



Intégrer un automate M340 ou TSX Premium dans un système Schneider Electric (Unity Pro)

- 08 - Unity Pro: les langages de programmation
FBD, LD, IL, ST

Unity Pro: les langages de programmation FBD, LD, IL, ST

1. PRESENTATION

- LES SECTIONS
- LES SOUS-PROGRAMMES

4

2. LE LANGAGE A BLOCS FONCTION (FBD)

- PRESENTATION
- LA CREATION D'UNE SECTION « FBD »
- LE PARAMETRAGE DES OPTIONS DU PROJET
- LES OUTILS DE L'EDITEUR
- LA SAISIE DES FONCTIONS ELEMENTAIRES (EF)
- LA SAISIE DES BLOCS FONCTIONS ELEMENTAIRES (EFB)
- LA SAISIE DES BLOCS FONCTIONS DERIVES (DFB)
- LES SOUS-PROGRAMMES
- LE SAUT A UNE ETIQUETTE

5

3. LE LANGAGE A CONTACTS (LD)

- PRESENTATION
- LA CREATION D'UNE SECTION « LD »
- LE PARAMETRAGE DES OPTIONS DES SECTIONS (LD)
- LES ELEMENTS DE TEST
- LES ELEMENTS D'ACTIONS BOOLEENNES
- LES EXPRESSIONS NUMERIQUES
- LES SAUTS DANS UNE SECTION
- LES COMMENTAIRES
- LA SAISIE DES FONCTIONS

12

4. LE LANGAGE LISTE D'INSTRUCTIONS (IL)

- PRESENTATION
- LA CREATION D'UNE SECTION « IL »
- LES INSTRUCTIONS BOOLEENNES
- LES INSTRUCTIONS ARITHMETIQUES
- LES INSTRUCTIONS DE COMPARAISON
- LES INSTRUCTIONS D'APPEL
- LES INSTRUCTIONS DE SAUT
- LA SAISIE DES FONCTIONS

16

5. LE LANGAGE LITTERAL STRUCTURE (ST)

19

- PRESENTATION
- LA CREATION D'UNE SECTION « ST »
- LE PARAMETRAGE DES OPTIONS DU PROJET
- LES OPERATEURS DE BASE
- LES INSTRUCTIONS D'AFFECTATION
- L'INSTRUCTION « APPEL D'UN SOUS-PROGRAMME »
- L'INSTRUCTION « APPEL D'UNE FONCTION »
- LA STRUCTURE CONDITIONNELLE « IF.....THEN.....END_IF »
- LA STRUCTURE CONDITIONNELLE « IF...THEN...ELSE...END_IF »
- LA STRUCTURE CONDITIONNELLE « CASE...OF...ELSE...END_CASE »
- LA STRUCTURE REPETITIVE « FOR...TO...DO...END_FOR »
- LA STRUCTURE ITERATIVE CONDITIONNELLE « WHILE...DO...END_WHILE »
- LA STRUCTURE ITERATIVE CONDITIONNELLE « REPEAT...UNTIL...END_REPEAT »
- L'INSTRUCTION RECURRENTE « EXIT »

institut des ressources
industrielles
- AFPI Lyon -

Unity Pro: les langages de programmation FBD, LD, IL, ST

1. PRESENTATION

□ LES SECTIONS

- ✓ Les sections sont des unités de programme autonomes dans lesquelles est créée la logique du projet.
- ✓ Les sections sont exécutées dans leur ordre de représentation dans le navigateur de projet (vue structurelle).
- ✓ Une même section ne peut pas appartenir à plusieurs tâches en même temps.
- ✓ Une section peut être programmée dans **5 langages de programmation** conformes à la norme CEI 61131-3 :
 - langages à contacts (**LD**),
 - liste d'instructions (**IL**),
 - littéral structuré (**ST**),
 - diagramme fonctionnel en séquence (**SFC**),
 - langage en blocs fonctionnels (**FBD**).

□ LES SOUS-PROGRAMMES

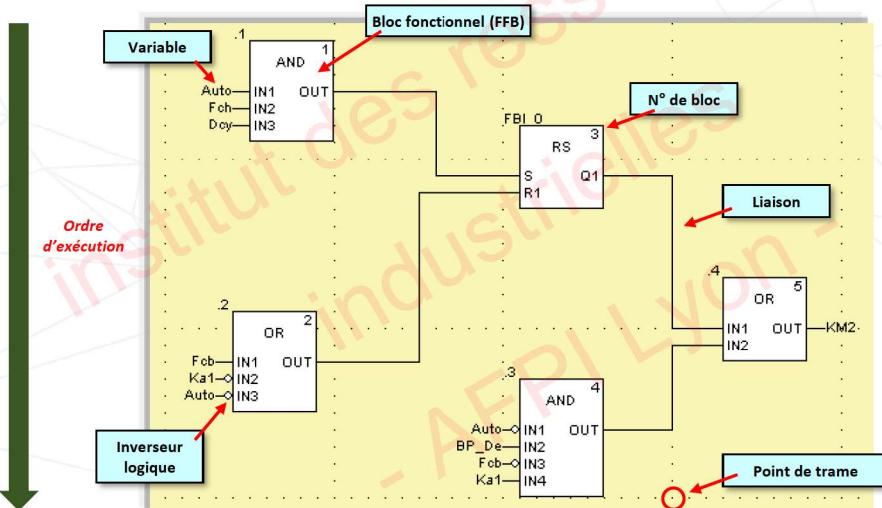
- ✓ Les sous-programmes sont créés en tant qu'unités distinctes dans des sections de sous-programme.
- ✓ Les appels aux sous-programmes s'effectuent à partir des sections ou d'un autre sous-programme. Une imbrication de huit niveaux maximum est possible.
- ✓ Une même section ne peut pas appartenir à plusieurs tâches en même temps.
- ✓ Les sous-programmes sont affectés à une tâche. Un même sous-programme ne peut pas être appelé par différentes tâches.
- ✓ Un sous-programme peut être programmé dans **4 langages de programmation**:
 - langages à contacts (**LD**),
 - liste d'instructions (**IL**),
 - littéral structuré (**ST**),
 - langage en blocs fonctionnels (**FBD**).

Unity Pro: les langages de programmation FBD, LD, IL, ST

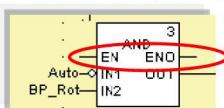
1. LE LANGAGE A BLOCS FONCTION (FBD)

□ PRESENTATION

- ✓ L'éditeur FBD permet la **programmation graphique de blocs fonction** conformément à la norme CEI 61131-3
- ✓ Une section FBD se compose:
 - d'une fenêtre comportant une seule page,
 - cette **fenêtre** comporte **30 unités de grille horizontalement et 23 unités verticalement**,
 - chaque **unité** se compose de **100 points de trame**.
- ✓ Les **points de trame** servent:
 - à **positionner les blocs fonctionnels**,
 - à déterminer l'**ordre d'exécution** des blocs.
- ✓ Les **fonctions** sont exécutées dans l'**ordre des numéros de bloc**.



- ✓ L'entrée EN (enable) et la sortie ENO ne sont pas obligatoires.

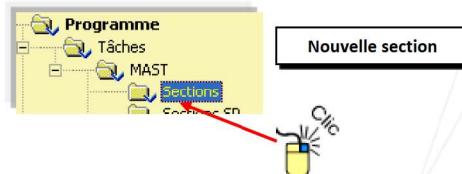


- ✓ Lorsque l'entrée EN est activée, une sortie ENO est automatiquement établie.
- ✓ Si EN = 0, le bloc n'est pas activé, son programme interne n'est pas exécuté et ENO est à 0.
- ✓ Si EN = 1, le programme interne du bloc est exécuté et ENO est à 1. Si une erreur survient, ENO prend la valeur 0.
- ✓ Si l'entrée EN n'est pas connectée, elle est automatiquement réglée sur 1.

Unity Pro: les langages de programmation FBD, LD, IL, ST

5. LE LANGAGE A BLOCS FONCTION (FBD)

□ LA CREATION D'UNE SECTION « FBD »

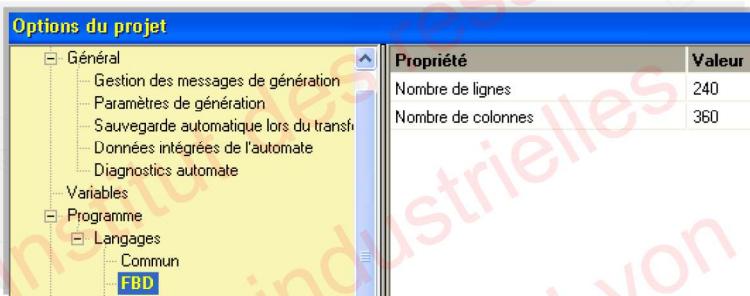


- ✓ Nommer la section (ex: POMPE_4)
- ✓ Sélectionner le langage ST
- ✓ Valider la section

□ LE PARAMETRAGE DES OPTIONS DU PROJET

Outils

Options du projet...



□ LES OUTILS DE L'EDITEUR



ASSISTANT de SAISIE des DONNEES

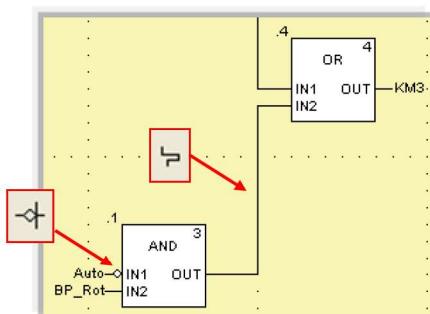
ASSISTANT de SAISIE des FONCTIONS

NAVIGATEUR des BIBLIOTHEQUES des TYPES

APPEL d'un SOUS-PROGRAMME

LIAISON

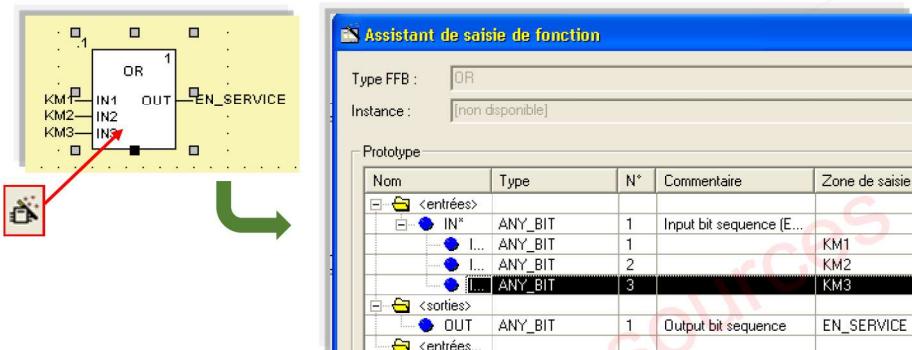
INVERSEUR LOGIQUE



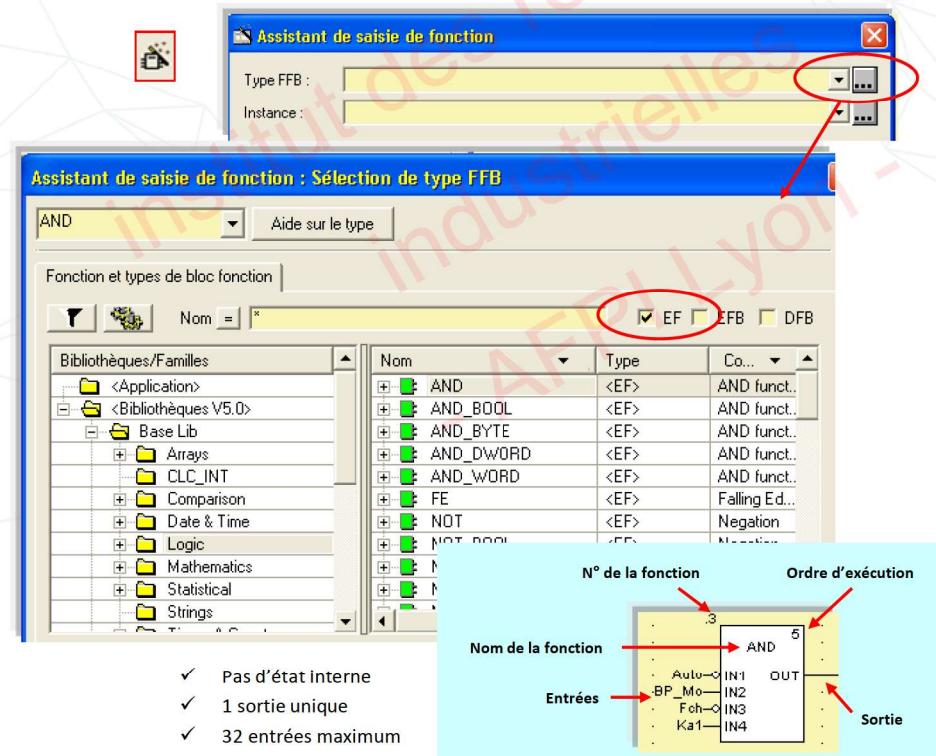
Unity Pro: les langages de programmation FBD, LD, IL, ST

5. LE LANGAGE A BLOCS FONCTION (FBD)

□ LES OUTILS DE L'EDITEUR



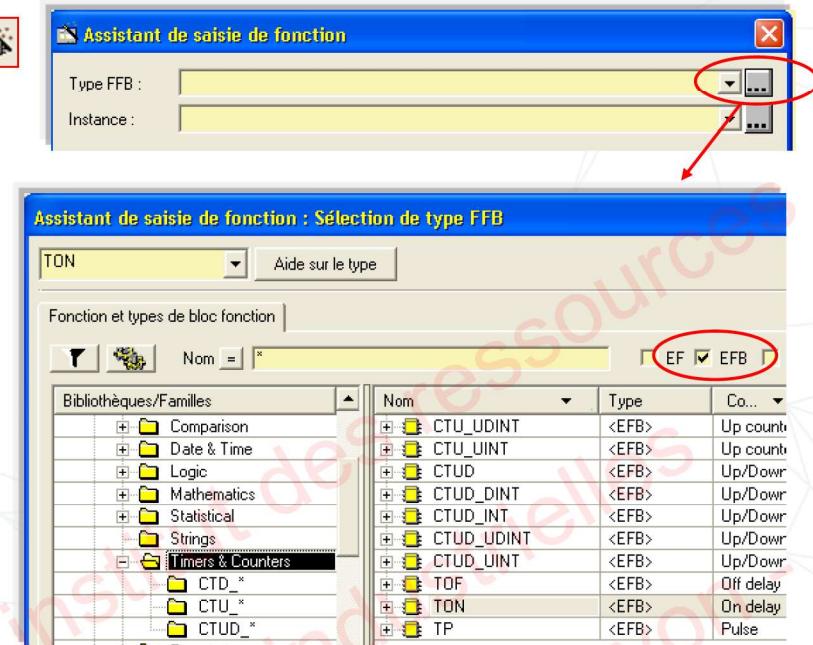
□ LA SAISIE DES FONCTIONS ELEMENTAIRES (EF)



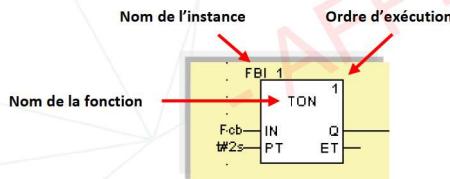
Unity Pro: les langages de programmation FBD, LD, IL, ST

5. LE LANGAGE A BLOCS FONCTION (FBD)

□ LA SAISIE DES BLOCS FONCTIONS ELEMENTAIRES (EFB)



- ✓ Exemple: temporisations/compteurs TOF, TON, TP, CTU, CTD.....

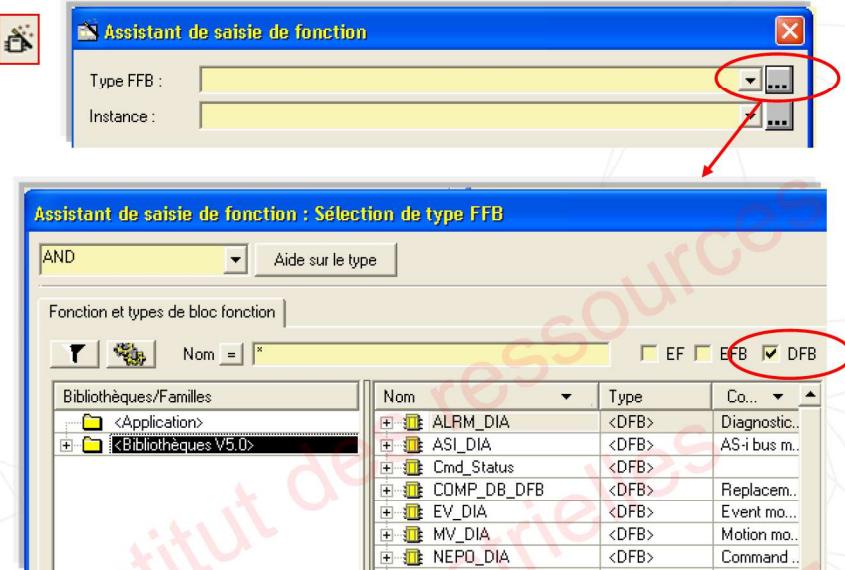


- ✓ 32 sorties maximum
- ✓ 32 entrées maximum
- ✓ Des états internes
- ✓ L'algorithme de traitement est développé en langage C

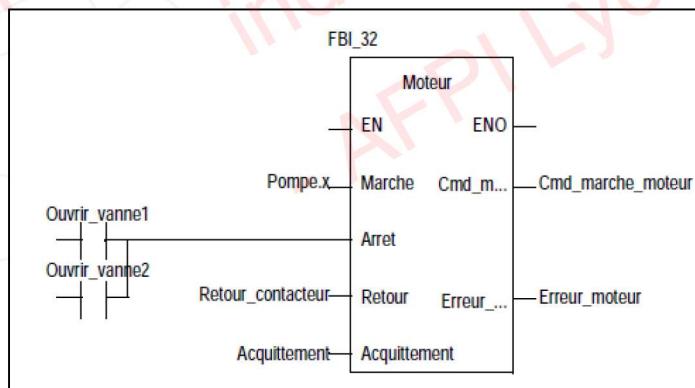
Unity Pro: les langages de programmation FBD, LD, IL, ST

5. LE LANGAGE A BLOCS FONCTION (FBD)

□ LA SAISIE DES BLOCS FONCTIONS DERIVES (DFB)



✓ Exemple: « Gestion de la commande d'un moteur ».....



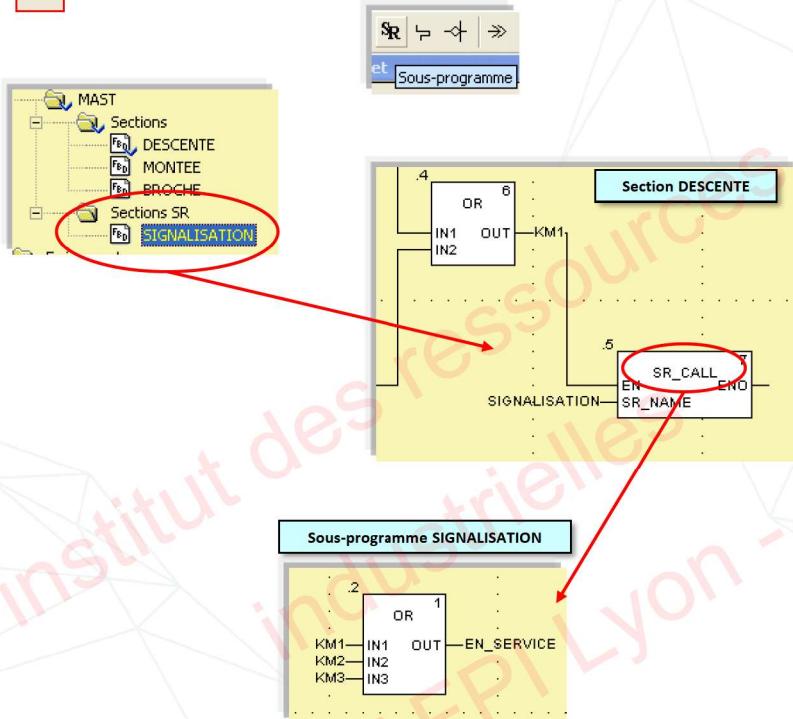
- ✓ 32 sorties maximum
- ✓ 32 entrées maximum
- ✓ Des états internes
- ✓ L'algorithme de traitement est développé en langage d'automatismes (LD, IL, ST, FBD).

Unity Pro: les langages de programmation FBD, LD, IL, ST

5. LE LANGAGE A BLOCS FONCTION (FBD)

□ LES SOUS-PROGRAMMES

- ✓  : Appel d'un sous-programme



- ✓  : Retour conditionnel
- ✓ Le sous-programme à appeler doit se trouver dans la même tâche que la section FBD appelante.
- ✓ Il est possible d'appeler des sous-programmes au sein de sous-programmes

Unity Pro: les langages de programmation FBD, LD, IL, ST

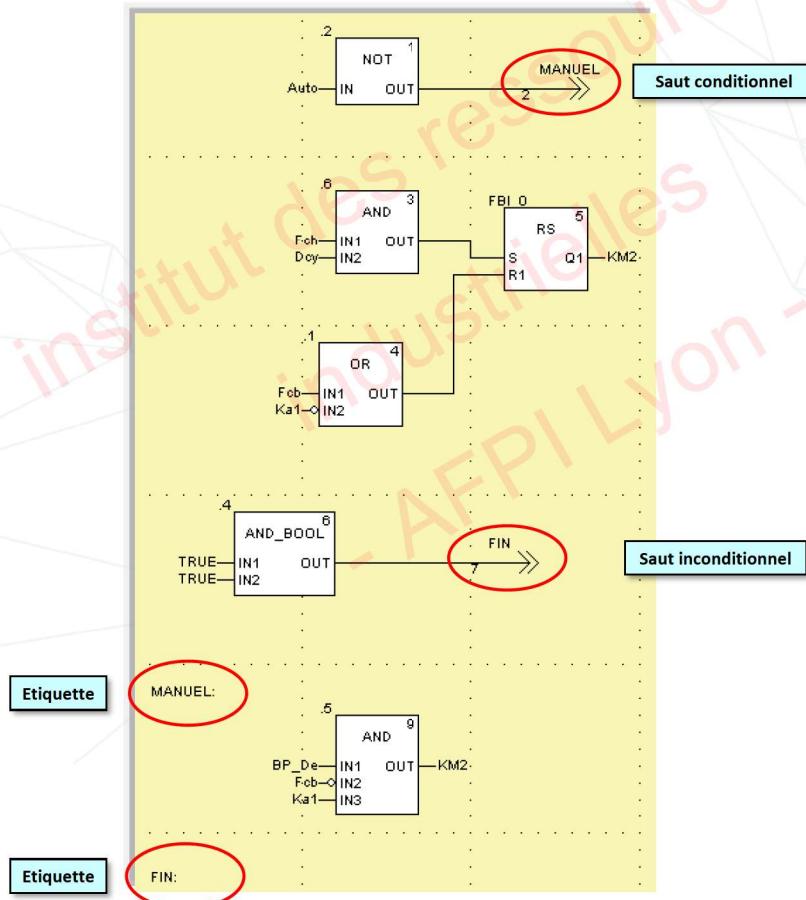
5. LE LANGAGE A BLOCS FONCTION (FBD)

□ LE SAUT A UNE ETIQUETTE

✓ ➔ : Saut à une étiquette

JL: : Etiquette de saut

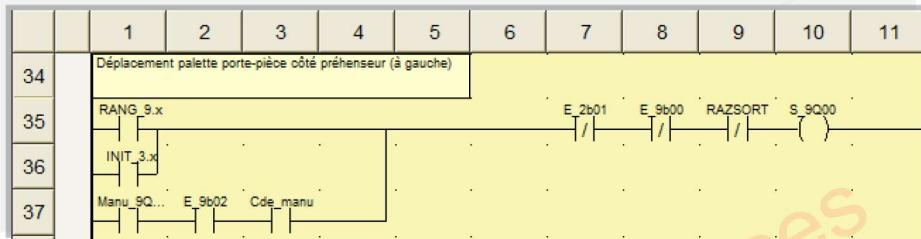
- un **saut** est exécuté jusqu'à l'**étiquette** située dans la **section courante**.
- Pour générer un **saut conditionnel**, l'objet « **saut** » est lié à une **sortie FFB booléenne**.
- Pour générer un **saut inconditionnel**, la valeur **1** est affectée à l'objet « **saut** » via la fonction **AND**.
- Le **texte de l'étiquette** est limité à **32 caractères** et se termine par « **:** », placé automatiquement.



Unity Pro: les langages de programmation FBD, LD, IL, ST

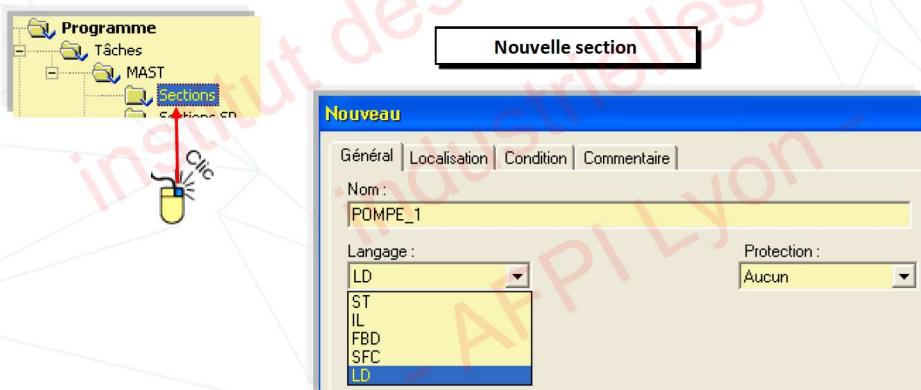
2. LE LANGAGE A CONTACTS (LD)

□ PRESENTATION

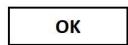


- ✓ Les sections LD disposent d'une grille d'arrière-plan qui divise la section en lignes et en colonnes (**11 à 64 colonnes; 7 à 2000 lignes**).
- ✓ Le langage de programmation LD est axé sur les cellules, ce qui signifie qu'un seul objet peut être placé dans chaque cellule.

□ LA CREATION D'UNE SECTION « LD »



- ✓ Nommer la section (ex: POMPE_1)
- ✓ Sélectionner le langage LD
- ✓ Valider la section:



Unity Pro: les langages de programmation FBD, LD, IL, ST

2. LE LANGAGE A CONTACTS (LD)

□ LE PARAMETRAGE DES OPTIONS DES SECTIONS (LD)

Outils **Options du projet...**

Options du projet

- Options du projet
 - Général
 - Gestion des messages de génération
 - Paramètres de génération
 - Sauvegarde automatique lors du transf.
 - Données intégrées de l'automate
 - Diagnostics automatique
 - Variables
 - Programme
 - Langages
 - Commun
 - FBD

Propriété

Propriété	Valeur
Autoriser les procédures	<input checked="" type="checkbox"/>
Autoriser les sous-programmes	<input checked="" type="checkbox"/>
Autoriser les commentaires imbriqués	<input type="checkbox"/>
Autoriser les affectations en cascade [a:=b:=c:] (ST /LD)	<input type="checkbox"/>
Autoriser les paramètres vides dans les appels informels (ST /IL)	<input checked="" type="checkbox"/>
Utilisation d'expressions ST (LD/FBD)	<input checked="" type="checkbox"/>
Maintenir les liens de sortie sur les EF désactivés (EN=0)	<input type="checkbox"/>
Afficher les commentaires complets d'élément de structure	<input type="checkbox"/>

Options du projet

- Options du projet
 - Général
 - Gestion des messages de génération
 - Paramètres de génération
 - Sauvegarde automatique lors du transf.
 - Données intégrées de l'automate
 - Diagnostics automatique
 - Variables
 - Programme
 - Langages
 - Commun
 - FBD
 - LD
 - Vue mixte**

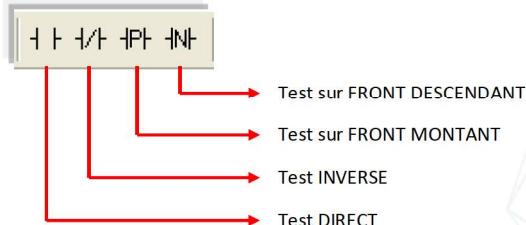
Propriété

Propriété	Valeur
Hauteur des lignes de commentaire	1
Hauteur des lignes de symbole	1
Hauteur des lignes d'adresse	1

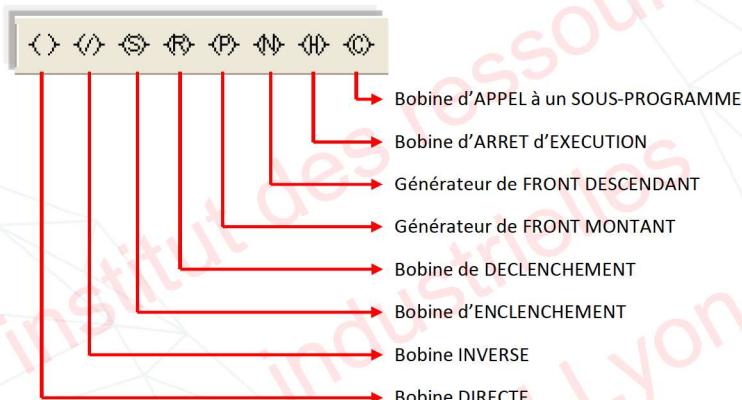
Unity Pro: les langages de programmation FBD, LD, IL, ST

2. LE LANGAGE A CONTACTS (LD)

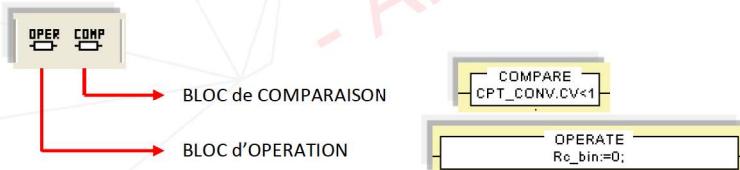
□ LES ELEMENTS DE TEST



□ LES ELEMENTS D'ACTIONS BOOLEENNES



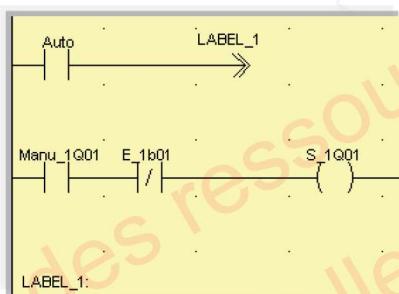
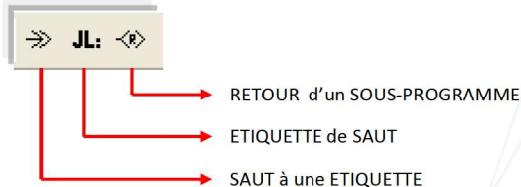
□ Les EXPRESSIONS NUMERIQUES



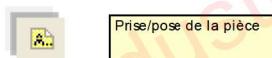
Unity Pro: les langages de programmation FBD, LD, IL, ST

2. LE LANGAGE A CONTACTS (LD)

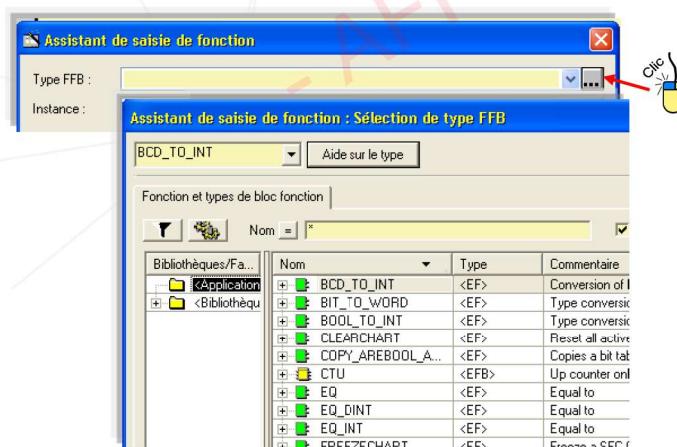
□ LES SAUTS DANS UNE SECTION



□ LES COMMENTAIRES



□ LA SAISIE DES FONCTIONS

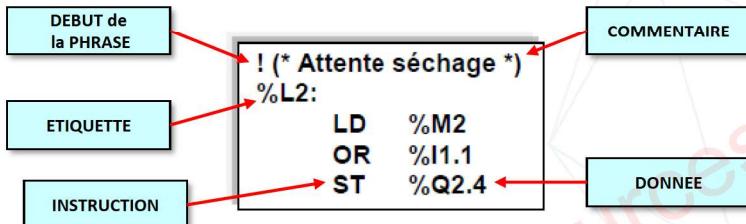


Unity Pro: les langages de programmation FBD, LD, IL, ST

3. LE LANGAGE LISTE D'INSTRUCTIONS (IL)

□ PRESENTATION

- ✓ Le langage IL se programme sous forme de **phrases** de 128 instructions maximum, comportant un **commentaire** et une **étiquette**.



- ✓ Les commentaires sont précédés de (*) et se terminent par (*).
- ✓ Une phrase peut s'écrire en une seule ligne:

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

□ LA CREATION D'UNE SECTION « IL »



- ✓ Nommer la section (ex: POMPE_2)
- ✓ Sélectionner le langage IL
- ✓ Valider la section:

OK

Unity Pro: les langages de programmation FBD, LD, IL, ST

3. LE LANGAGE LISTE D'INSTRUCTIONS (IL)

LES INSTRUCTIONS BOOLEENNES

LD	Charge un résultat booléen (commence une phrase)
LDN	Charge le complément
LDF	Charge le front montant
LDR	Charge le front descendant
AND	Et
OR	Or
ANDN	Et pas
ORN	Ou pas
ANDF	Et front montant
ORF	Ou front montant
ANDR	Et front descendant
ORR	Ou front descendant
XOR	Ou exclusif
XORN	Ou pas exclusif
XORF	Ou front montant exclusif
XORR	Ou front descendant exclusif
ST	Range le résultat
STN	Range le complément
MPS	Stock (empile) le résultat booléen (pour une utilisation ultérieure)
MPP	Destock (dépile) un résultat
MRD	Lit la dernière valeur stockée sans la dépiler
TRUE	toujours Vrai
FALSE	toujours Faux

LES INSTRUCTIONS ARITHMETIQUES

ADD	Addition
SUB	Soustraction
MUL	Multiplication
DIV	Division
MOD	Reste d'une division

LES INSTRUCTIONS DE COMPARAISON

GT	Supérieur strictement (>)
GE	Supérieur ou égal (>=)
EQ	Egal (=)
NE	Different (<>)
LT	Inférieur strictement (<)
LE	Inférieur ou égal (<=)

Unity Pro: les langages de programmation FBD, LD, IL, ST

3. LE LANGAGE LISTE D'INSTRUCTIONS (IL)

□ LES INSTRUCTIONS D'APPEL

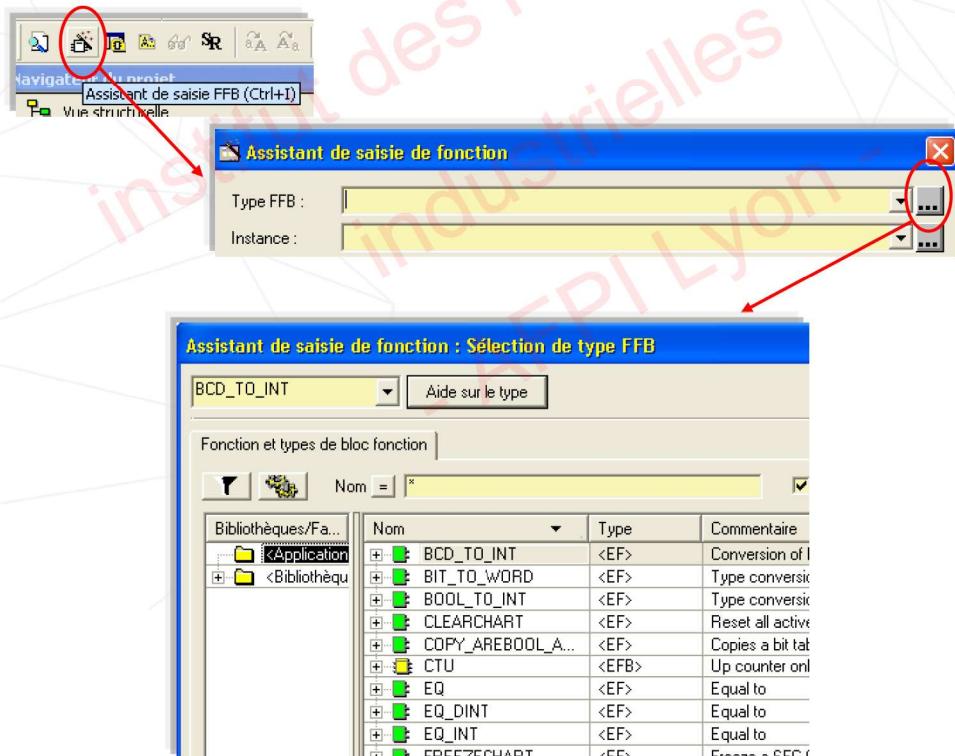
CALL	Appel d'un bloc fonction, d'un DFB ou d'un sous-programme (SR)
NOM_DE_FONCTION	Exécution d'une fonction
NOM_DE_PROCEDURE	Exécution d'une procédure

□ LES INSTRUCTIONS DE SAUT

JMP	Saut à une étiquette
RET	Retour d'un DFB ou d'un sous-programme (SR)

□ LA SAISIE DES FONCTIONS

✓ La saisie des fonctions est réalisé depuis l'assistant de saisie:

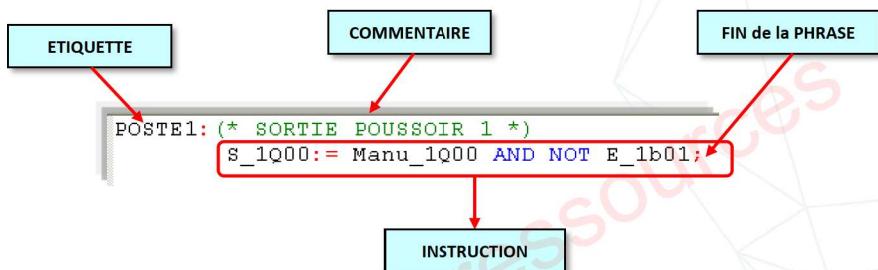


Unity Pro: les langages de programmation FBD, LD, IL, ST

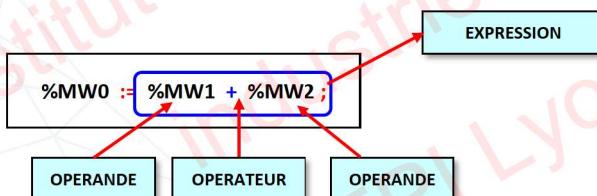
4. LE LANGAGE LITTERAL STRUCTURE (ST)

□ PRESENTATION

- ✓ Le langage littéral structuré ST se programme sous forme de **phrases ou instructions** d'une **longueur maximale de 300 caractères**, comportant éventuellement un **commentaire** et une **étiquette**.
- ✓ Une instruction se termine toujours par un **point virgule**:



- ✓ Les **expressions** sont des constructions comprenant **opérateurs** et **opérandes** qui livrent un résultat lors de leur exécution:



- ✓ Plusieurs **instructions** peuvent être **programmées sur la même ligne** en étant **séparées** par un **point virgule**:

```
(* DESACTIVATION DES SORTIES *)
S_2Q00:=0; S_2Q01:=0; S_9Q00:=0;
```

Unity Pro: les langages de programmation FBD, LD, IL, ST

4. LE LANGAGE LITTERAL STRUCTURE (ST)

□ LA CREATION D'UNE SECTION « ST »

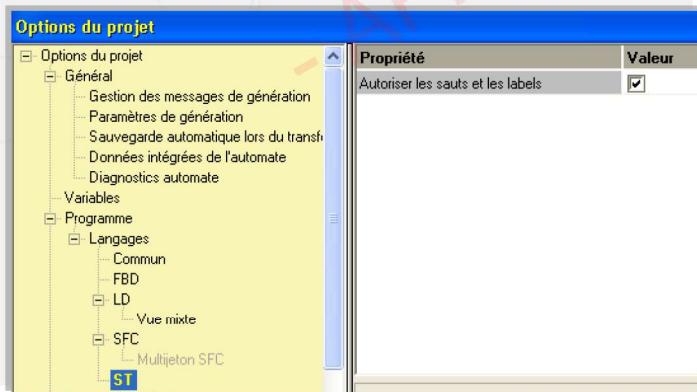


- ✓ Nommer la section (ex: POMPE_3)
- ✓ Sélectionner le langage ST
- ✓ Valider la section:

□ LE PARAMETRAGE DES OPTIONS DU PROJET

Outils Options du projet...

- ✓ Paramétrage du langage ST:



Unity Pro: les langages de programmation FBD, LD, IL, ST

4. LE LANGAGE LITERAL STRUCTURE (ST)

□ LES OPERATEURS DE BASE

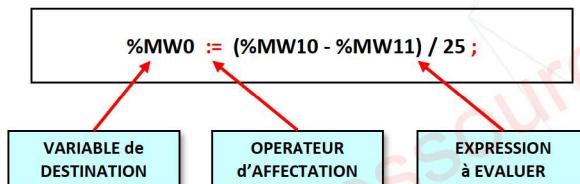
OPERATEURS	SIGNIFICATION	EXEMPLES
OPERATEURS ARITHMETIQUES		
()	Parenthèses	%MW0 := (%MW1 - %MW2) /12;
-	Négation	%MW0 := - %MW1;
NOT	Complément	%MW0 := NOT %MW1;
**	Elévation à la puissance	%MW0 := %MW1 ** 4;
*	Multiplication	%MW0 := %MW1 * %MW2;
/	Division	%MW0 := %MW1 / %MW2;
MOD	Reste de la division	%MW0 := %MW1 MOD 2;
+	Addition	%MW0 := %MW1 + 5;
-	Soustraction	%MW0 := %MW1 - 5;
COMPARATEURS		
<	« Inférieur strictement à »	%M4 := %MW0 <= %MW1; %M5 := %MW1 <> %MW2!
>	« Supérieur strictement à »	
<=	« Inférieur ou égal à »	
>=	« Supérieur ou égal à »	
=	« Egal à »	
<>	« Différent de »	
OPERATEURS LOGIQUES		
NOT	Complément	%MW0 := NOT %MW1; %MW10 := %MW1 AND %MW2;
AND	ET	
OR	OU	
XOR	OU exclusif	
RE	Front montant d'une variable de type « Ebool »	RE (%IO.1.1)
FE	Front descendant d'une variable de type « Ebool »	FE (%IO.1.1)

Unity Pro: les langages de programmation FBD, LD, IL, ST

4. LE LANGAGE LITTERAL STRUCTURE (ST)

□ LES INSTRUCTIONS D'AFFECTATION

- ✓ L'affectation remplace la valeur courante d'une variable par le résultat de l'évaluation d'une expression.
- ✓ Une affectation est composée d'une **variable de destination à gauche**, suivie de l'opérateur d'affectation := puis de l'expression à évaluer.
- ✓ Les **variables de destination et de l'expression** doivent être du **même type** de données.
- ✓ **Exemple:** affectation du résultat d'un calcul à une variable



□ L'INSTRUCTION « APPEL D'UN SOUS-PROGRAMME »

- ✓ L'appel d'un sous-programme comprend le nom de la section du sous-programme suivi d'une liste de paramètres vide.
- ✓ Le sous-programme à appeler doit se trouver dans la même tâche que la section ST appelante.
- ✓ Il est possible d'appeler des sous-programmes au sein de sous-programmes.

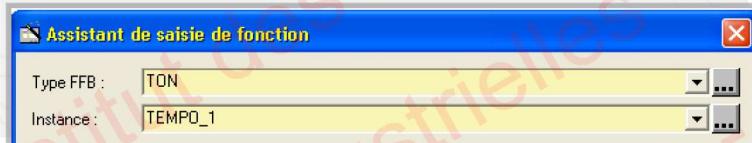


Unity Pro: les langages de programmation FBD, LD, IL, ST

4. LE LANGAGE LITTERAL STRUCTURE (ST)

❑ L'INSTRUCTION « APPEL D'UNE FONCTION »

- ✓ Les fonctions élémentaires sont disponibles sous forme de bibliothèques.
- ✓ La logique des fonctions est créée dans le langage de programmation C et ne peut pas être modifiée dans l'éditeur ST.



➤ Compléter la zone de saisie puis valider:

Nom	Type	N°	Commentaire	Zone de saisie
IN	BOOL	1	Start delay	%I0.2.0
PT	TIME	2	Preset delay time	T#5s
Q	BOOL	1	Delayed output	%Q0.3.4
ET	TIME	2	Internal time	VAL_TEMPS

OK

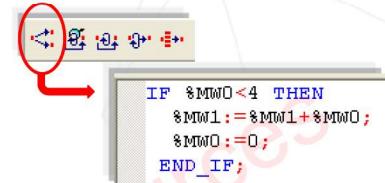
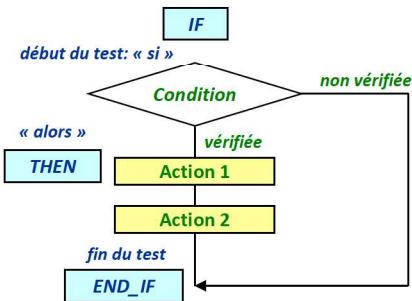
```
TEMPO_1 (IN := %I0.2.0,  
PT := T#5s,  
Q => %Q0.3.4,  
ET => VAL_TEMPS);
```

Unity Pro: les langages de programmation FBD, LD, IL, ST

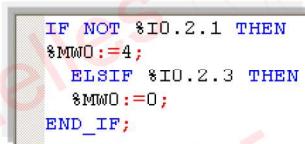
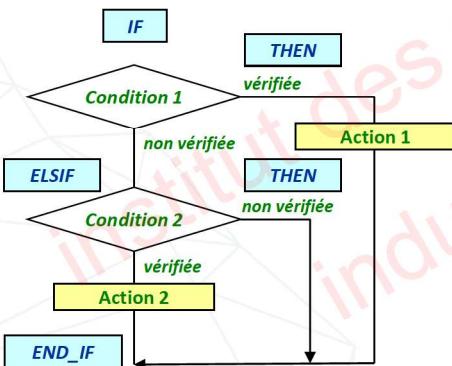
4. LE LANGAGE LITTERAL STRUCTURE (ST)

□ LA STRUCTURE CONDITIONNELLE « IF.....THEN.....END_IF »

- ✓ Cette structure de contrôle réalise une ou plusieurs actions si la condition est vraie.

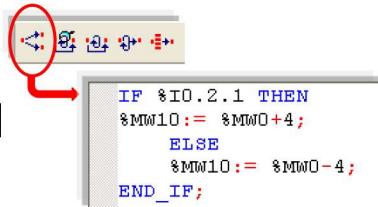
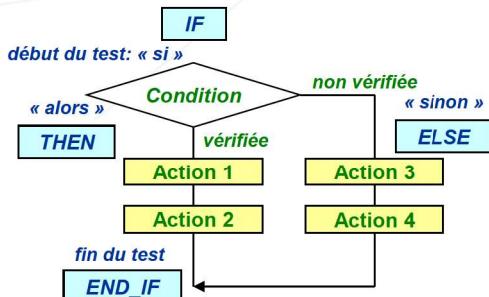


- ✓ Dans sa forme générale les conditions peuvent être multiples.



□ LA STRUCTURE CONDITIONNELLE « IF...THEN...ELSE...END_IF »

- ✓ Cette structure de contrôle réalise:
 - une ou plusieurs actions si la condition est vraie,
 - une ou plusieurs actions si la condition est fausse,



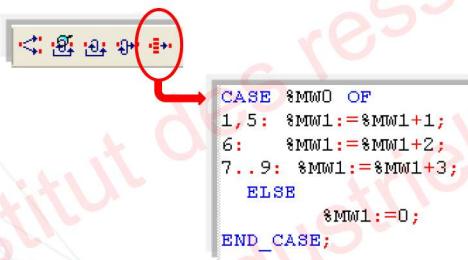
Unity Pro: les langages de programmation FBD, LD, IL, ST

4. LE LANGAGE LITTERAL STRUCTURE (ST)

□ LA STRUCTURE CONDITIONNELLE « CASE...OF...ELSE...END_CASE »

- ✓ L'instruction **CASE** est composée d'une expression de type données INT (le "sélecteur") et d'une liste de groupes d'instructions.
- ✓ Chaque groupe porte un repère composé d'un ou de plusieurs entiers ou de plages de valeurs entières.
- ✓ Le premier groupe d'instructions dont le repère contient la valeur calculée du sélecteur est exécuté. Sinon aucune des instructions n'est exécutée;
- ✓ L'instruction **OF** caractérise le début des repères.
- ✓ A l'intérieur d'une instruction **CASE**, on peut définir une instruction **ELSE** dont les instructions seront exécutées si aucun repère ne contient la valeur du sélecteur.
- ✓ L'instruction **END_CASE** marque la fin de l'instruction (des instructions).

✓ Exemple:



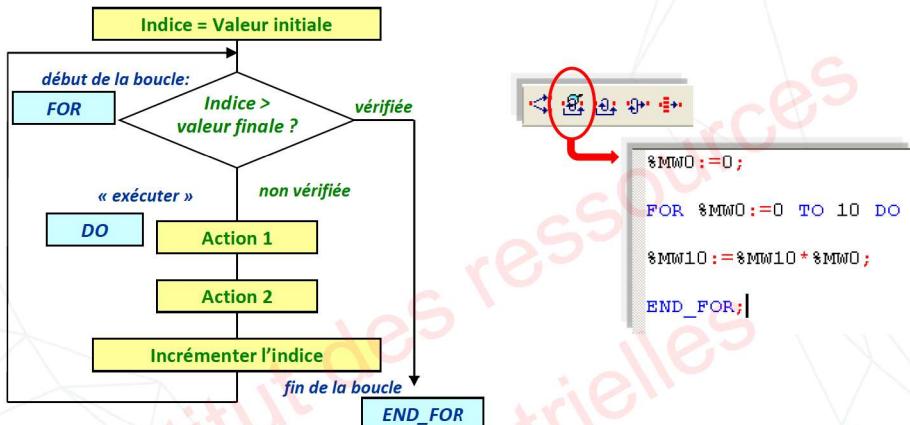
- Si %MW0 est égal à 1 ou à 5 alors %MW1 est incrémenté de 1.
- Si %MW0 est égal à 6 alors %MW1 est incrémenté de 2.
- Si %MW0 est compris entre 7 et 9 alors %MW1 est incrémenté de 3.
- Pour toute autre valeur de %MW0, %MW1 est mis à 0.

Unity Pro: les langages de programmation FBD, LD, IL, ST

4. LE LANGAGE LITTERAL STRUCTURE (ST)

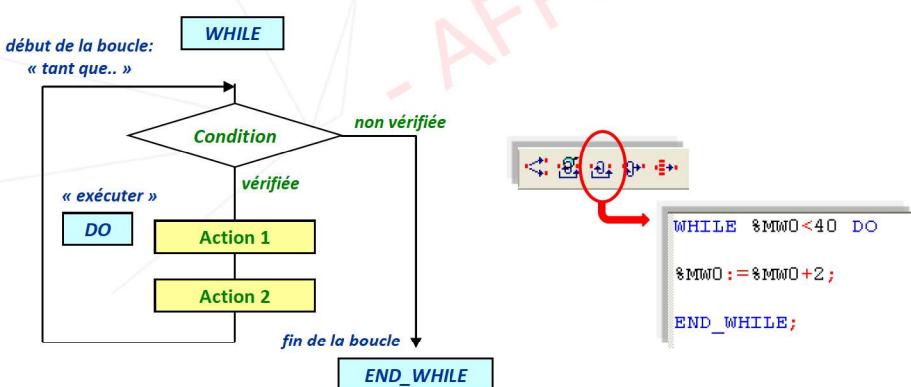
□ LA STRUCTURE REPETITIVE « FOR...TO...DO...END_FOR »

- ✓ Cette structure de contrôle réalise un traitement un certain nombre de fois en incrémentant un indice à chaque boucle.
- ✓ L'incrémentation de l'indice est effectuée automatiquement.
- ✓ Plusieurs structures de contrôle FOR peuvent être imbriquées.



□ LA STRUCTURE ITERATIVE CONDITIONNELLE « WHILE...DO...END WHILE »

- ✓ Cette structure de contrôle réalise une ou plusieurs actions TANT QUE la condition est vraie.
- ✓ Plusieurs structures de contrôle WHILE peuvent être imbriquées.

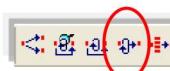
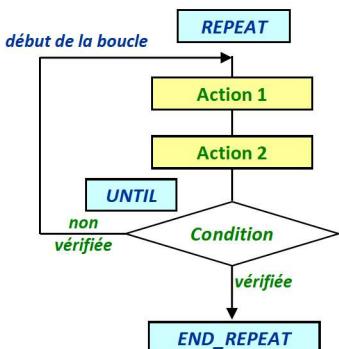


Unity Pro: les langages de programmation FBD, LD, IL, ST

4. LE LANGAGE LITTERAL STRUCTURE (ST)

□ LA STRUCTURE ITERATIVE CONDITIONNELLE « REPEAT...UNTIL...END_REPEAT »

- ✓ L'instruction REPEAT provoque la **répétition d'une chaîne d'instructions** (au moins une fois) jusqu'à ce que la condition booléenne correspondante soit 1 (vraie).
- ✓ L'instruction UNTIL marque la condition de fin.



```
%MW0 := 0;  
  
REPEAT  
    %MW0 := %MW0 + 2;  
  
UNTIL %MW0 >= 100  
  
END_REPEAT;
```

□ L'INSTRUCTION RECURRENTE « EXIT »

- ✓ L'instruction EXIT est utilisée pour **terminer les instructions récurrentes** (FOR, WHILE, REPEAT) avant que la condition de fin ne soit atteinte.
- ✓ Si l'instruction EXIT se trouve dans une occurrence imbriquée, la boucle la plus interne (dans laquelle se trouve EXIT) est abandonnée. Et l'instruction suivante qui est exécutée est la première instruction après la fin de la boucle (END_FOR, END WHILE ou END_REPEAT).

✓ Exemple:

```
FOR %MW0 := 0 TO 10 DO  
  
    %MW1 := %MW1 + %MW0;  
  
    IF %IO.2.7 THEN EXIT;  
  
    END_IF;  
  
END_FOR;
```

- L'exécution de la boucle est terminée si l'entrée «%IO.2.7 est à l'état 1.