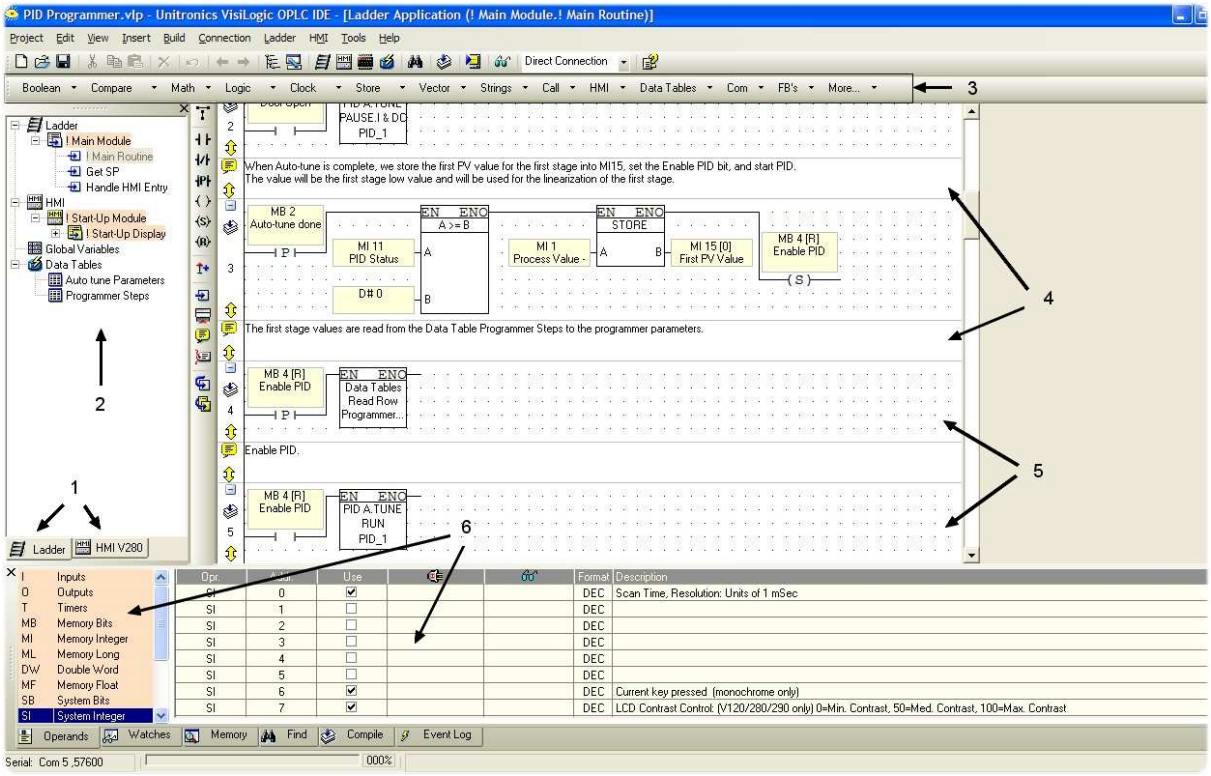
Plan du document

Plan du document

Plan du document

L'interface Visilogic

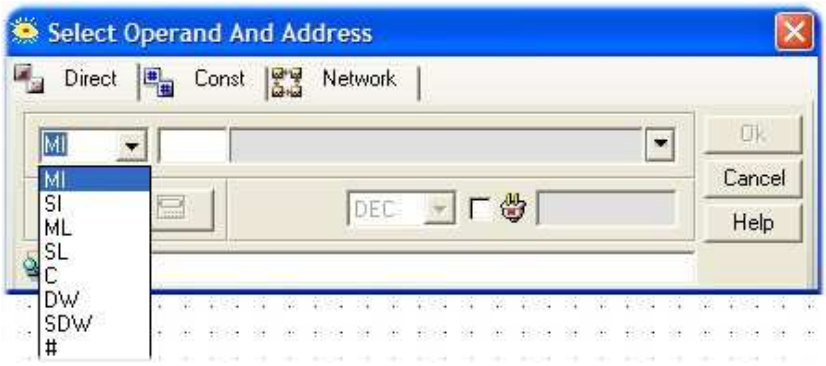


- 1 => Choix de l'éditeur (Ladder ou IHM)
- 2 => Navigation entre les différents sous programmes et différentes pages
- 3 => Choix des éléments Ladder (conditions, actions…)
- 4 => Commentaires
- 5 => Segments contenants les éléments Ladder
- 6 => Liste des différentes variables

Principe et variables

Un programme Ladder se lit par ligne, de gauche à droite. Ces lignes contiennent des éléments correspondants à des conditions et d'autres correspondants à des actions à effectuer. Le principe est le suivant : lorsqu'un élément est validé, l'automate lit celui à sa droite ; puis passe à la ligne d'en dessous lorsque qu'il ne trouve plus d'éléments. Presque toutes les fonctions Ladder utilise des variables que vous devez définir correctement pour parvenir à vos fins. Voici ce que vous devez savoir pour paramétrer correctement vos variables :

- Le type de donnée



*	MB	=>	Bit interne (0 ou 1 ; gérés dans le Ladder)
*	SB	=>	Bit système (gérés par l'automate)
*	MI	=>	Mot interne de 16 bits (-32 768 à + 32 767)
*	SI	=>	Mot système de 16 bits (- 32 768 à + 32 767)
*	ML	=>	Mot Long interne ; résolution 32 bits (- 2 147 483 648 à + 2 147 483 647)
*	SL	=>	Mot Long système ; résolution 32 bits (- 2 147 483 648 à + 2 147 483 647)
*	DW	=>	Double Mot interne ; résolution 32 bits non signé (0 à 4 294 967 295)
*	SDW	=>	Double Mot système ; résolution 32 bits non signé (0 à 4 294 967 295)
*	MF	=>	Entier interne ; résolution 32 bits norme IEEE
*	I	=>	Entrée
*	0	=>	Sortie
*	T	=>	Temporisation ; résolution maximale 10ms
*	#	=>	Constante
*	C	=>	Compteur

Les variables systèmes contiennent des informations qui évoluent en fonction de certains événements (mise sous tension, date, numéro de la page…). Elles sont gérées par l'automate et servent généralement pour poser des conditions.

Principe et variables

- L'adresse de la donnée

* MB => 0 à 4095
* MI => 0 à 2047
* ML => 0 à 255
* DW => 0 à 63
* MF => 0 à 23
* C => 0 à 23
* T => 0 à 191
* I & 0 => dépend de l'automate

- La description de la variable

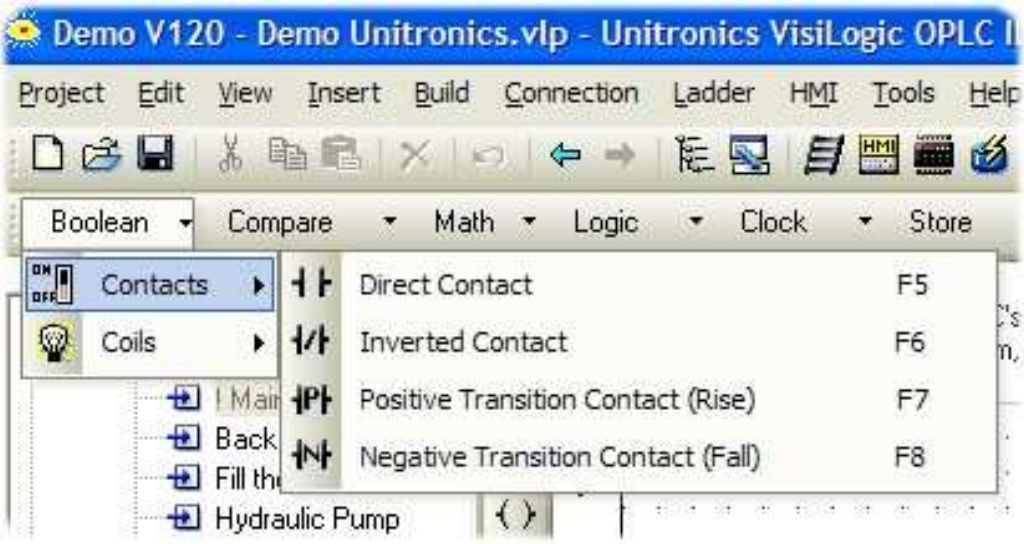
Principe et variables

- Le format de la valeur

- La valeur à la mise sous tension

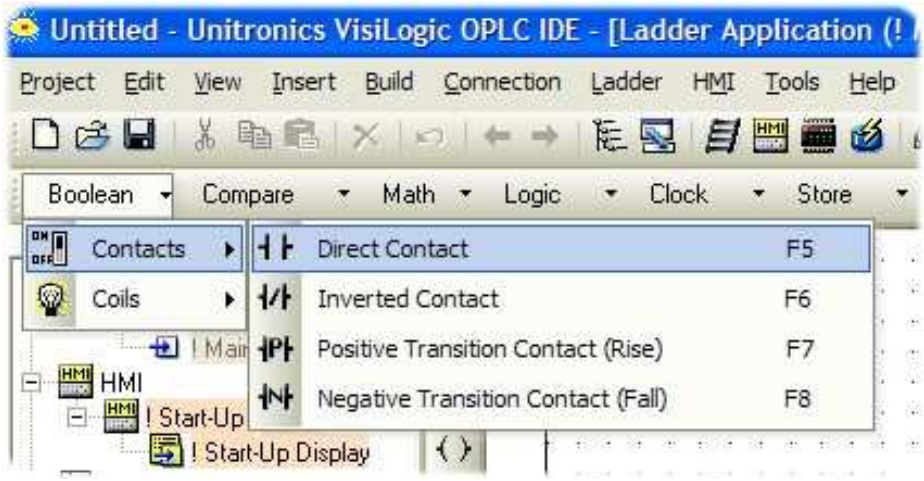
Les contacts

- Ils servent de conditions (sur l'état d'un bit, d'une entrée…)
- Il existe 4 types de contacts :



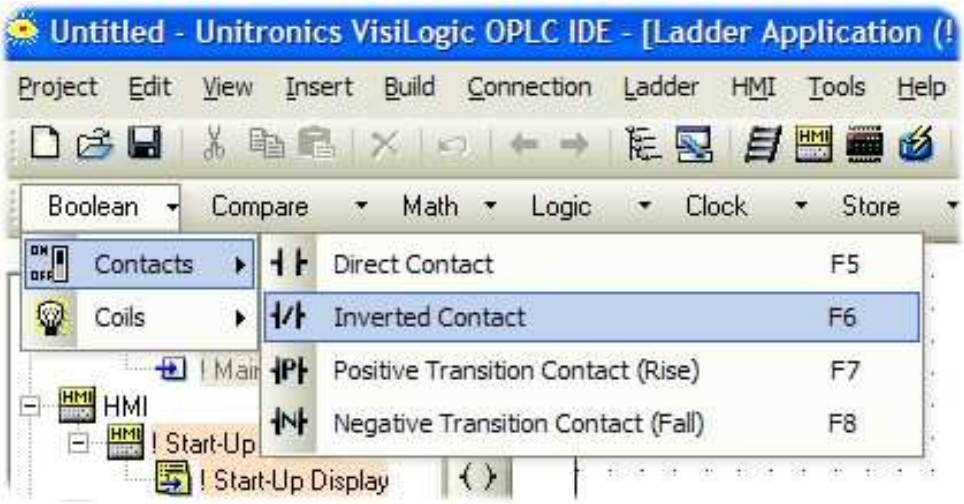
Les contacts

Les contacts directs :



- La condition est validée si l'élément qui lui est associé est à 1

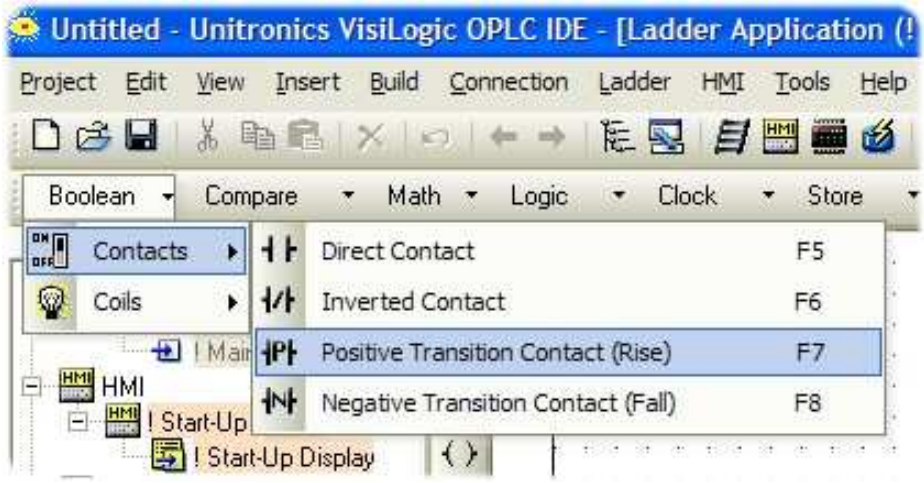
Les contacts indirects :



- La condition est validée si l'élément qui lui est associé est à 0

Les contacts

Les fronts montants :



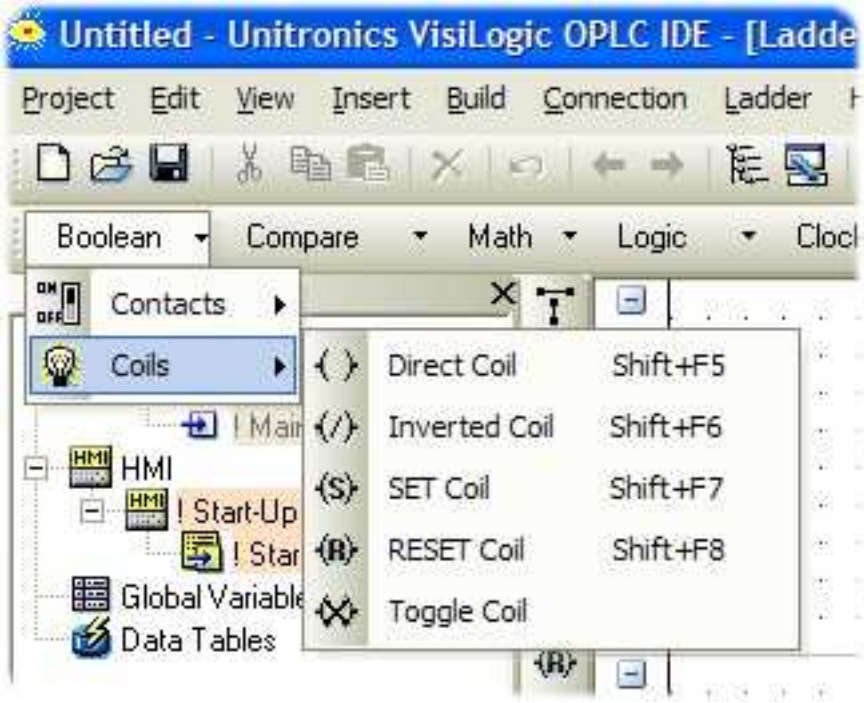
- La condition est activée seulement au moment du passage de 0 à 1 de l'élément qui lui est associé (On ne tient pas compte du temps pendant lequel l'élément reste à 1 par la suite)

Les fronts descendants :

- La condition est activée seulement au moment du passage de 1 à 0

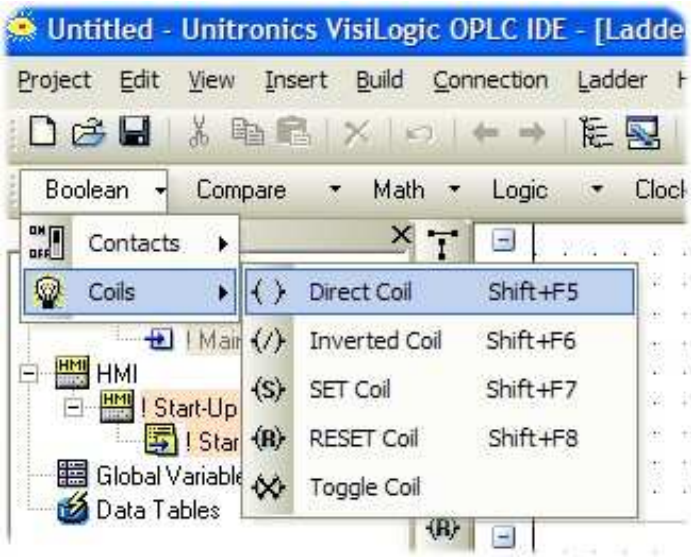
Les bobines

- Elles représentent l'activation ou la désactivation d'un élément
- Il existe 5 types de bobines



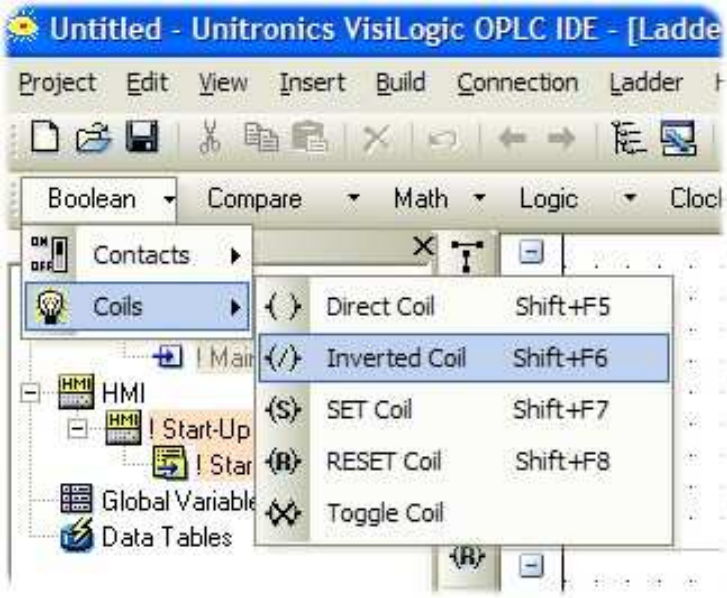
Les bobines

Les bobines directes :



- L'élément associé est activé pendant **un seul cycle** automate

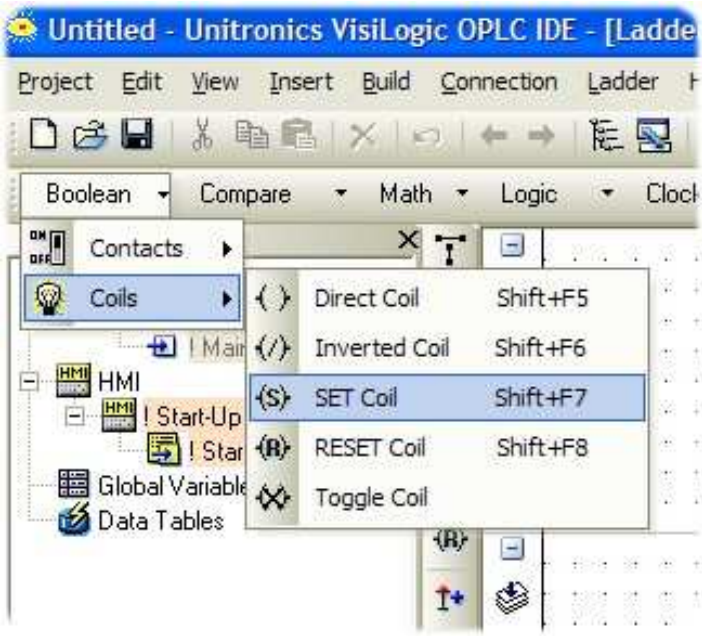
Les bobines indirectes :



- L'élément associé est désactivé pendant **un seul cycle** automate

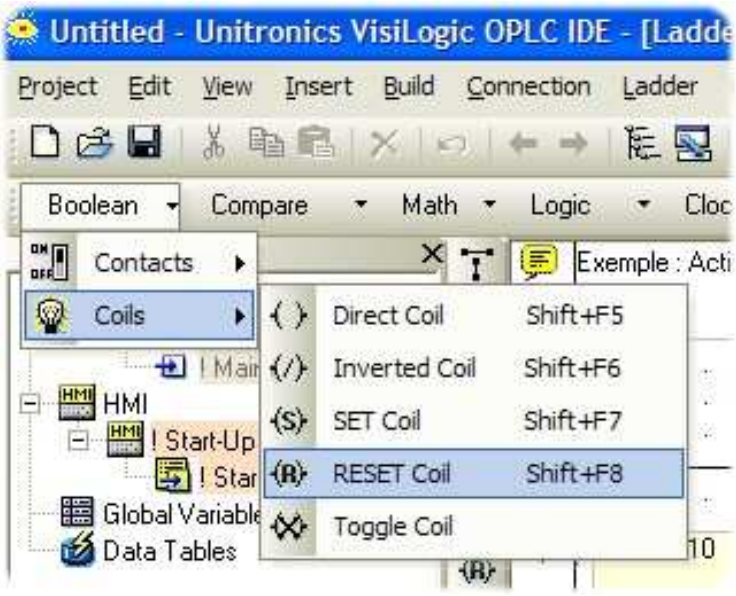
Les bobines

Les mises à 1 (Set) :



- L'élément associé est activé jusqu'à ce qu'on le remette à 0

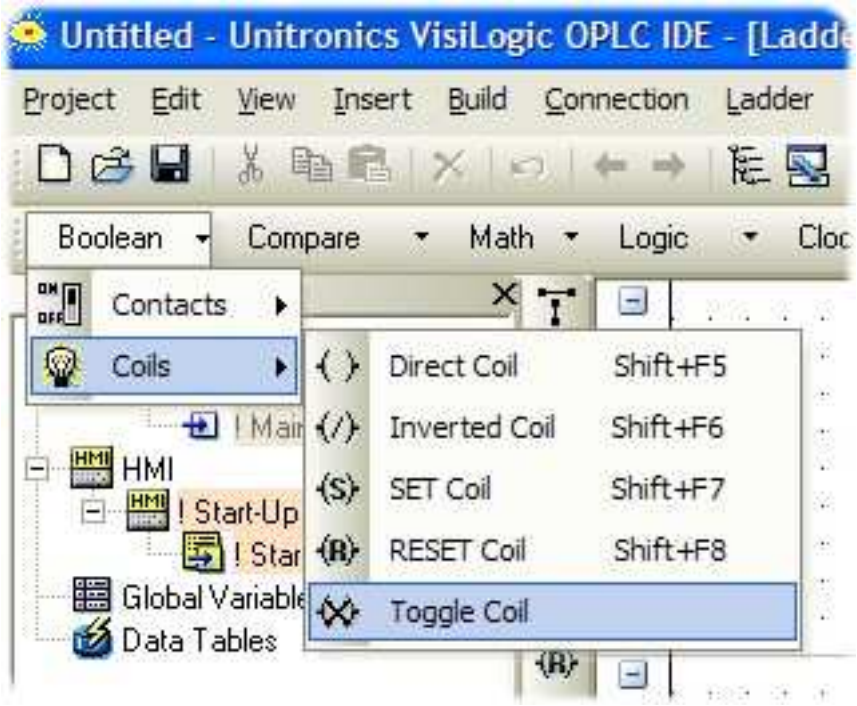
Les mises à 0 (Reset) :



- L'élément associé est désactivé jusqu'à ce qu'on le remette à 1

Les bobines

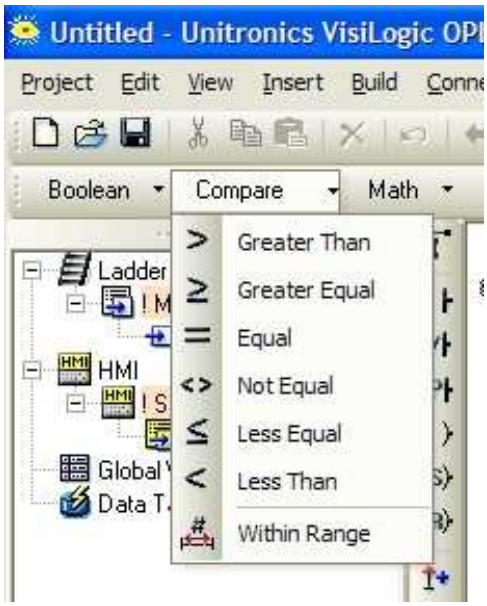
Les changements d'états (Toggle) :

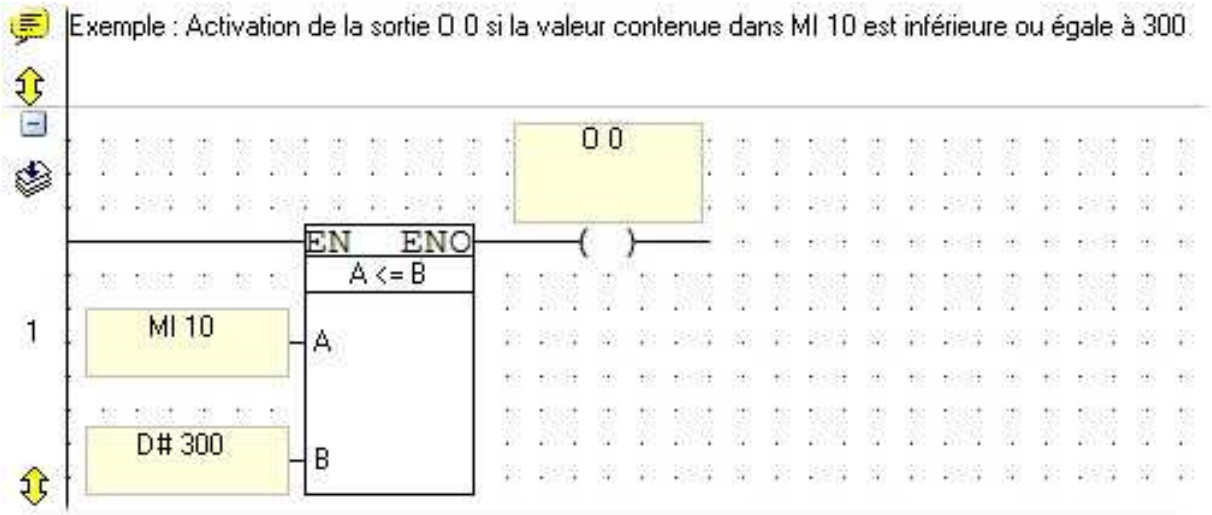


- L'état de l'élément associé est inversé (passe à 1 si il était à 0 et inversement)

Les fonctions de comparaison

- Elles servent de conditions
- Elles permettent de comparer deux valeurs

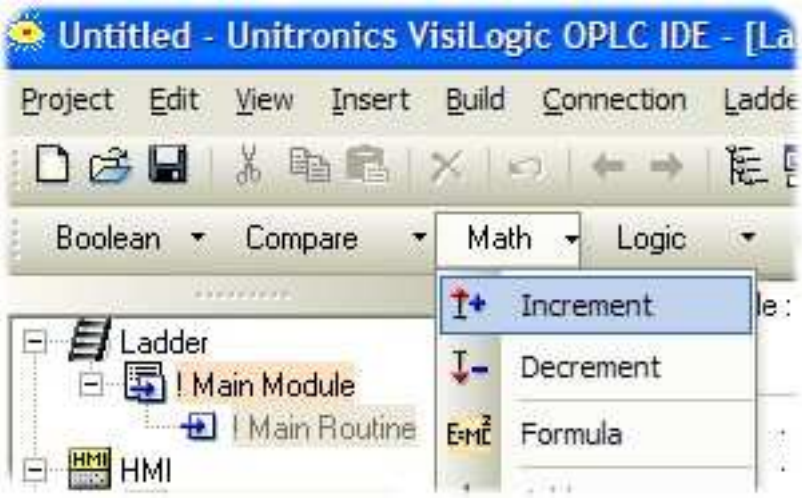


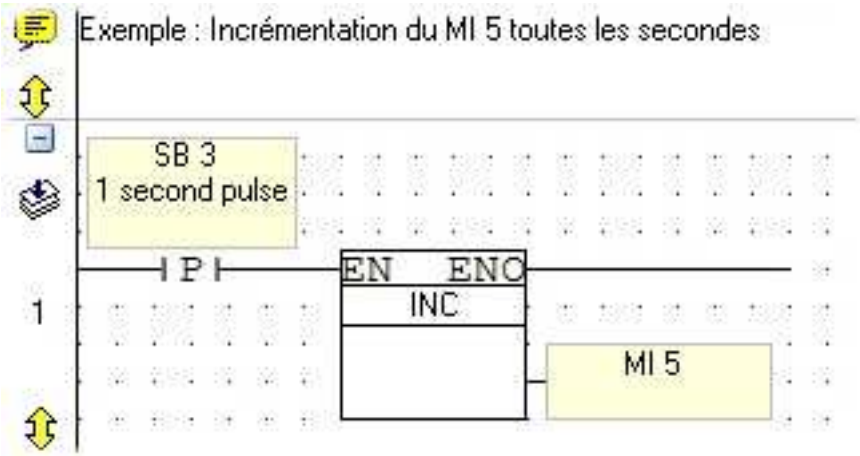


Les fonctions mathématiques

Incrémentation / Décrémentation :

- Ces fonctions permettent d'ajouter ou de soustraire 1 à la valeur d'un mot

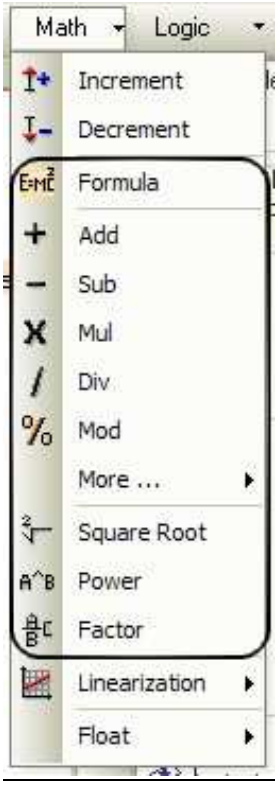




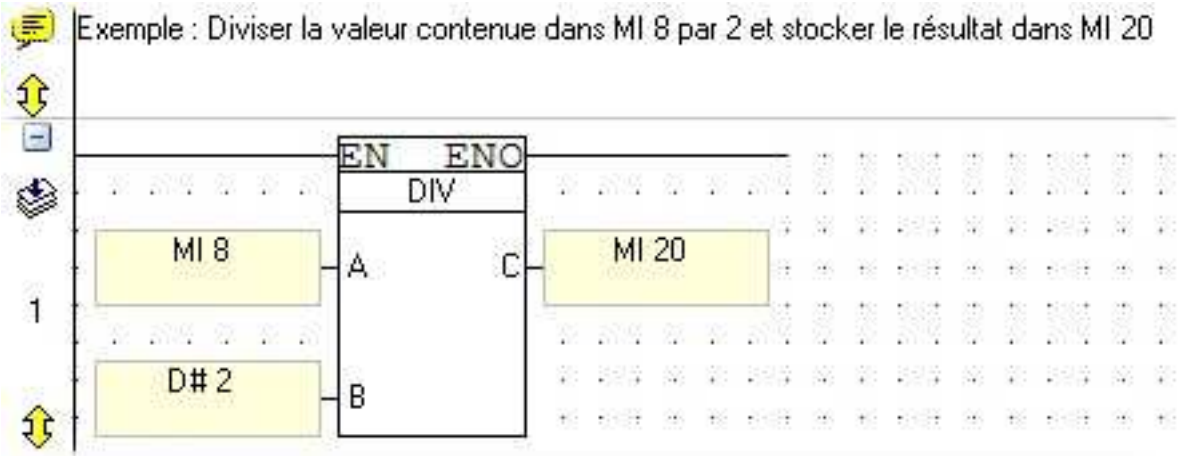
- Attention ! Ces fonctions s'utilisent avec des fronts (montants ou descendants) ; sinon le mot est incrémenté à chaque cycle automate.

Les fonctions mathématiques

Les outils de calculs :

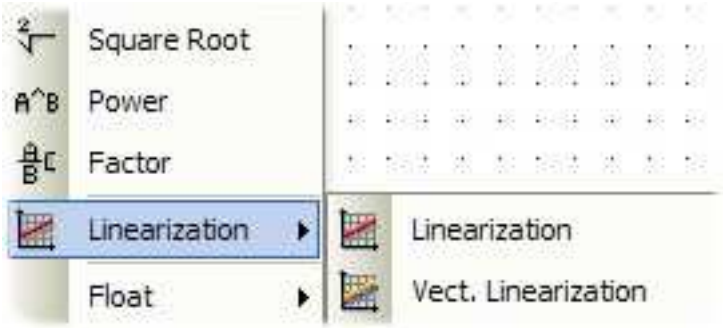


- Vous retrouvez ici tous les outils mathématiques de base (addition, multiplication…)
- « Modulo » permet de récupérer le reste d'une division
- « Formula » permet d'écrire de longs calculs dans un seul bloc et d'utiliser des outils mathématiques plus complexes (exponentielle, trigonométrie…)

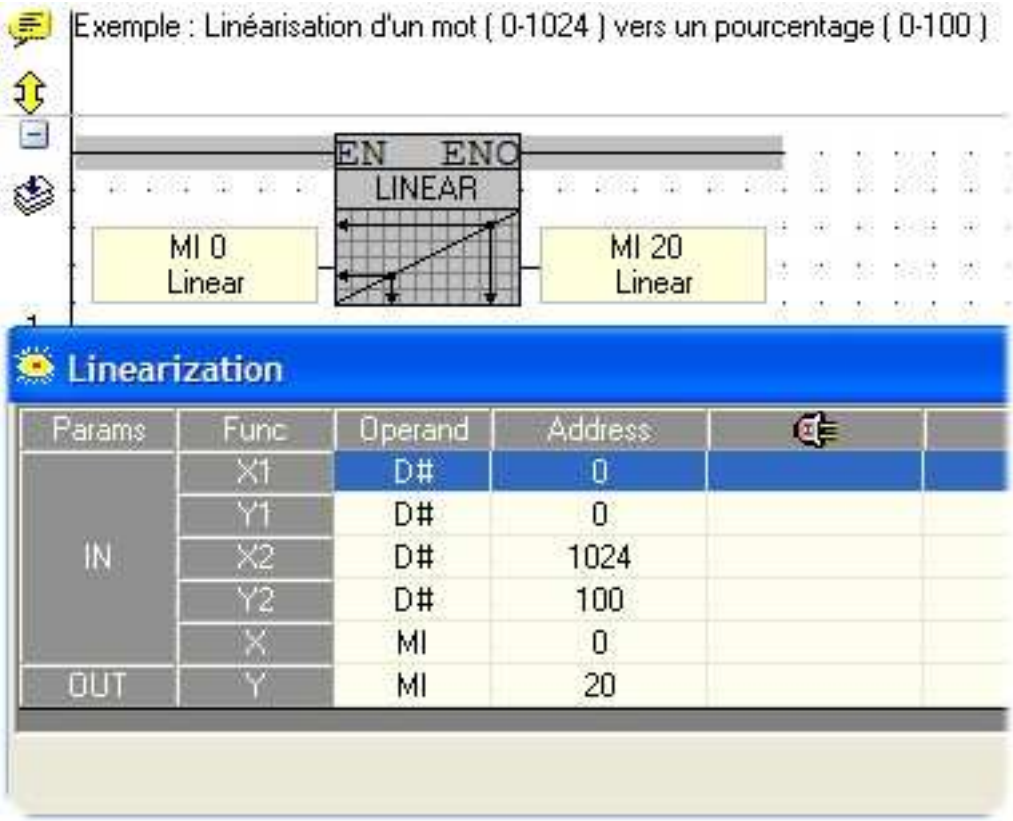


Les fonctions mathématiques

La linéarisation :



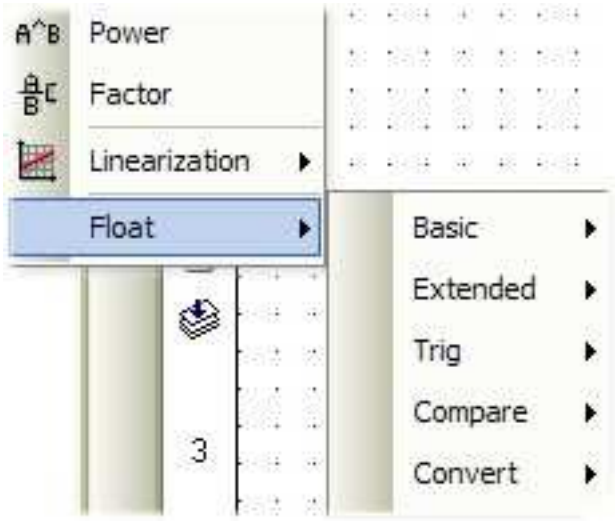
Elle est généralement utilisée pour passer d'un format à un autre.



L'outil « Vect. Linéarization » Permet de linéariser toute une suite de mot en même temps.

Les fonctions mathématiques

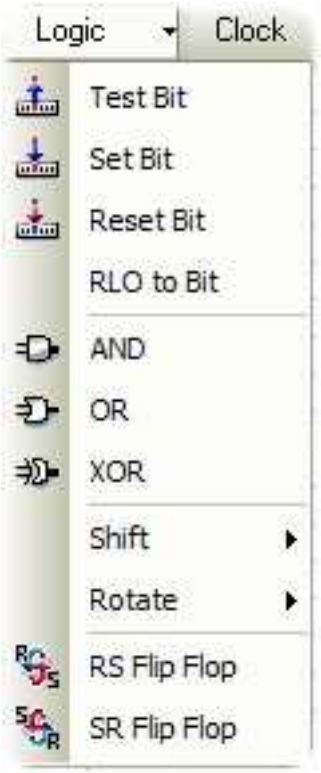
Les nombres réels (Float) :



Vous retrouvez ici tous les outils permettant de :

- Effectuer des calculs de base
- Effectuer des calculs plus évolués
- Utiliser des fonctions trigonométriques
- Faire des comparaisons avec des réels
- Convertir un réel en 2 entiers (partie entière et partie décimale) ou inversement

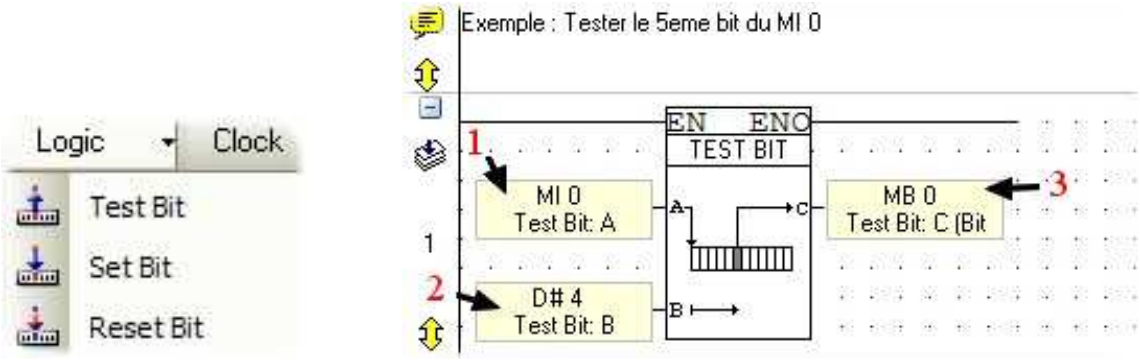
Les fonctions logiques



Les fonctions logiques

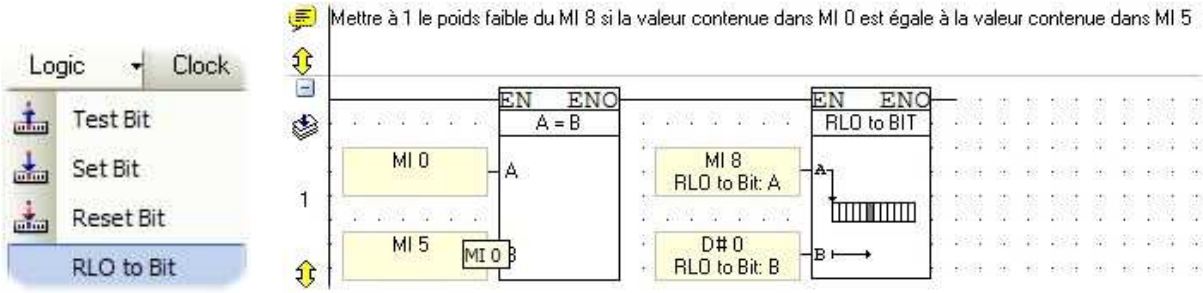
Les actions sur un bit :

Vous avez la possibilité, lorsque vous travaillez avec des mots, d'agir sur un seul des bits qui composent ce mot. Vous pouvez connaître son état (0 ou 1) ou le forcer (mise à 0 ou 1).



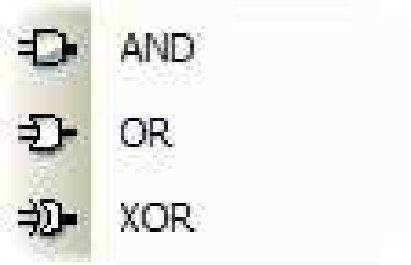
- 1 => Indiquez le mot dans lequel se trouve le bit sur lequel vous souhaitez agir
- 2 => Indiquez le décalage par rapport au poids faible
- 3 => Choisissez le bit dans lequel sera mis le résultat (demandé seulement lorsque vous testez un bit)

La fonction « RLO to Bit » vous permet de mettre à 1 un bit seulement si le bloc est activé



Les fonctions logiques

Vous avez la possibilité d'utiliser les fonctions logiques ET, OU et OU EXCLUSIF. Ces fonctions modifient l'état des bits du mot de sortie en fonction de l'état des mots d'entrée et de la table de vérité de la fonction choisie :

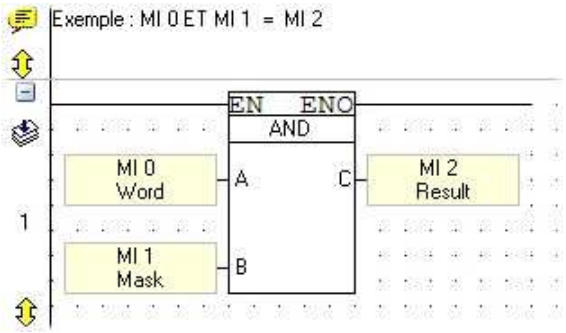


Voici les tables de vérité de ces fonctions :

Fonction ET		
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Fonction OU		
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

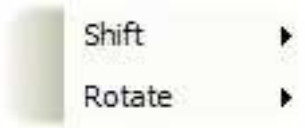
OU EXCLUSIF		
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0



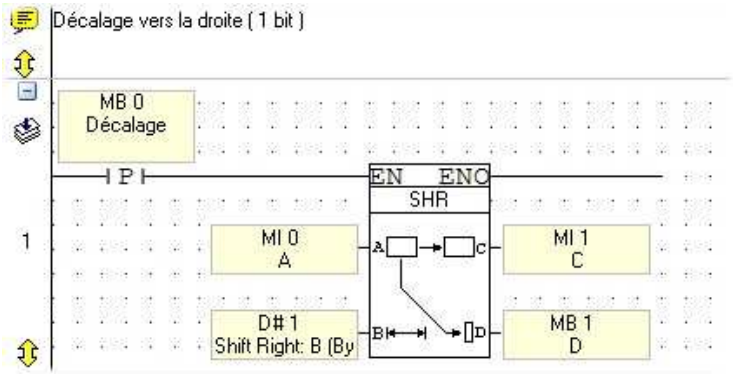
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word	1	0	0	0	1	1	0	1	0	1	0	1	0	1	1	1
AND																
Mask	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Result	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Les fonctions logiques

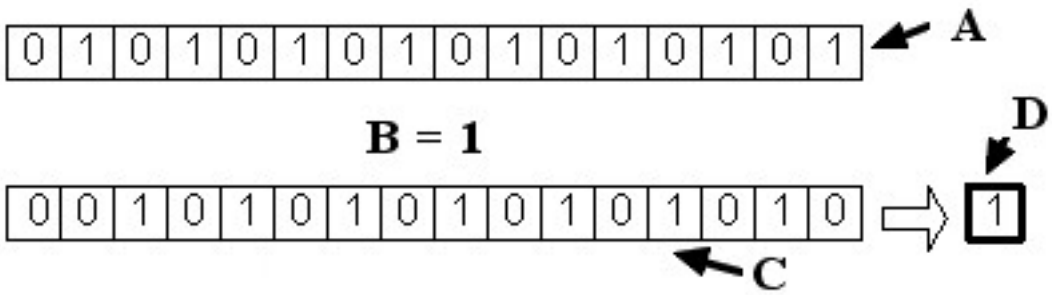
Les fonctions de décalage :



Ces fonctions vous permettent de décaler, vers la gauche ou la droite, les bits qui se trouvent à l'intérieur d'un mot.



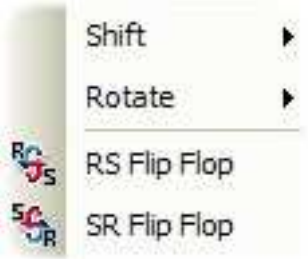
- A => Indiquez le mot dans lequel vous souhaitez travailler
- B => Choisissez le nombre de bits à décaler
- C => Indiquez le mot qui contiendra le résultat
- D => Choisissez le bit qui sauvegardera la valeur du dernier bit décalé



Les fonctions « Rotate » agissent de la même façon. La différence est que le poids faible n'est plus perdu mais transféré dans le poids fort.

Les fonctions logiques

Les bascules :



Vous retrouver ici les bascules RS et SR classiques dont voici les tables de vérité :

Bascule RS

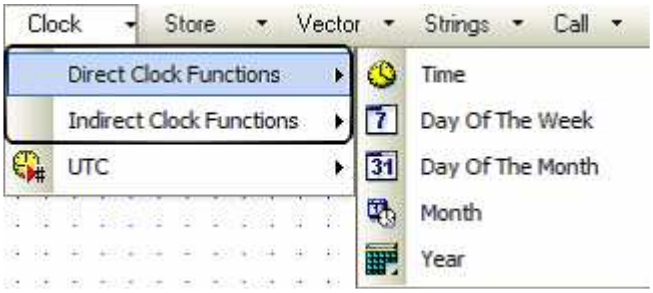
R (A)	S (B)	Q
0	0	No change
0	1	1
1	0	0
1	1	0

Bascule SR

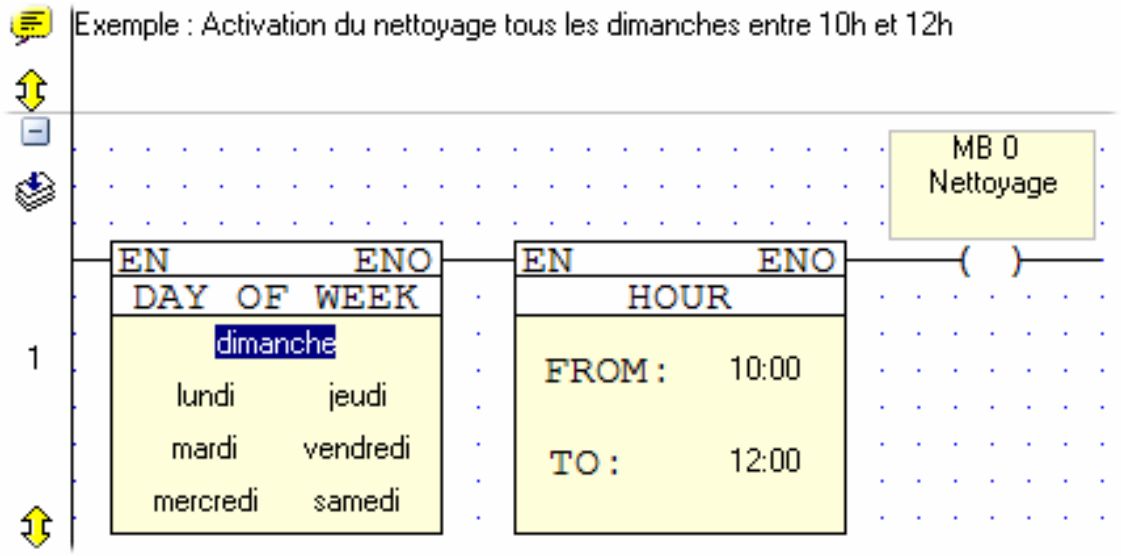
S (A)	R (B)	Q
0	0	No change
0	1	0
1	0	1
1	1	1

Les fonctions d'horloge

Les conditions :

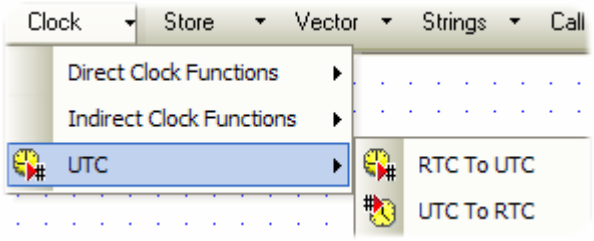


Ces fonctions permettent d'ajouter des conditions sur l'année, le mois, le jour ou l'heure. Les fonctions directes utilisent l'horloge interne de l'automate. Les fonctions indirectes, quant à elles, utilisent des valeurs stockées dans des mots.

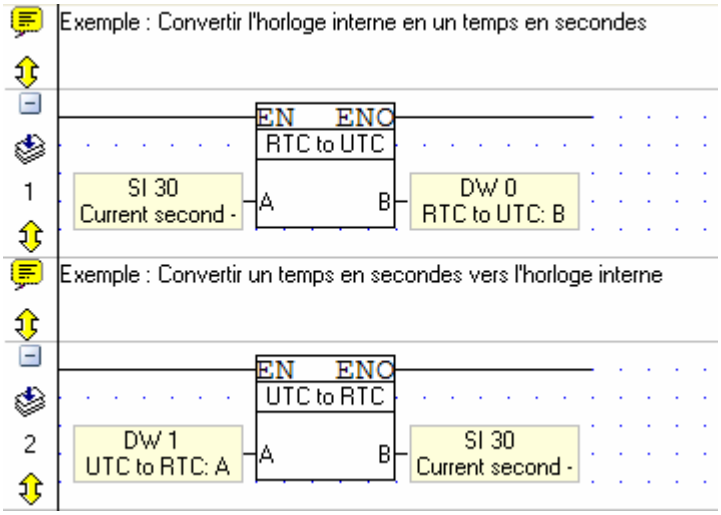


Les fonctions d'horloges

Conversion horloge ⇔ secondes :



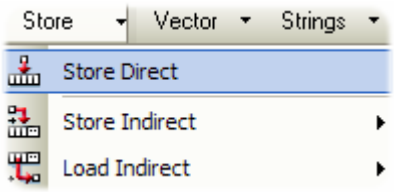
Ces fonctions permettent de convertir l'horloge interne en un temps en secondes et vice versa. Aller lire ou écrire le SI 30 (secondes temps réel) prendre automatiquement en compte les SI 31, 32, 33 et 34 (heure, date, année et jour de la semaine).

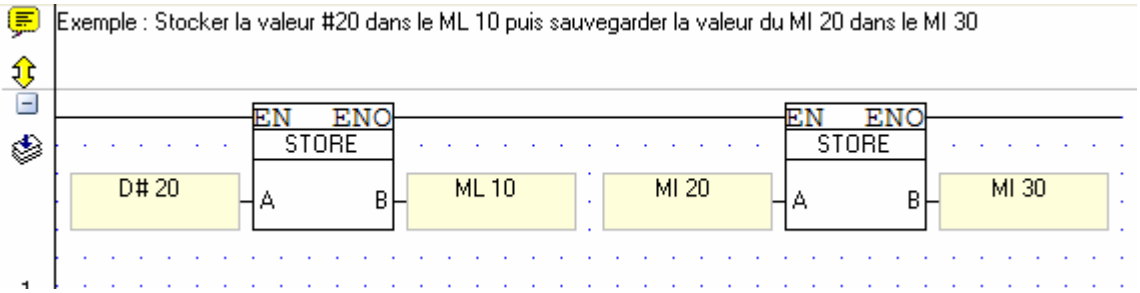


Les fonctions de stockage

Stockage direct :

- Le stockage direct permet de sauvegarder une valeur dans un mot

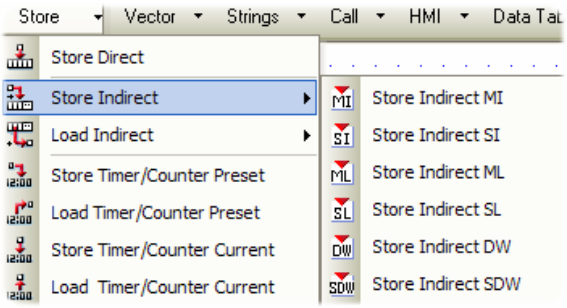




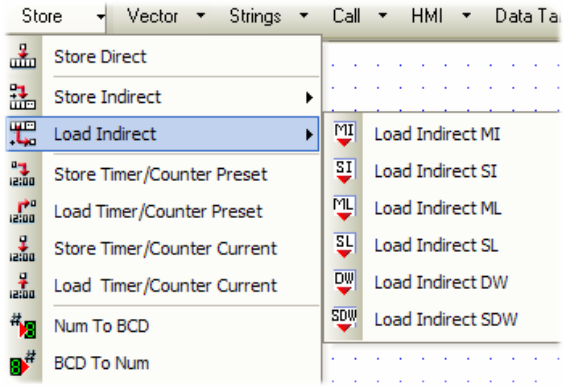
Les fonctions de stockage

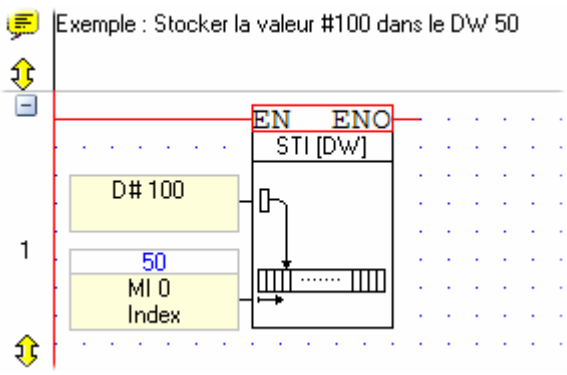
Stockage / chargement indirect

- Le stockage indirect permet de sauvegarder une valeur dans le mot se trouvant à l'adresse contenue dans l'opérande B



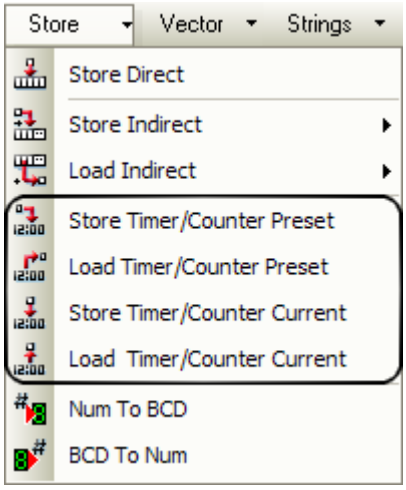
- Le chargement indirect permet de récupérer la valeur du mot se trouvant à l'adresse contenue dans l'opérande A





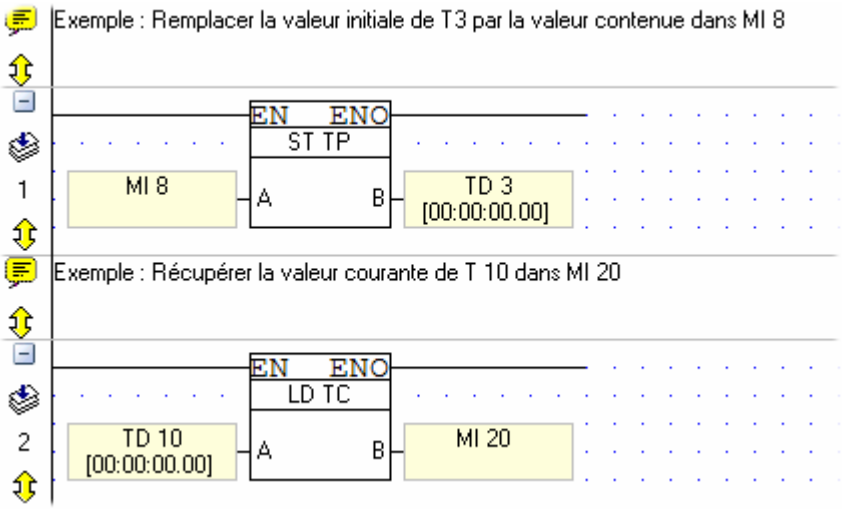
Les fonctions de stockage

Modification / récupération de la valeur d'une tempo :



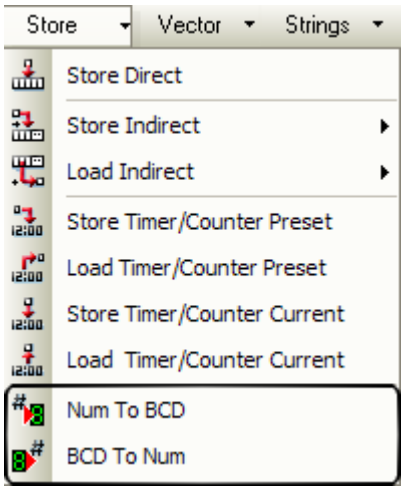
- Les fonctions « Store » vous permettent de modifier la valeur initiale ou courante d'une temporisation en la remplaçant par la valeur contenue dans l'opérande A

- Les fonctions « Load » vous permettent de récupérer la valeur initiale ou courante d'une temporisation en la sauvegardant dans l'opérande B



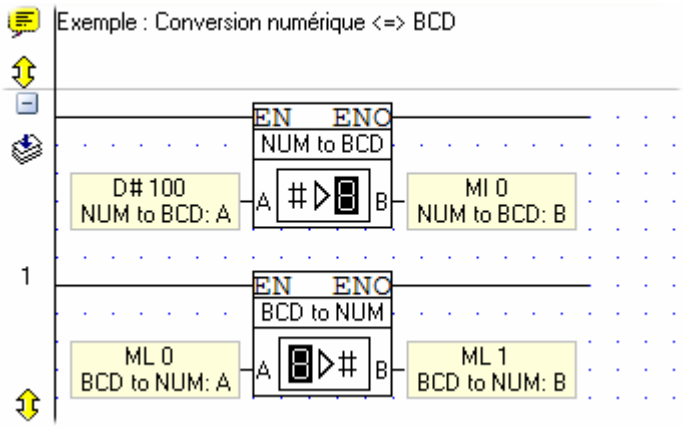
Les fonctions de stockage

Numérique ⇔ BCD :



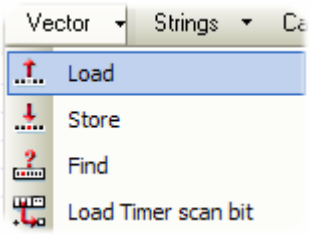
Ces fonctions vous permettent de convertir des valeurs du format numérique vers le format BCD et vice-versa.

Le format BCD (Binary Coded Decimal => Décimal Codé Binaire) est utilisé, par exemple, pour piloter des afficheurs 7 segments.

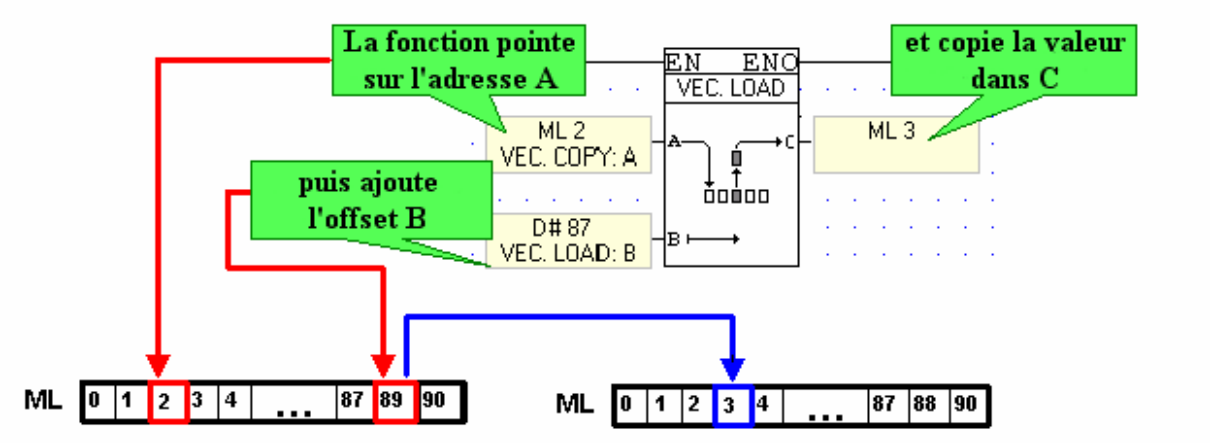


Les fonctions « Vector »

La récupération de valeurs :

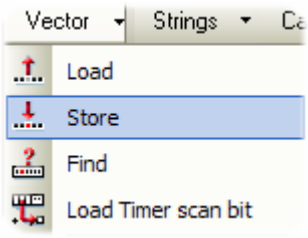


Cette fonction permet de récupérer la valeur contenue dans un mot via un offset

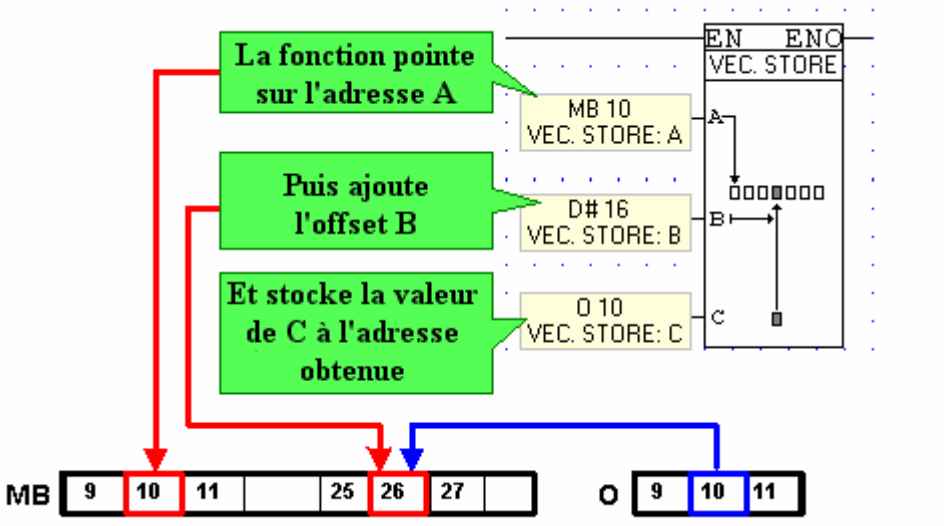


Les fonctions « Vector »

Le stockage de valeurs :

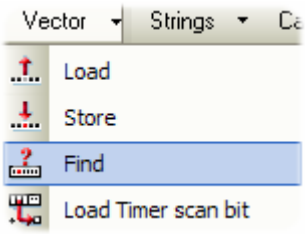


Cette fonction permet de stocker une valeur dans une adresse définie par un offset.

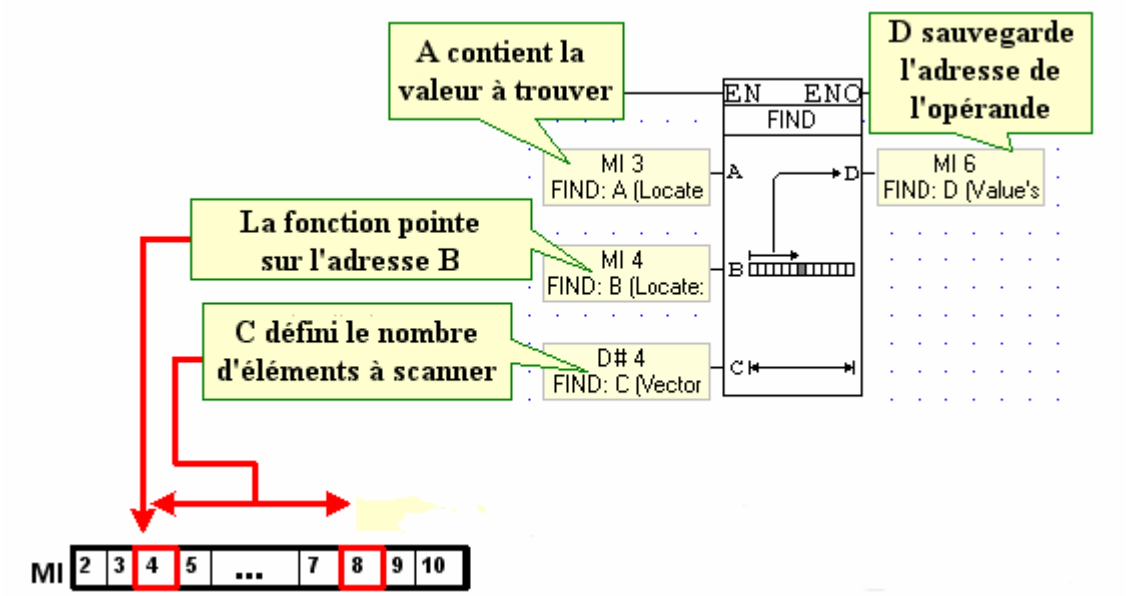


Les fonctions « Vector »

Trouver une valeur :



Cette fonction permet de scanner une plage mémoire afin de trouver le mot qui contient la valeur que l'on cherche ou le bit qui est dans l'état désiré.

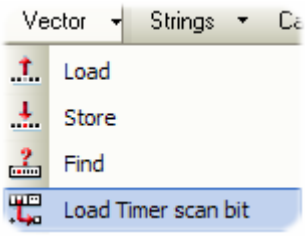


Une fois la valeur trouvée, son adresse est sauvegardée dans l'opérande D. Si aucun élément ne correspond à la valeur désirée, D contient la valeur « -1 ».

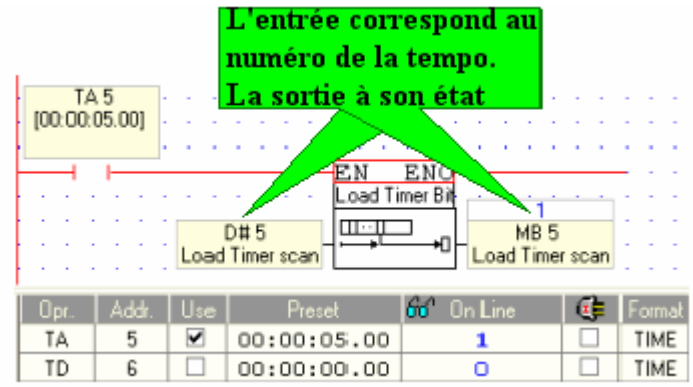
Si il existe plusieurs éléments correspondants à la recherche, seul le premier sera pris en compte.

Les fonctions « Vector »

Récupérer l'état d'une temporisation :

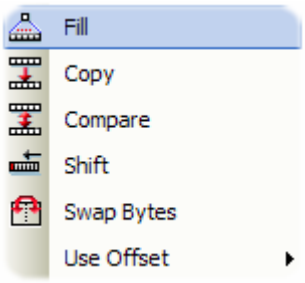


Cette fonction permet de connaître l'état d'une temporisation (0 si la temporisation est inactive, 1 dans le cas contraire).

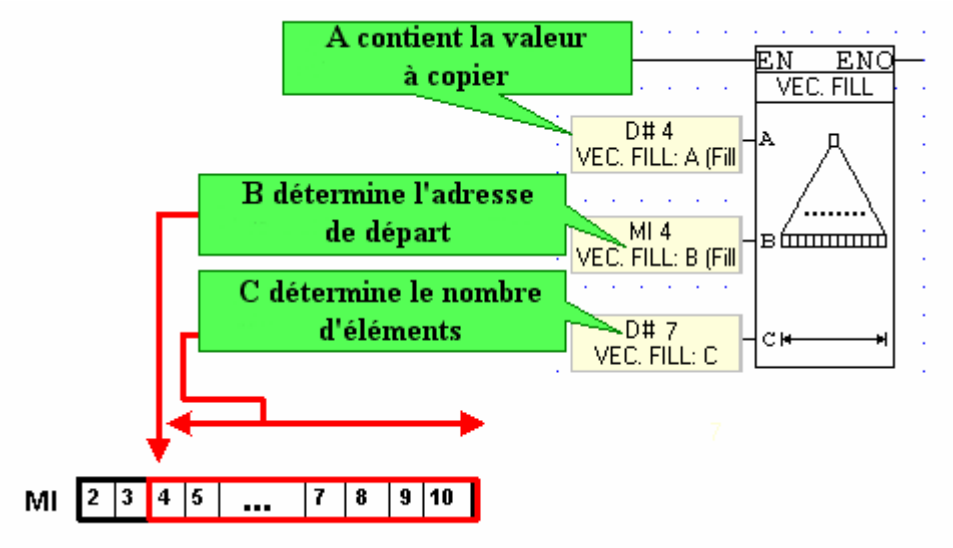


Les fonctions « Vector »

Copier une valeur dans une suite de mots :

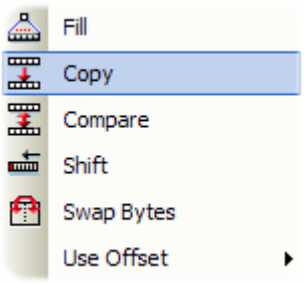


Cette fonction permet de copier la même valeur dans une suite de mots consécutifs.

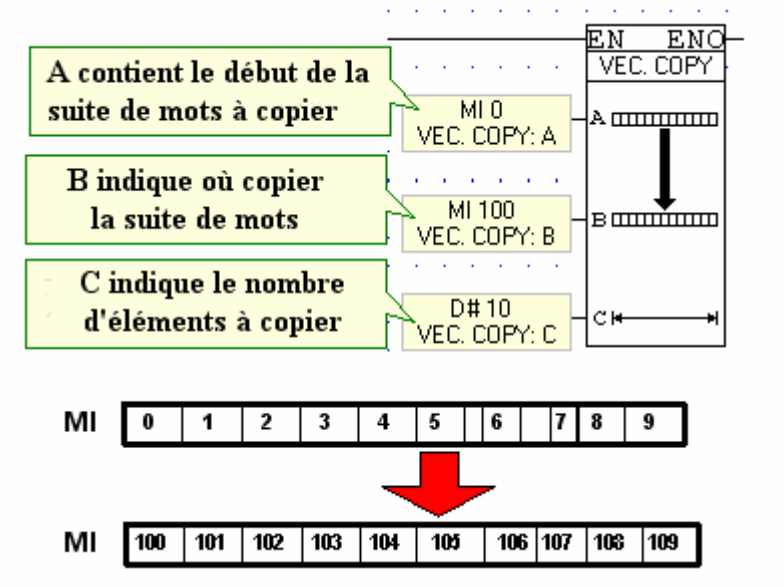


Les fonctions « Vector »

Copier une suite de mots :

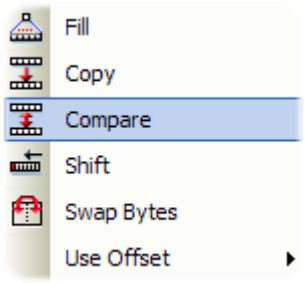


Cette fonction permet de copier une suite de mots dans une autre suite de mots de même longueur.

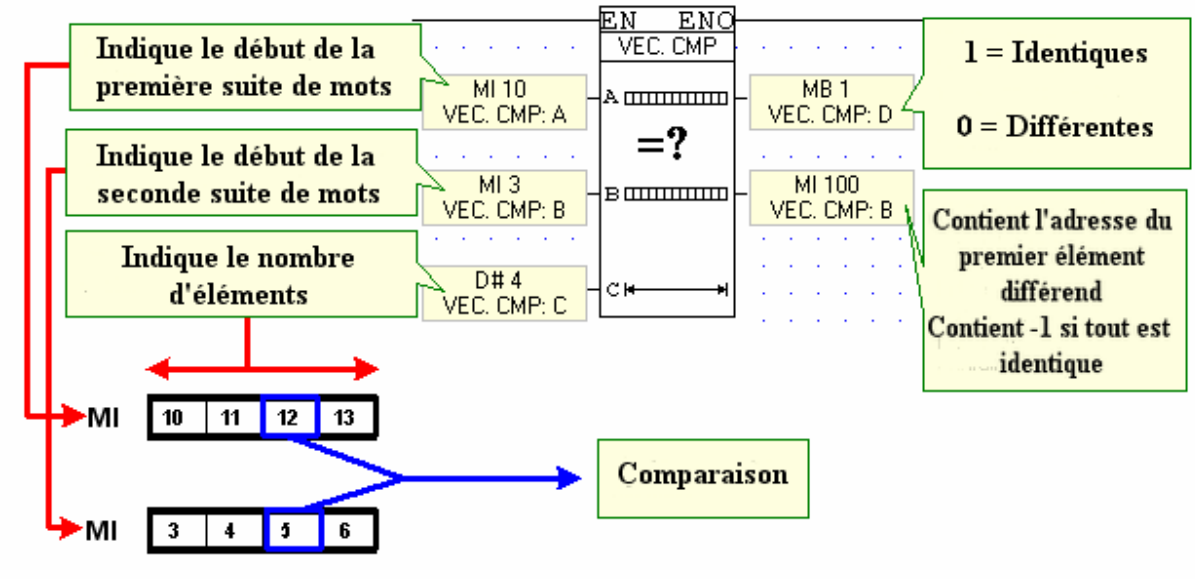


Les fonctions « Vector »

Comparer deux suites de mots :

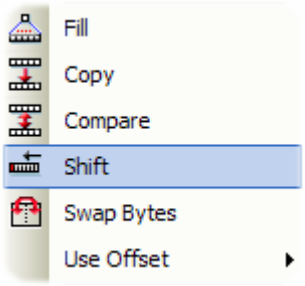


Cette fonction compare deux suites de mots et indique d'adresse à partir de laquelle elles ne sont plus identiques.



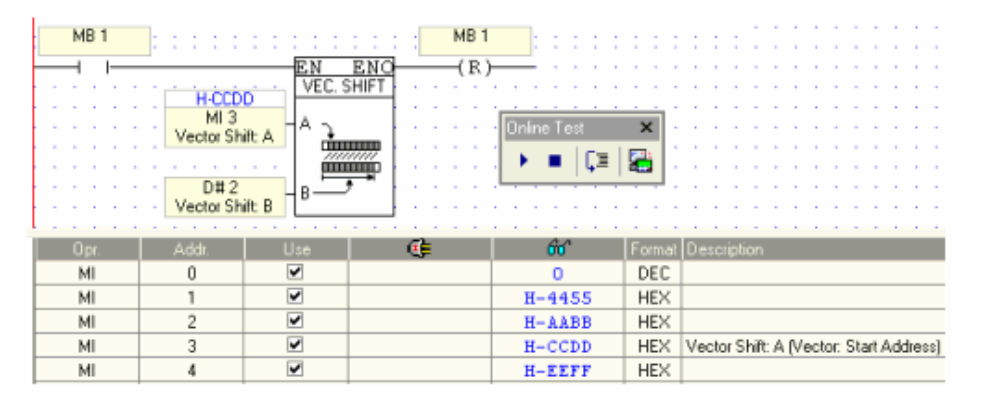
Les fonctions « Vector »

Décalage d'octets vers la droite :

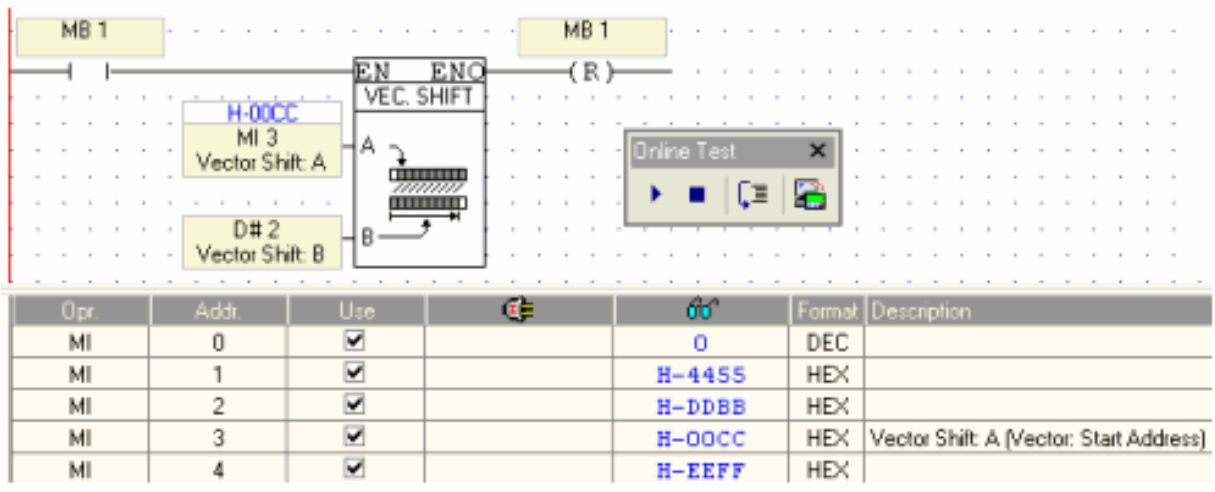


Cette fonction permet de décaler des octets vers la droite.

Exemple : MI 3 contient H-CCDD

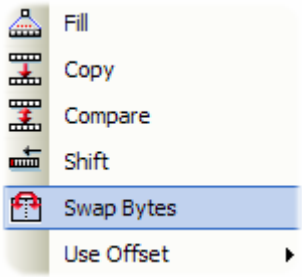


Après le décalage, le poids fort devient le poids faible. Le poids faible disparaît.



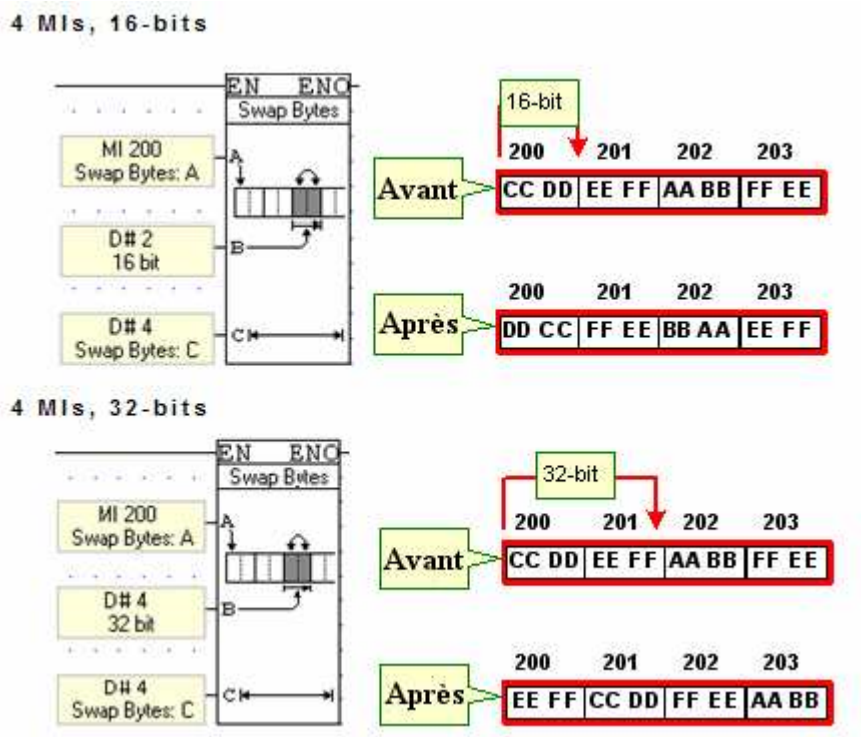
Les fonctions « Vector »

Inversion poids fort – poids faible :



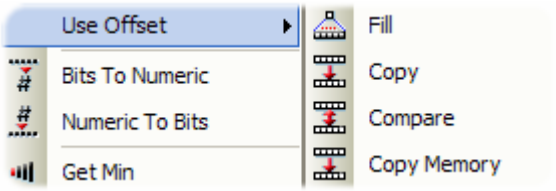
Cette fonction permet d'inverser l'octet de poids faible d'un mot avec celui de poids fort.

- A correspond à l'adresse à partir de laquelle vous désirez travailler
- B permet de choisir entre swapper 16 ou 32 bits
- C indique le nombre de mots sur lesquelles la fonction agira



Les fonctions « Vector »

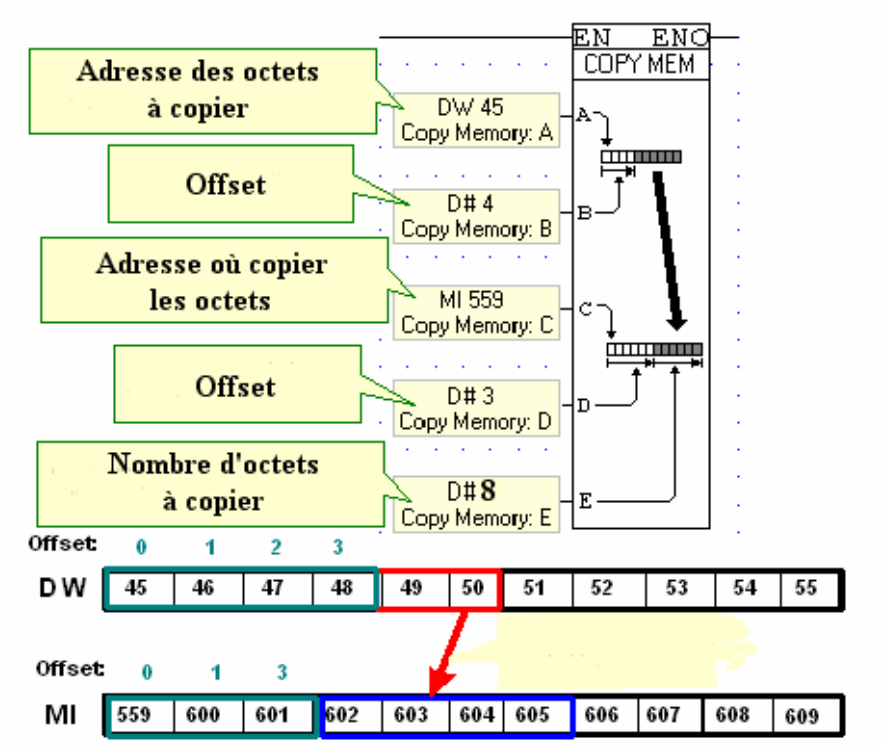
Les fonctions avec offset :



Vous retrouvez dans cette catégorie certaines des fonctions abordées précédemment et avec lesquelles vous avez la possibilité de travailler en utilisant un offset pour définir les adresses.

Conversion :

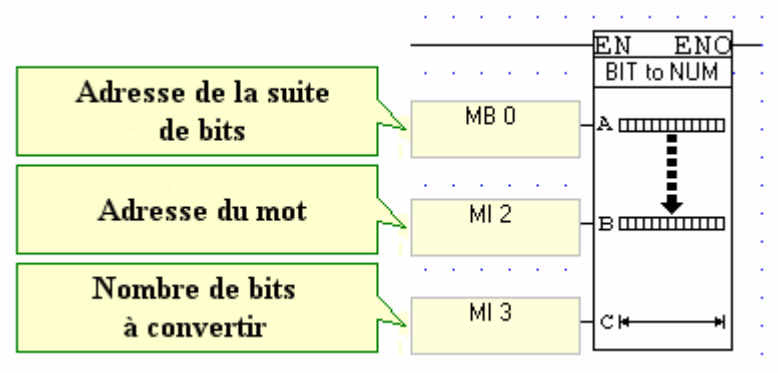
La fonction « Copy Memory » permet de copier des octets dans d'autres octets, sans tenir compte du type de mot dans lesquels ils se trouvent. Cela peut être utile pour convertir un DW en 2 MIs par exemple.



Les fonctions « Vector »

Conversion valeur numérique ⇔ suite de bits :

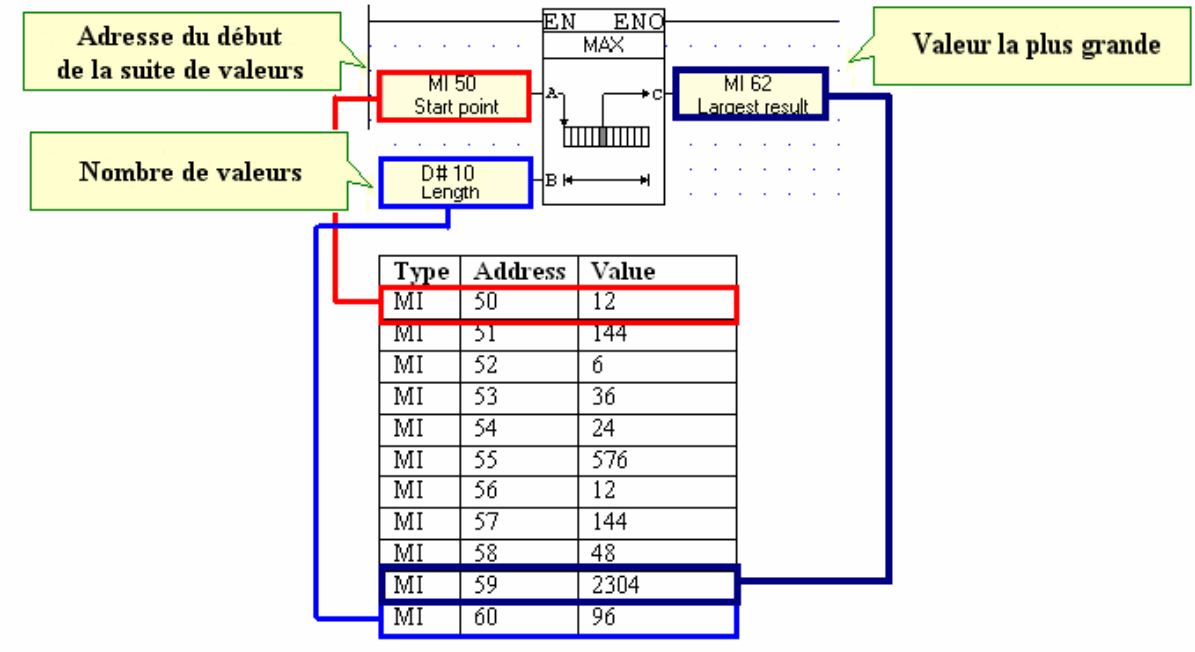
Ces fonctions son utilisées pour convertir une suite de bits en une valeur numérique et vice-versa.



Les fonctions « Vector »

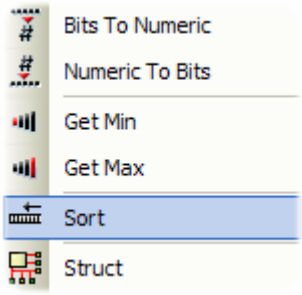
Trouver le maximum ou le minimum :

Ces fonctions servent à chercher le maximum et le minimum dans une suite de valeurs.

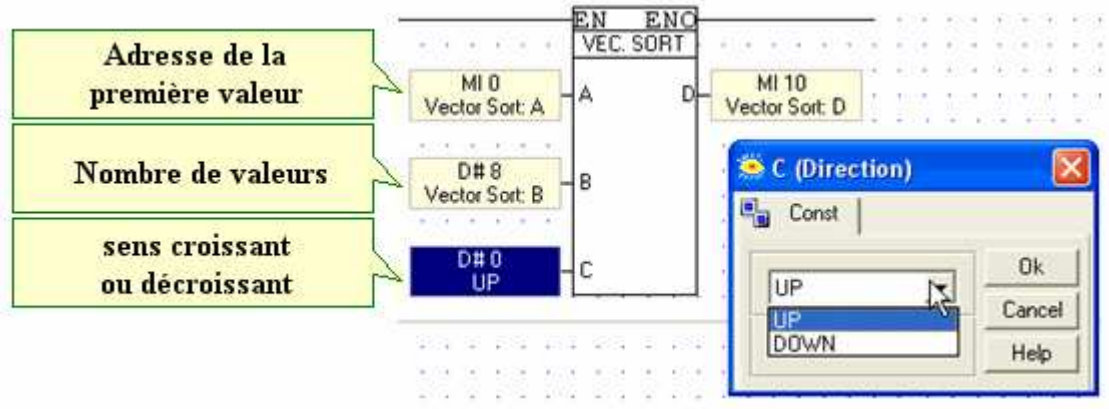


Les fonctions « Vector »

Trier des valeurs par ordre croissant ou décroissant :

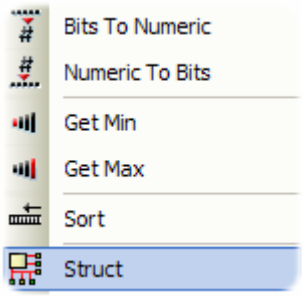


Cette fonction vous permet de trier par ordre croissant ou décroissant une suite de valeurs.

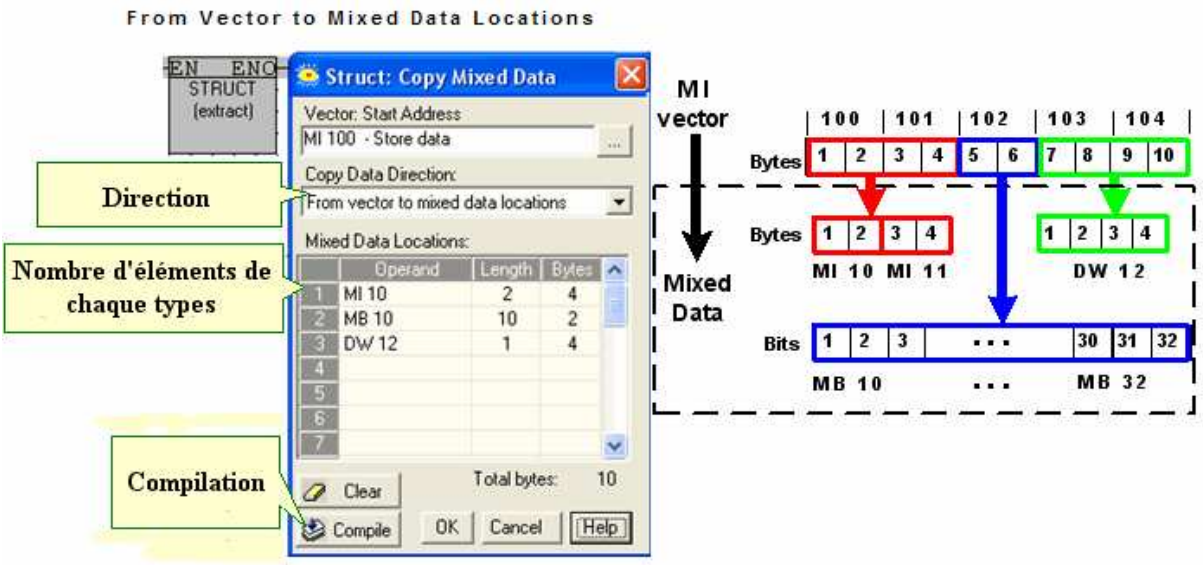


Les fonctions « Vector »

Séparer une suite d'octets ou regrouper des octets séparés :

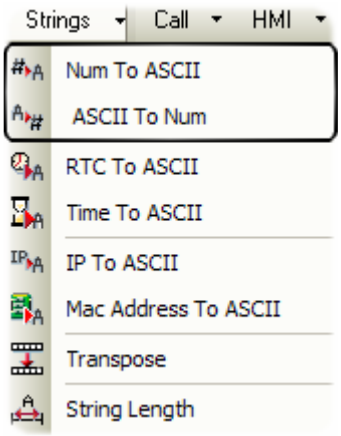


Cette fonction permet de regrouper des octets venant de différends type d'éléments vers une seule suite d'octet composée d'éléments de même nature ou, à l'inverse, de séparer une suite d'octets vers différentes suites d'octets plus petites et de différentes natures.

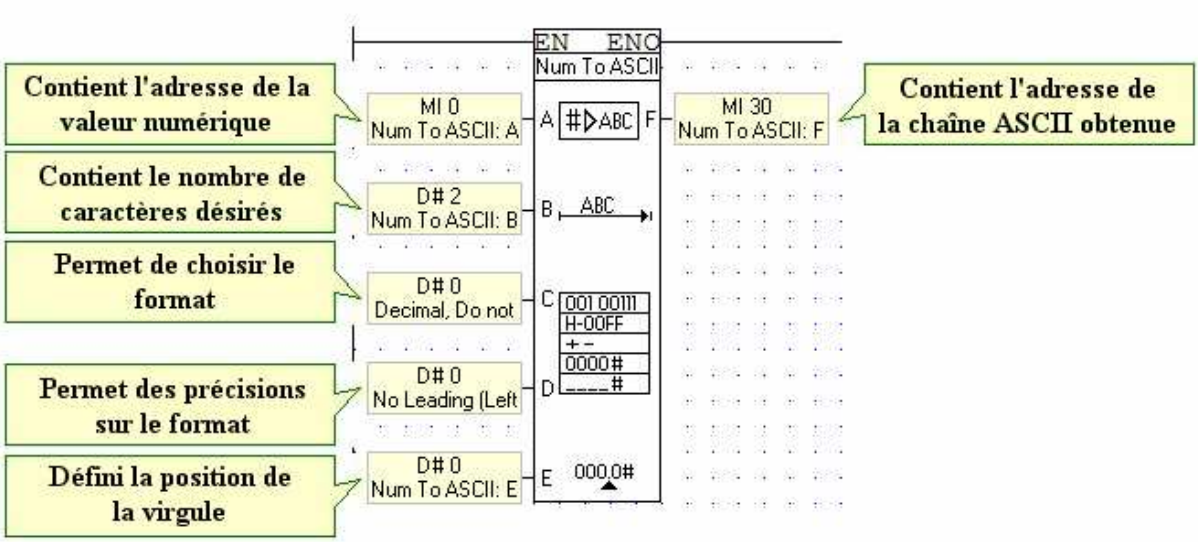


Les chaînes de caractères

Conversion ASCII ⇔ numérique :

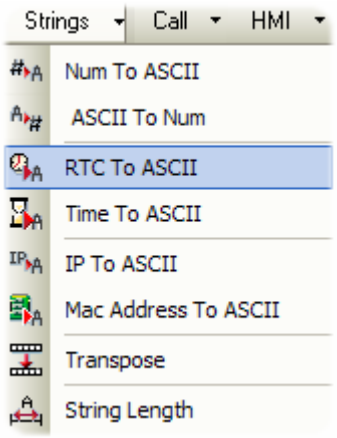


Ces fonctions permettent de passer du format ASCII au format numérique et vice-versa.

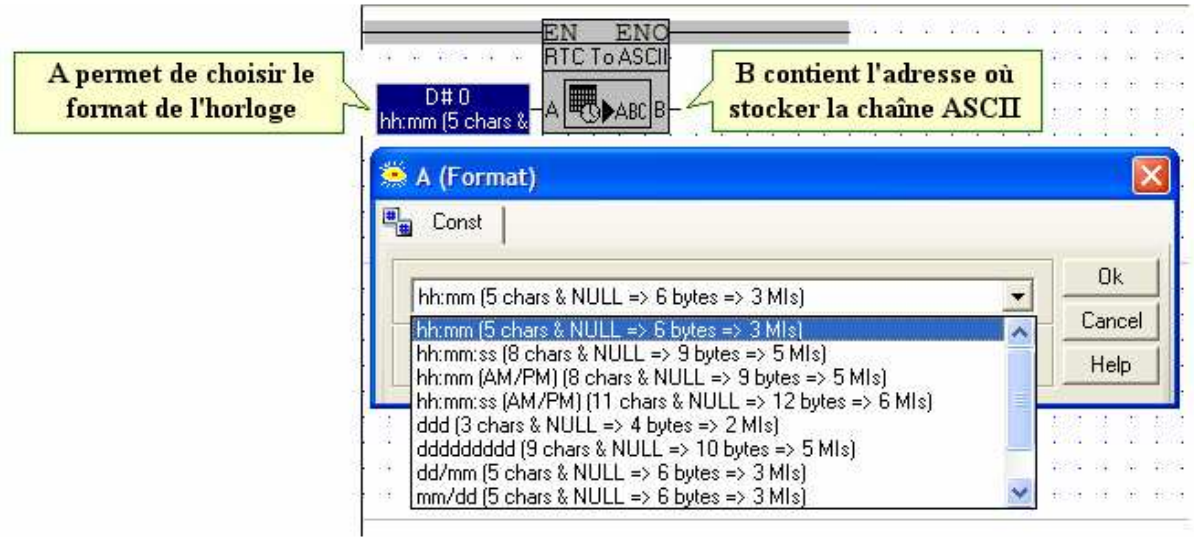


Les chaînes de caractères

Conversion de l'heure et la date en chaîne ASCII :

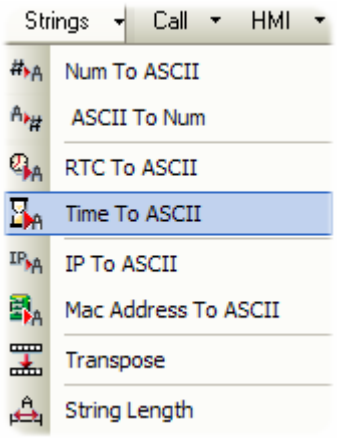


Cette fonction permet de convertir l'heure et la date (de l'horloge interne de l'automate) en une chaîne de caractères ASCII.

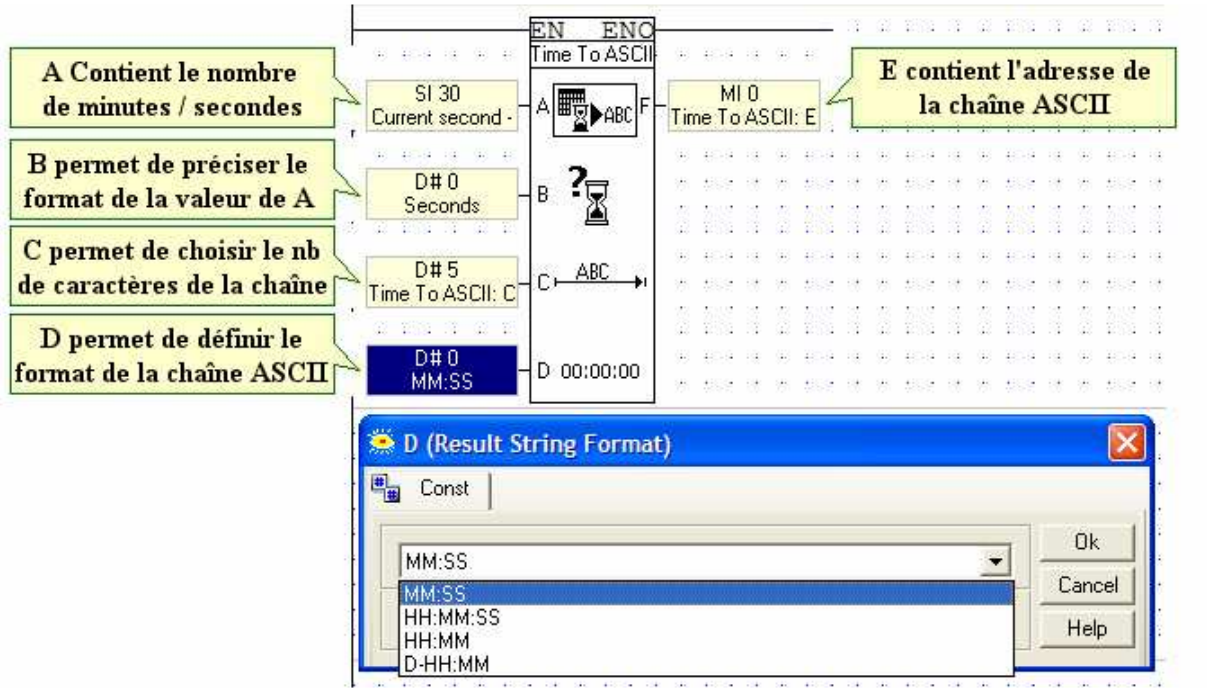


Les chaînes de caractères

Convertir des minutes ou des secondes en une chaîne ASCII :

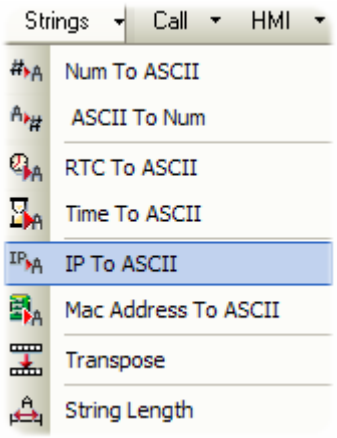


Cette fonction permet de récupérer une valeur correspondant à des secondes ou des minutes afin de la convertir en une chaîne ASCII avec un format défini.

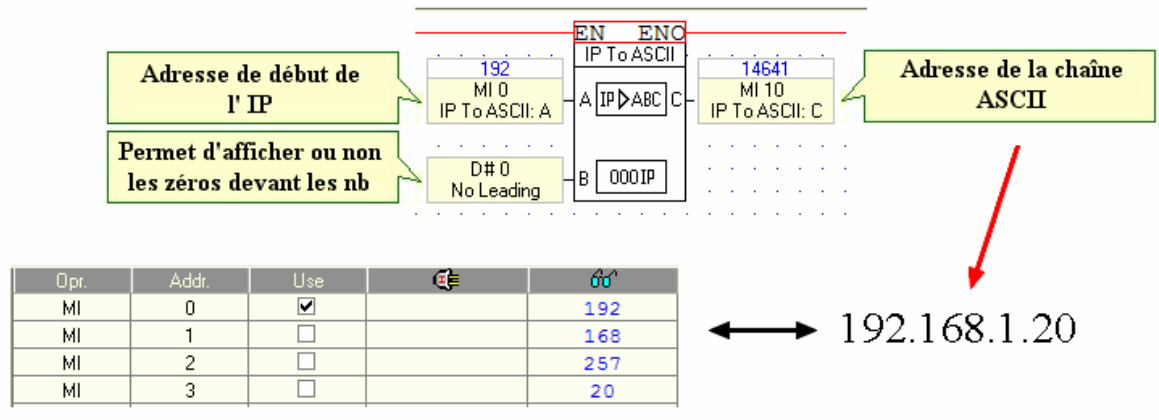


Les chaînes des caractères

Convertir une adresse IP en une chaîne ASCII :



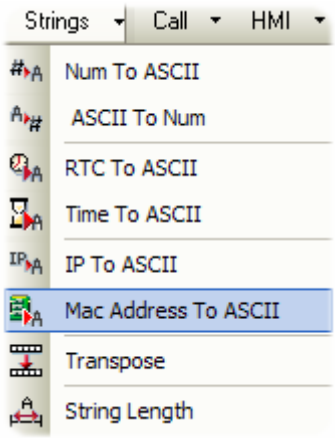
Cette fonction permet de convertir une adresse IP en une chaîne de caractères ASCII.



Remarque : Une adresse IP ne peut dépasser la valeur 255. Donc la valeur 256 deviendra 0, la valeur 257 deviendra 1… La valeur 3000 deviendra 184 (voir ci-dessus).

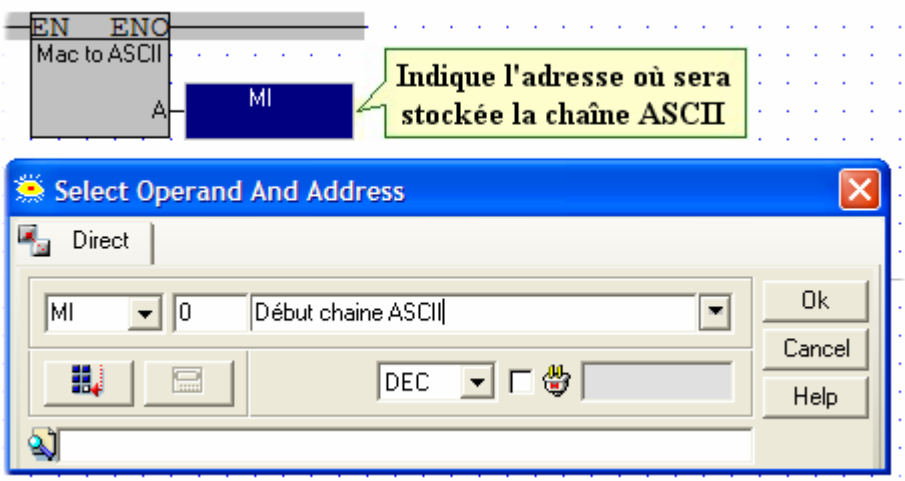
Les chaînes de caractères

Convertir l'adresse MAC en une chaîne ASCII :



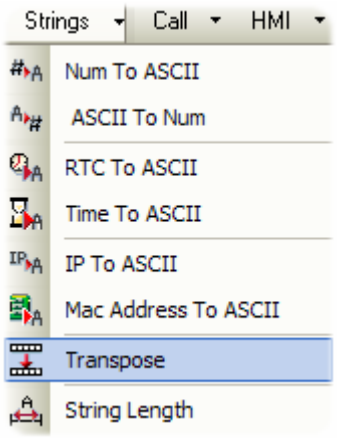
Cette fonction vous permet de convertir l'adresse MAC de votre automate en une chaîne de caractères ASCII.

Remarque : L'adresse MAC ne sera affichée seulement si votre automate contient une carte Ethernet et que cette carte à été initialisée (sinon la chaîne ASCII contiendra que des zéros).



Les chaînes de caractères

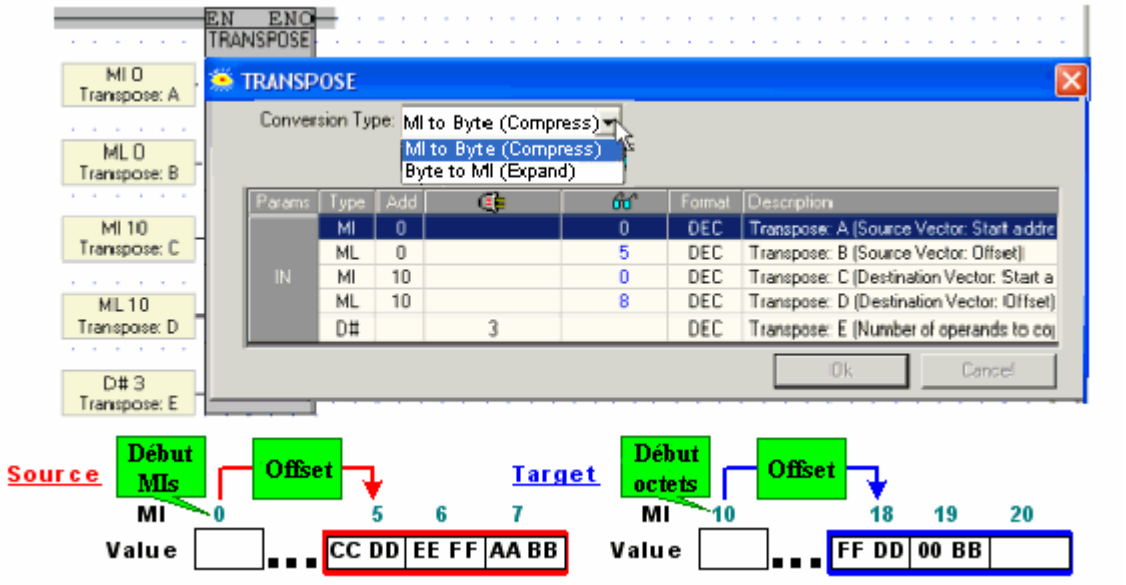
Compression / expansion : Octets ⇔ Mis :



Il y à deux manières d'utiliser cette fonction :

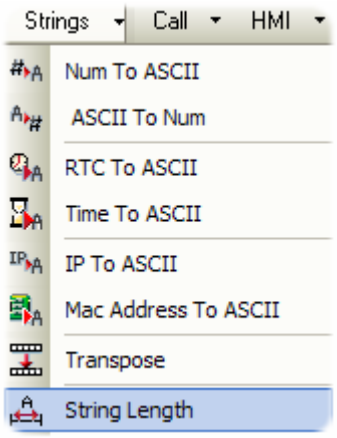
- MIs vers Octets (compression) : permet de copier l'octet de poids faible de chaque MI dans une suite d'octets consécutifs.

- Octets vers MIs (expansion) : permet de séparer les octets consécutifs pour les copier dans les octets de poids faibles de différends Mis

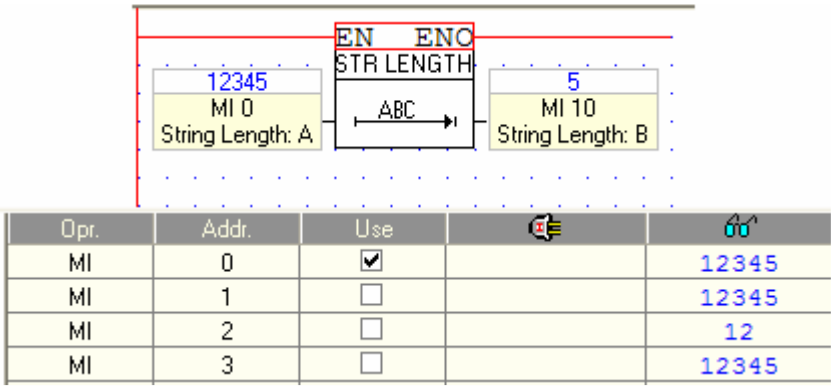


Les chaînes de caractères

Trouver le nombre de caractères dans une chaîne :

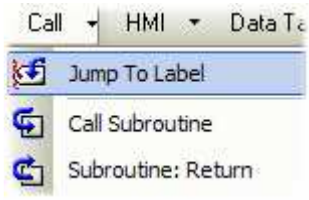


Cette fonction vous permet de compter le nombre de caractères consécutifs. Il faut savoir qu'un mot de 16 bits permet de stocker 2 caractères et que, par conséquent, si un mot ne contient qu'un seul caractère la fonction considère que la chaîne est terminée.

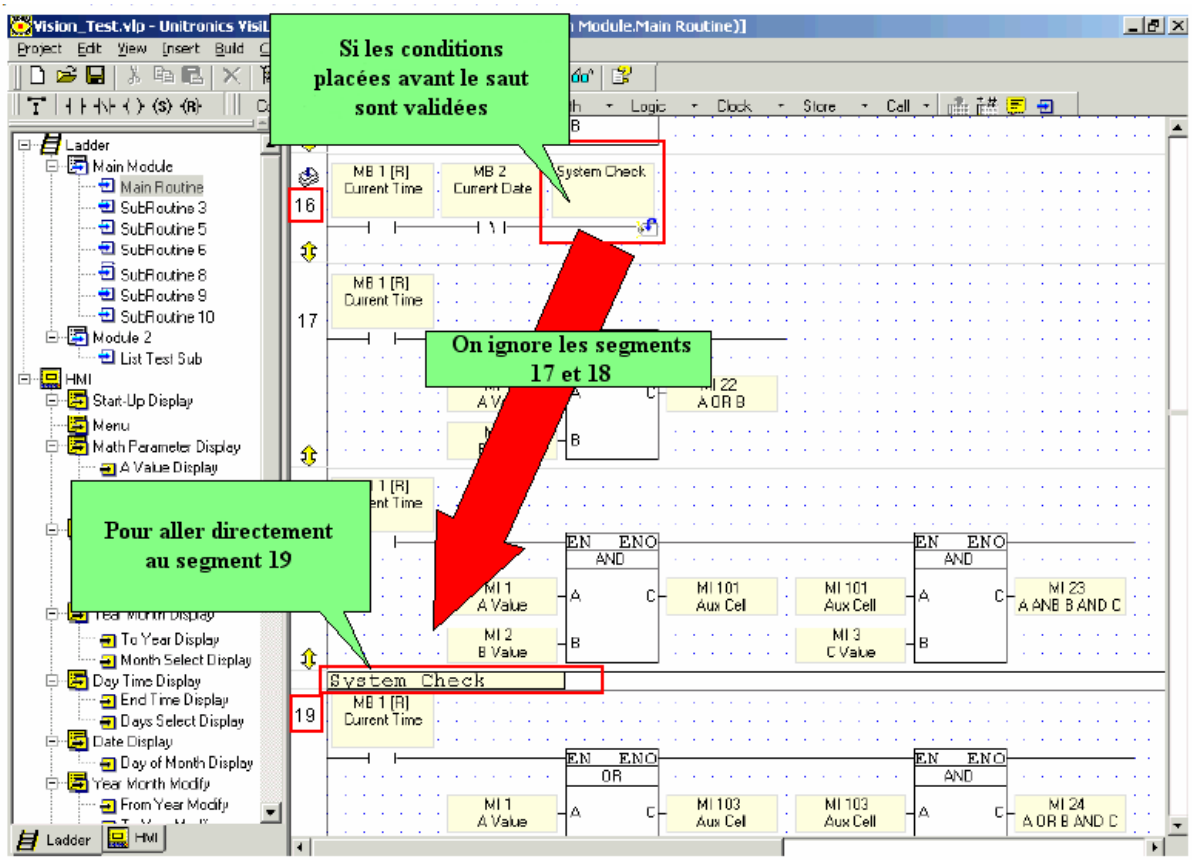


Sauts et sous routines

Sauter une partie du programme (saut vers un label) :

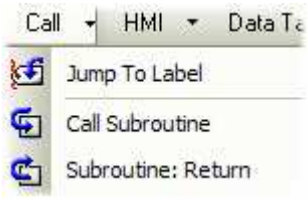


Les labels sont des repères que vous placer dans le programme auxquels vous pouvez revenir quand vous le désirez en ignorant les segments qui se trouvent avant ces labels.

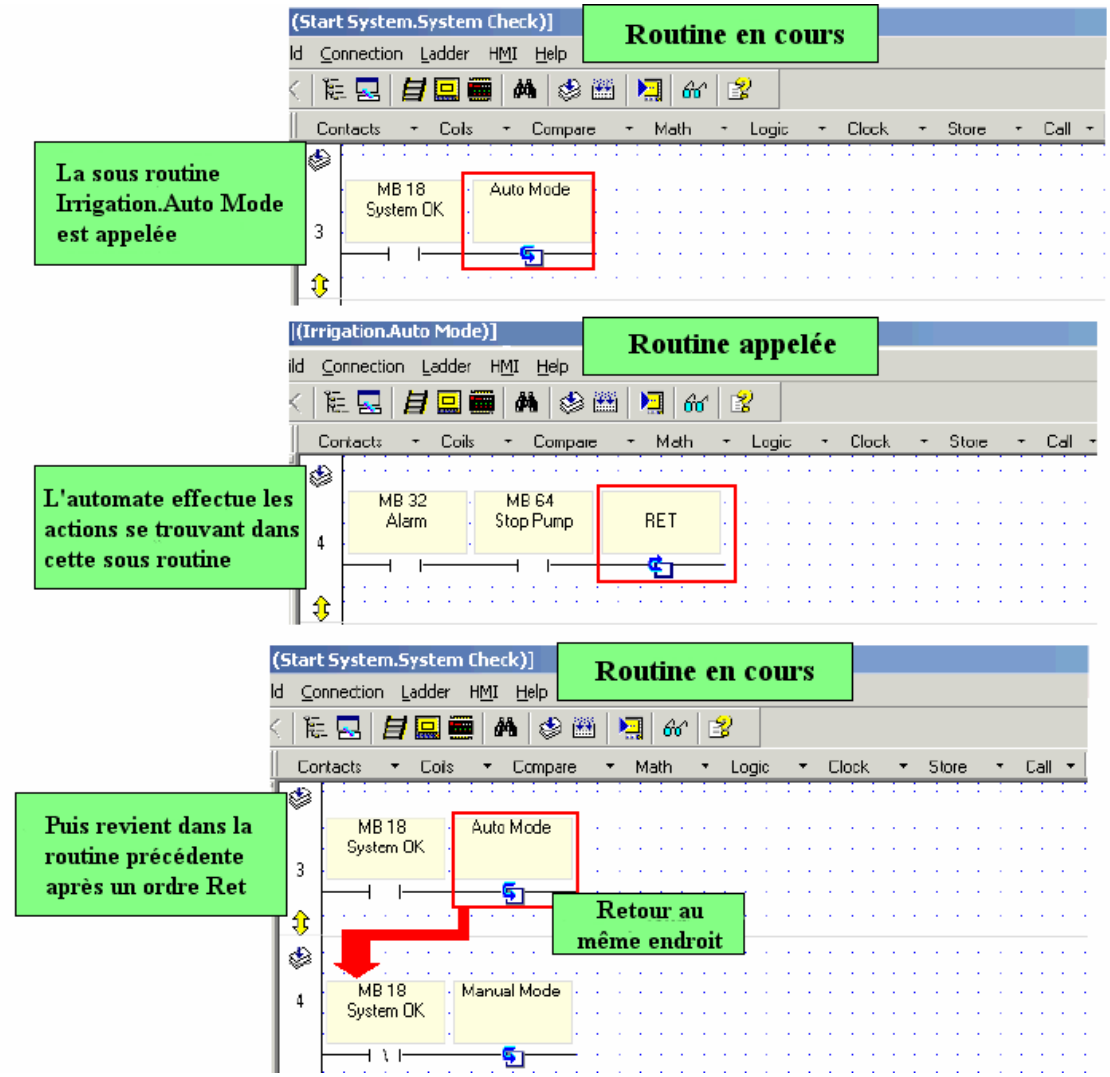


Sauts et sous routines

Appels et retours de sous routines :

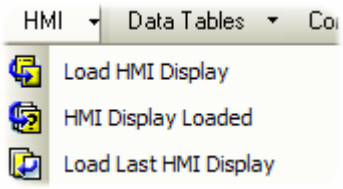


Vous avez la possibilité de placer certaines actions dans des sous routines puis d'appeler ces sous routines n'importe quand dans le programme.



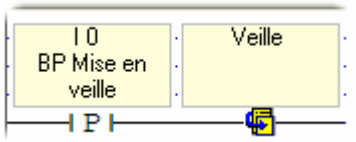
Les fonctions IHM

La gestion des pages :

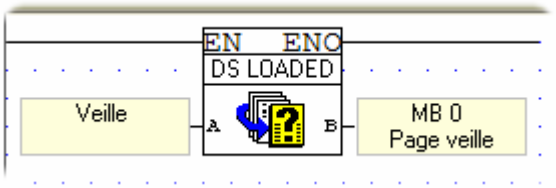


La gestion des pages IHM ne se fait pas uniquement depuis l'éditeur IHM. Il est aussi possible de naviguer entre les différentes pages de l'automate grâce au programme Ladder avec les fonctions suivantes :

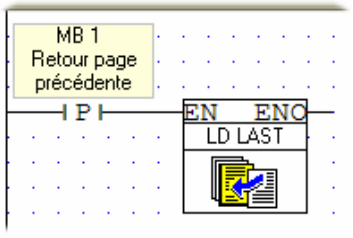
- Load HMI Display : permet d'appeler une page. Il est préférable de l'utiliser à la suite d'une transition (front montant ou descendant).



- HMI Display Loaded : Permet d'associer un bit à une page IHM. Ce bit sera à 1 lorsque la page associée est affichée et restera à 0 dans le cas contraire.

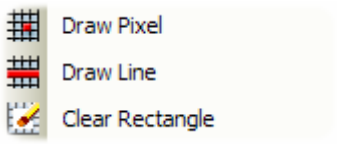


- Load Last HMI Display : Permet d'afficher la page précédente



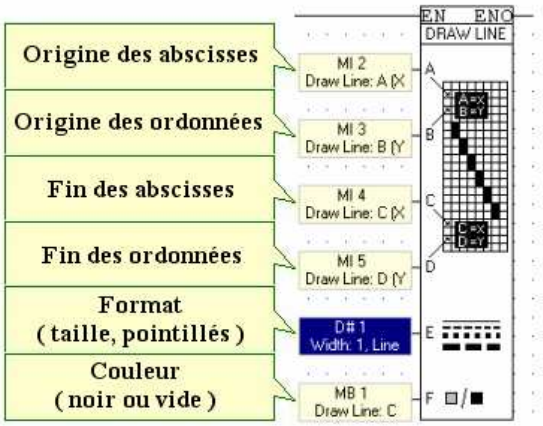
Les fonctions IHM

Les outils de dessin :



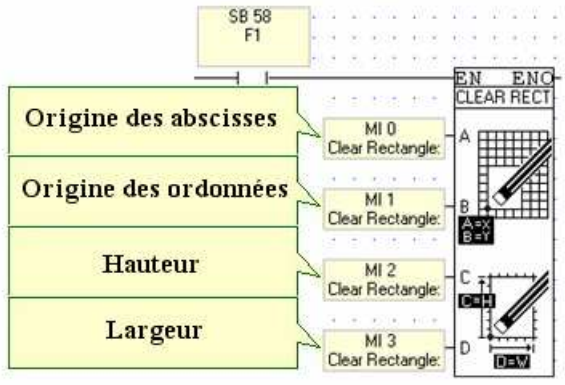
Ces outils vous permettent de dessiner des points, des lignes ou d'effacer une zone de l'écran.

- Dessiner une ligne :



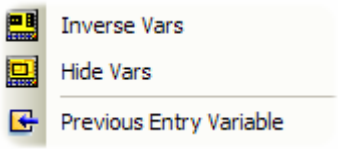
Remarque : L'outil « Draw Pixel » s'utilise de la même façon que l'outil « Draw Line » (voir ci-dessus). La seule différence étant que, pour dessiner un pixel, les paramètres C, D et E ne sont pas demandés.

- Effacer une zone de l'écran :

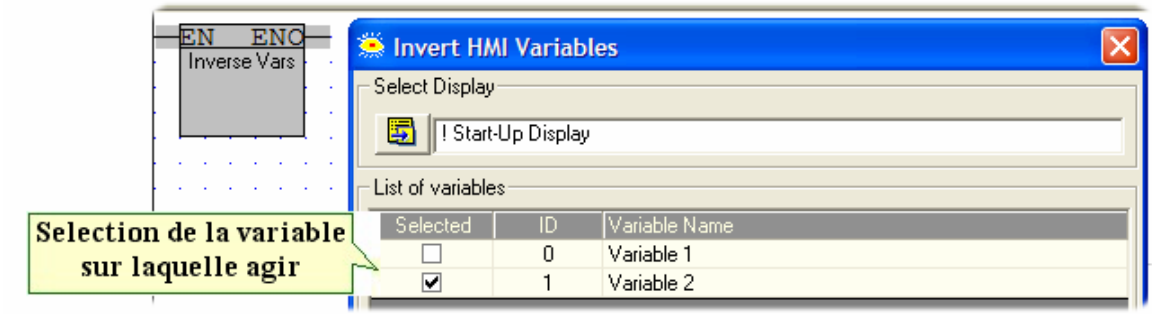


Les fonctions IHM

La gestion de l'affichage des variables :



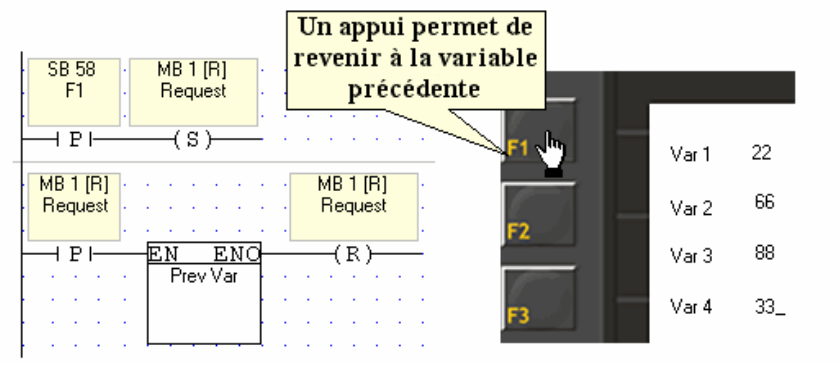
- Inverse Vars : permet de mettre en surbrillance une variable.



- Hide Vars : permet de cacher une variable. Cette fonction se présente de la même façon que la précédente.

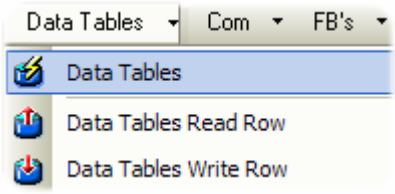
- Previous Entry Variable : lorsqu'il y à plusieurs variables à saisir sur la même page, vous pouvez utiliser cette fonction pour revenir aux variables précédentes.

Dans l'exemple suivant, si la variable 4 est active un appui sur F1 active la variable 3. Un second appui active la variable 2. Un appui lorsque la variable 1 est active entraîne l'activation de la variable 4.

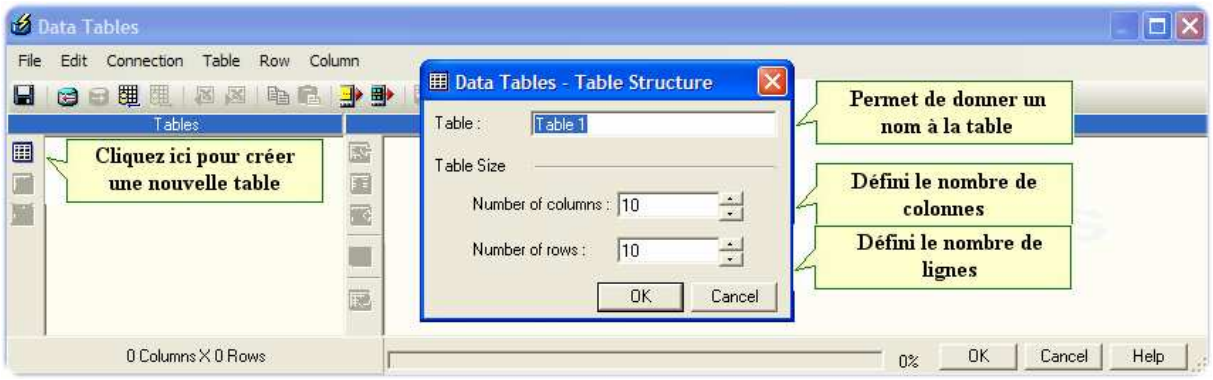


Les tables de données

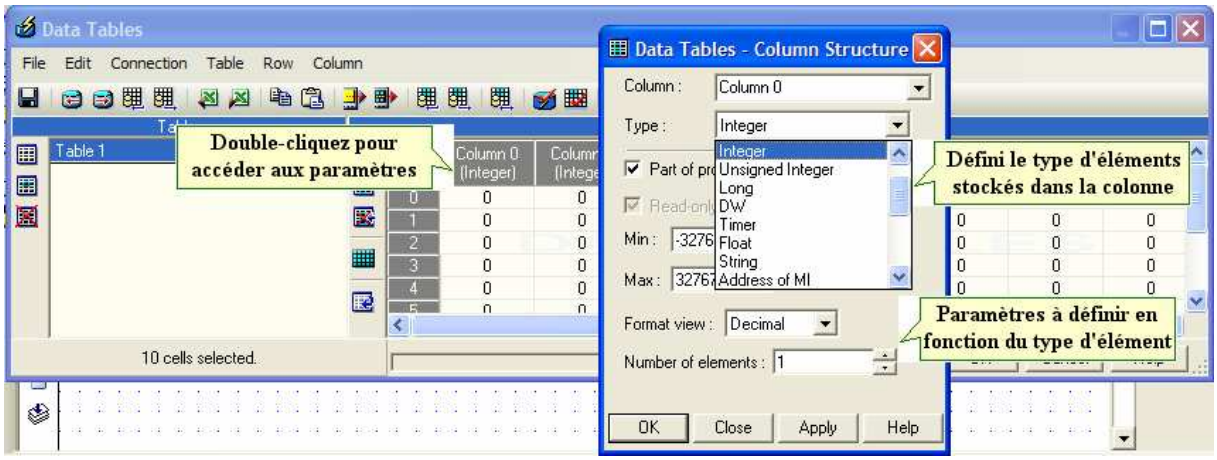
Créer une table de données :



Dans la mémoire de votre automate, il existe 60 000 mots de 16 bits dédiés à la création de tables de données. Ces tables sont peuvent contenir n'importe quel type d'information et permettent donc de stocker une énorme quantité de données.

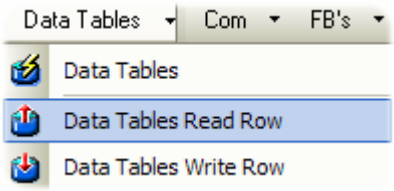


Une fois votre table créée, vous pouvez alors paramétrer chaque colonne :

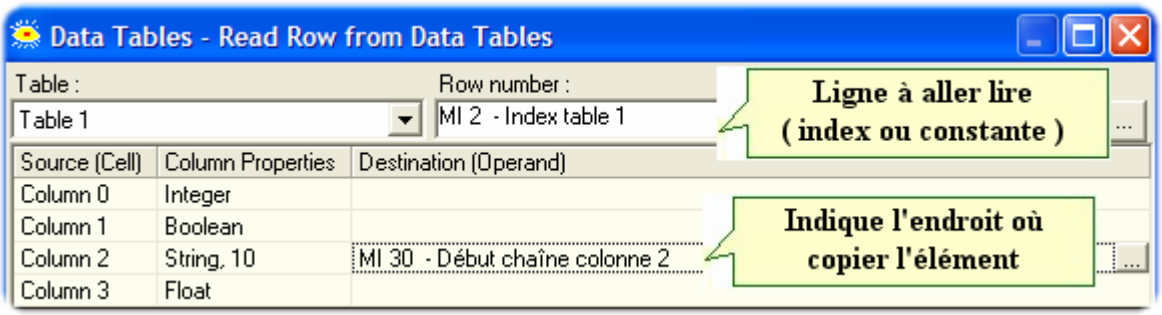


Les tables de données

Lire une ligne de la table :

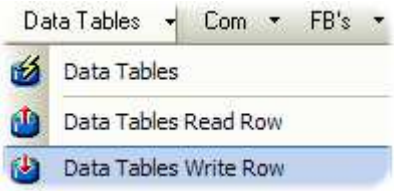


Cette fonction vous permet d'aller lire une ligne de la table et de copier les informations stockées dans les colonnes dans différents éléments auxquels vous avez accès directement dans le programme Ladder.

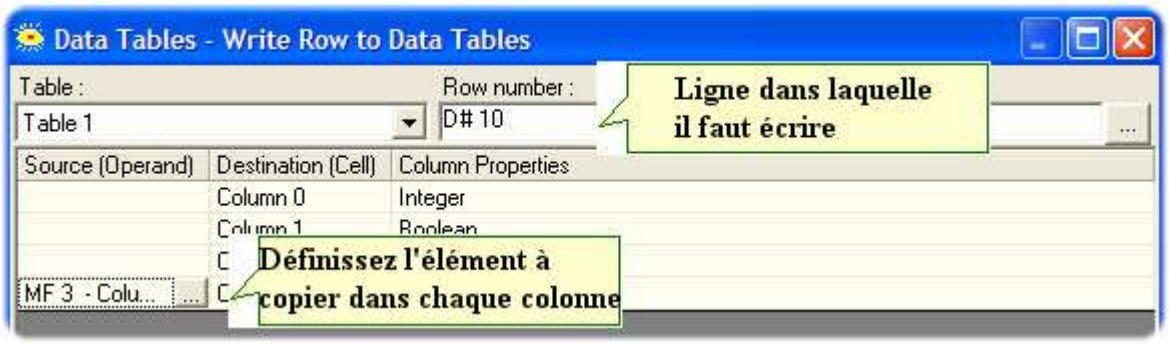


Les tables de données

Ecrire dans une ligne de la table :

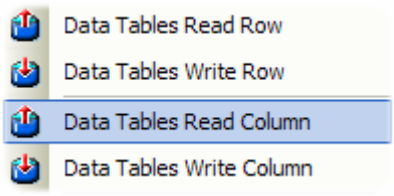


Cette fonction vous permet d'aller écrire dans une ligne de la table afin de stocker vos données.



Les tables de données

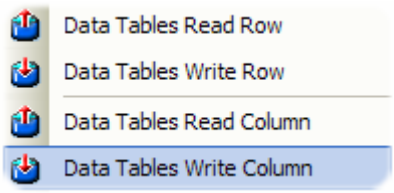
Lire une colonne de la table :



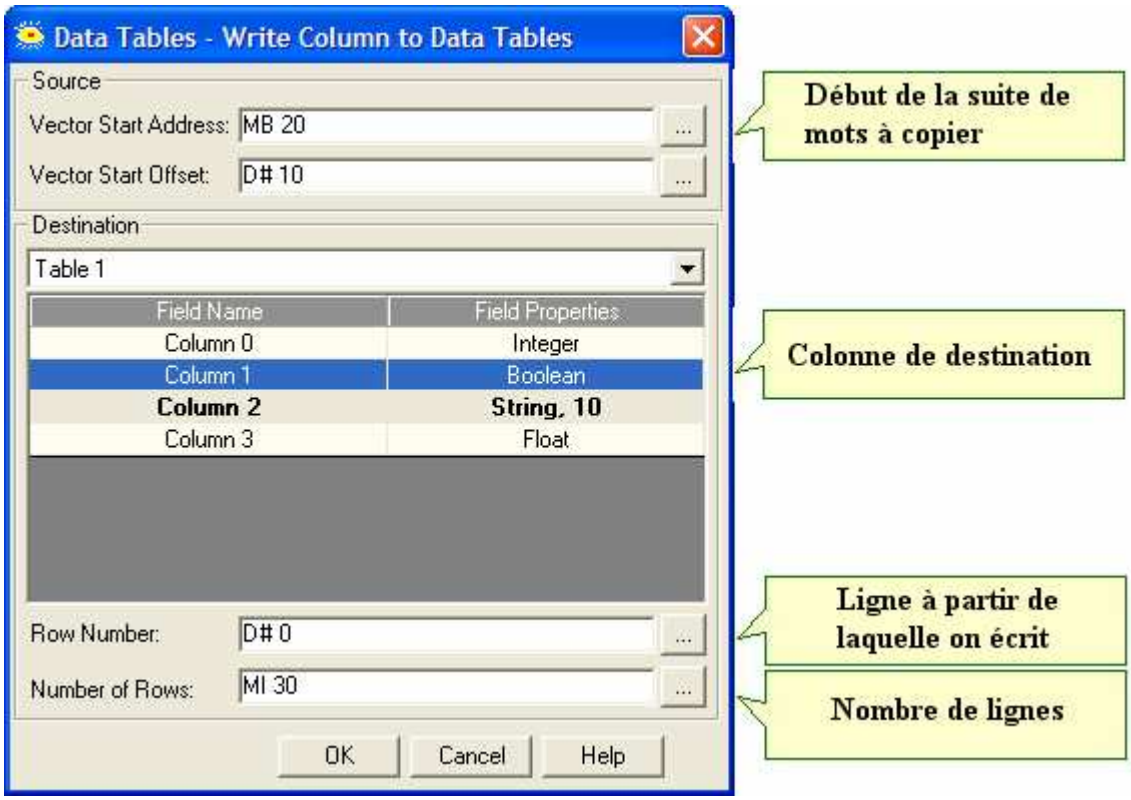
Cette fonction permet de copier une colonne (ou une partie d'une colonne) dans une suite de mots consécutifs.

Les tables de données

Ecrire dans une colonne de la table :



Cette fonction permet de copier une suite de mots consécutifs vers une colonne de la table.

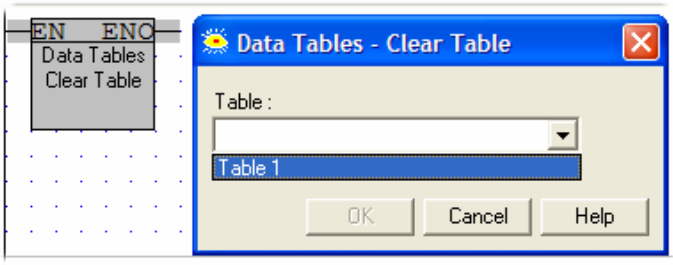


Les tables de données

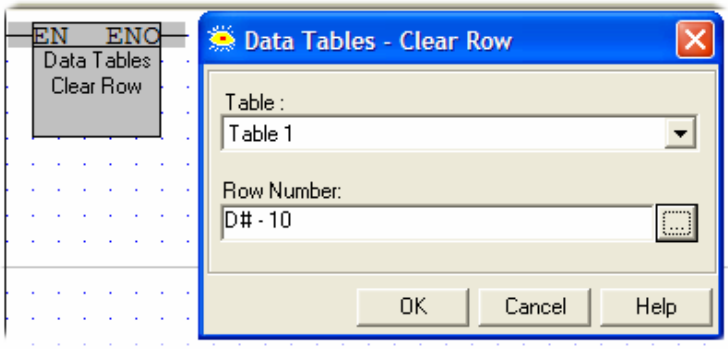
Effacer des valeurs de la table :

Vous pouvez effacer les valeurs qui sont dans la table en y stockant la valeur 0 ou par l'intermédiaire des fonctions suivantes :

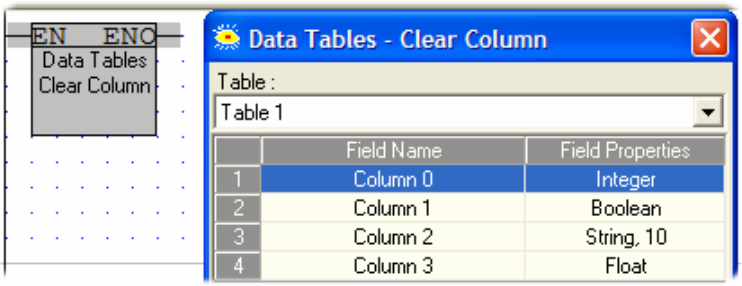
- Clear Table : permet d'effacer toutes les valeurs stockées dans la table



- Clear Row : permet d'effacer une ligne de la table

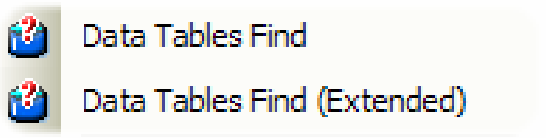


- Clear Column : permet d'effacer un colonne de la table



Les tables de données

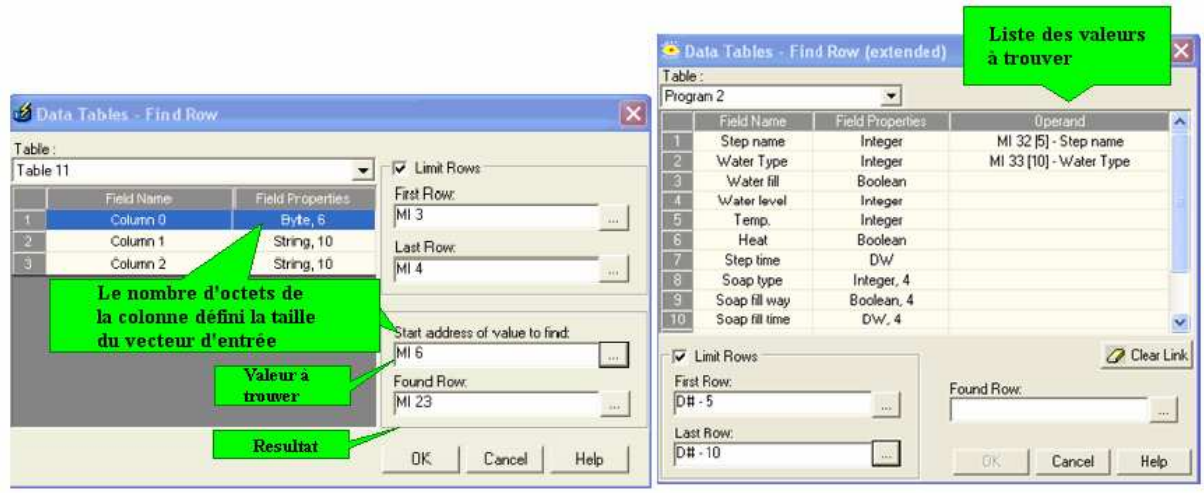
Trouver des valeurs dans la table :



Ces fonctions permettent de trouver une valeur dans la table :

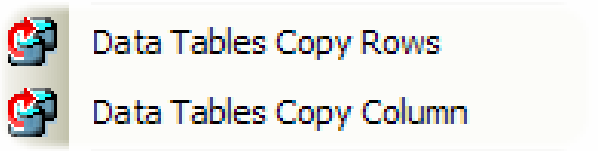
La fonction « Find » compare la valeur désirée avec les valeurs de chaque ligne puis, lorsqu'une ligne contient cette valeur,stocke le numéro de cette ligne dans le résultat.

La fonction « Find (Extented) » agit de la même façon en cherchant cette fois plusieurs valeurs, le résultat correspondra au numéro de la ligne contenant toutes les valeurs recherchées.



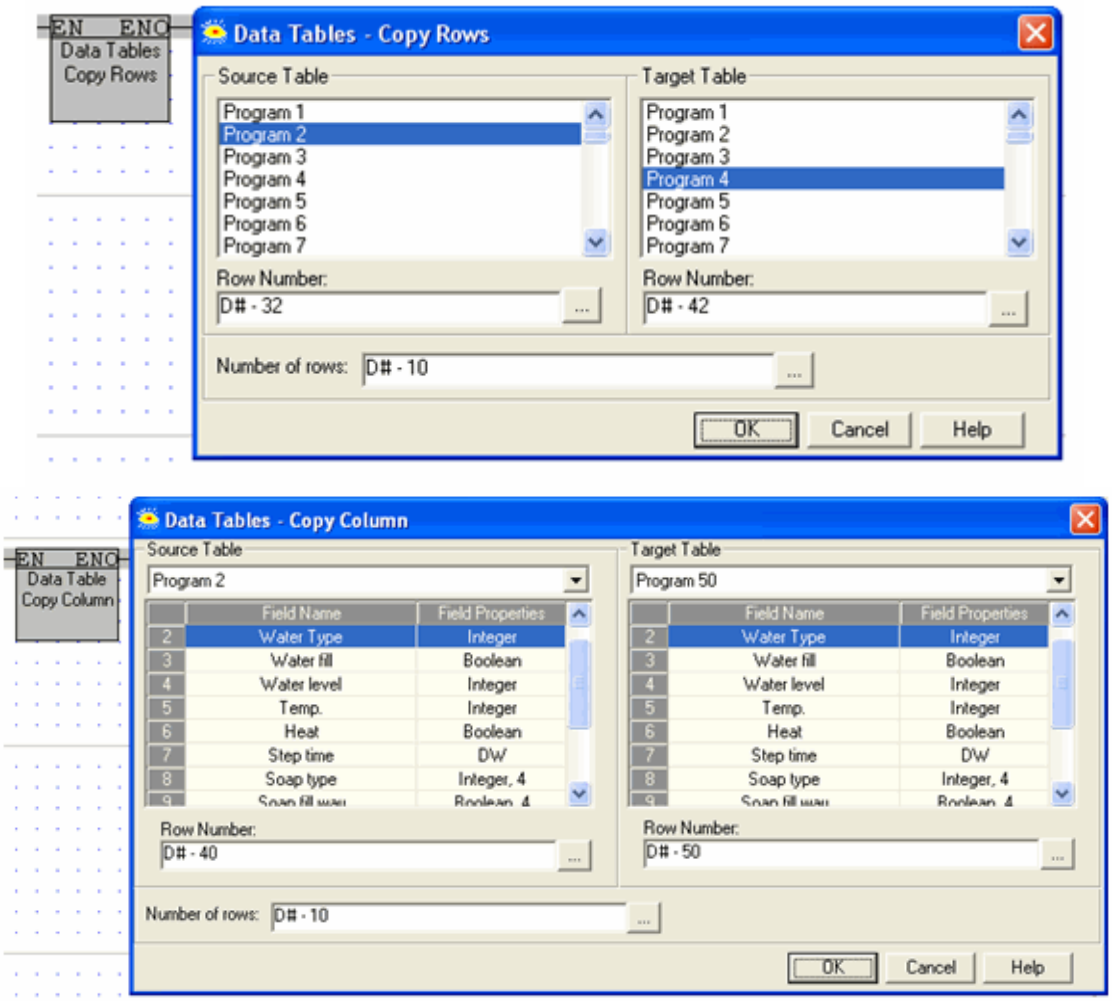
Les tables de données

Copier des lignes ou des colonnes :



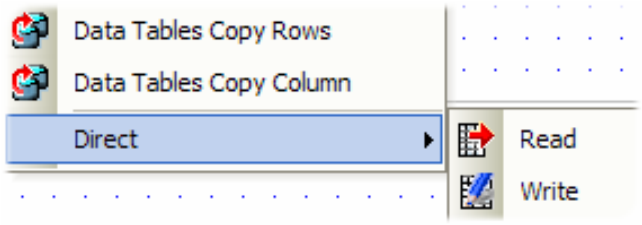
Ces fonctions permettent de copier des morceaux de tables au sein d'une même table ou d'une table à une autre.

Il suffit de sélectionner la ligne de départ, le nombre de lignes et de faire attention à ce que les formats soient compatibles (les deux colonnes doivent contenir le même type d'éléments).



Les tables de données

Lecture / écriture directe :

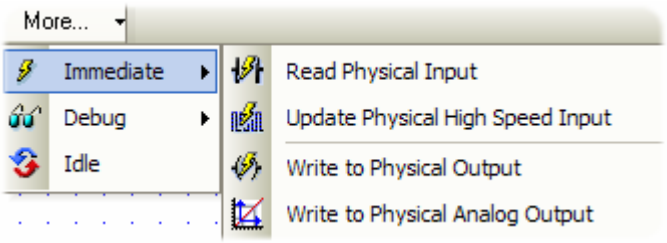


Ces fonctions permettent d'aller lire ou écrire dans les tables sans tenir compte de la structure de ces tables.

Ces fonctions agiront donc sur des suites d'octets sans chercher à savoir si ils ont un rapport entre eux.

Le menu « More… »

Lecture / écriture immédiate :



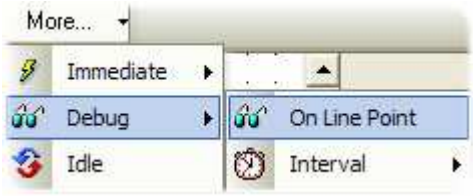
Un cycle automate se décrit de la manière suivante :

- Lecture des entrées
- Lecture du programme
- Mise à jour des sorties

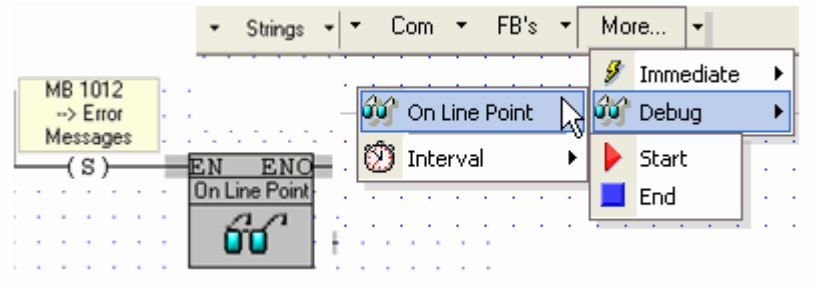
Ces fonctions vous permettent de modifier les valeurs des entrées / sorties sans attendre la fin du cycle automate puis de finir le cycle avec les nouvelles valeurs.

Le menu « More… »

Mettre un point d'arrêt lors du test en ligne :

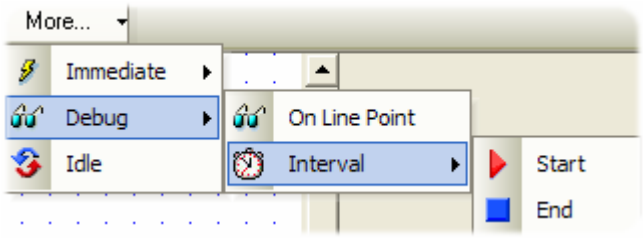


Cette fonction est utile lorsque vous vous mettez en ligne afin de débuger votre programme. Lorsque vous êtes en ligne avec l'automate et que ce bloc s'active, le programme est mis en pause. Cela vous permet de visualiser toutes les valeurs dont vous avez besoin (entrées, sorties, valeurs des temporisations…) au moment exact de l'activation du bloc « On Line Point »



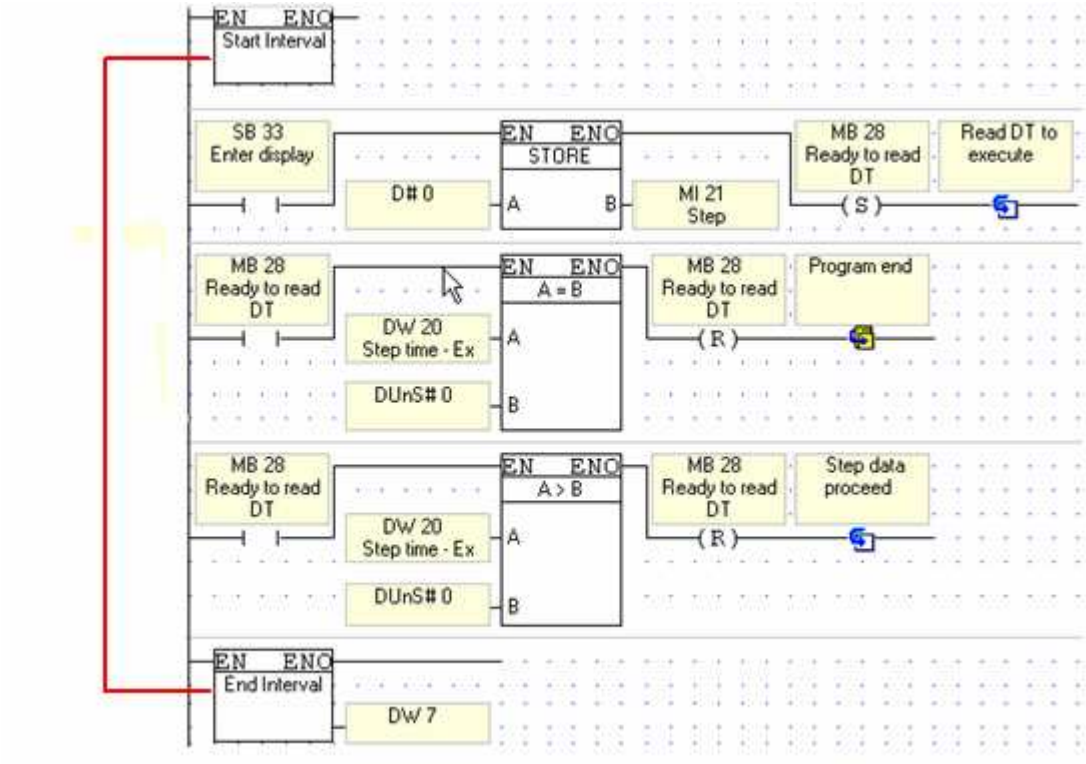
Le menu « More… »

Mesurer l'intervalle de temps entre deux points du programme :



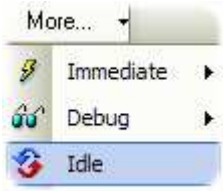
Cette fonction vous permet de mesurer le temps mis par l'automate pour aller du bloc « Start » au bloc « End ».

Le résultat est stocké dans le DW que vous associé au bloc « End ». Il est exprimé en dizaines de microsecondes.



Le menu « More… »

Arrêter le cycle automate :



Cette fonction vous permet de stopper le programme automate pendant un temps donné (exprimé en microsecondes).

Pendant ce temps, toutes les actions sont suspendues (lecture du programme, mise à jour des entrées / sorties …)