipements émetteur et destinataire du message



de l'asynchronisme entre les cycles bus et unités de traitement.

TCB: Tps de cycle bus

TC1: Tps de cycle équipement 1

TC2: Tps de cycle équipement 2