

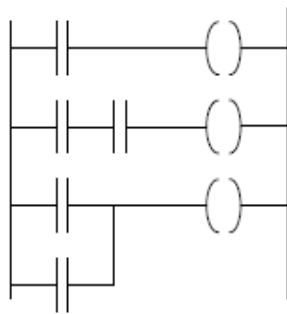
SEMAINE 4

LA PROGRAMMATION DES AUTOMATES

FICHE 21 : LES LANGAGES DE PROGRAMMATION

LE FBD (Function Block Diagram)

Ladder



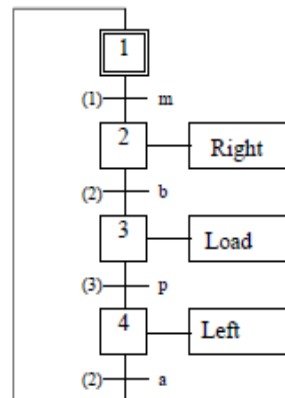
Structured Text (ST)

```
If %I1.0 THEN
  %Q2.1 := TRUE
ELSE
  %Q2.2 := FALSE
END_IF
```

Instruction List (IL)

LD	%M12
AND	%I1.0
ANDN	%I1.1
OR	%M10
ST	%Q2.0

Grafcet



Automation & Sense



Objectifs :

Après la consultation de cette fiche, vous :

- Pourrez décrypter un programme conçu en langage FBD (Function Block Diagram)
- Pourrez concevoir un programme en langage FBD



SOMMAIRE

- I) Les opérateurs logiques
- II) Les fonctions de comparaison
- III) Les fonctions mémoire
- IV) Les timers IEC
- V) Les compteurs IEC
- VI) La fonction MOVE
- VII) Les fonctions arithmétiques
- VIII) Les fonctions de conversion
- IX) Les fonctions de décalage

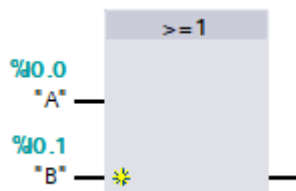


I) Les opérateurs logiques

Le langage FBD (Function Block Diagram) est un langage graphique qui utilise les portes logiques (OR, AND, OU Exclusif etc..) que l'on a pu voir dans les précédentes fiches.

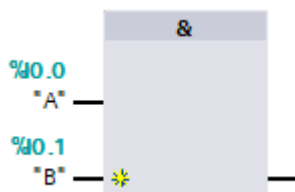
a) La fonction OR

Elle représente le « OU logique ». Sa sortie est à 1 si au moins une de ses entrées est égale à 1.



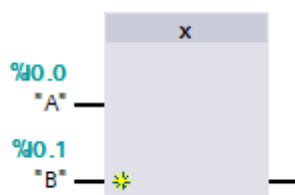
b) La fonction AND

Elle représente le « ET logique ». Sa sortie est à 1 si toutes ses entrées sont simultanément égales à 1.



c) Le OU Exclusif

Sa sortie est à 1 si ses entrées sont différentes.

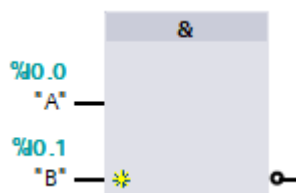




d) Inversion du résultat logique RLO d'une fonction logique

En FBD, on a la possibilité d'inverser le résultat logique d'une fonction logique en insérant le symbole de négation à sa sortie (petit rond).

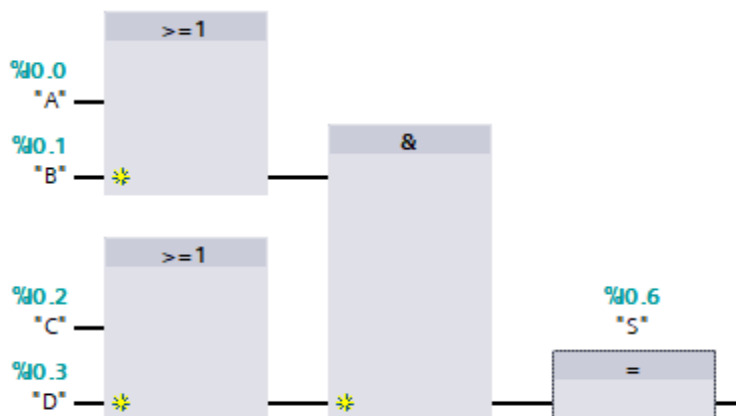
Par exemple, le symbole logique AND, inversé, donne le symbole ci-dessous :



Cela correspond à la fonction NAND (Non-ET).

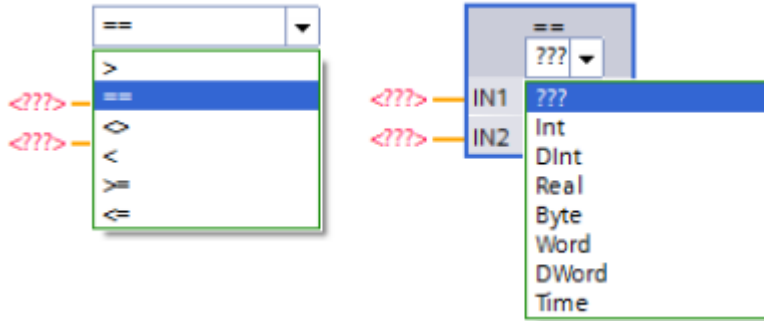
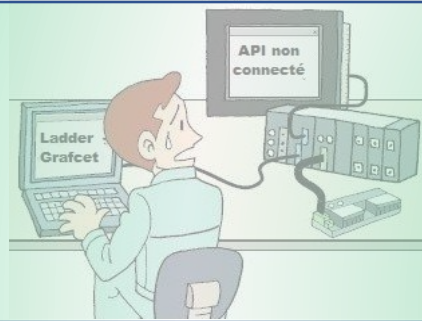
e) Exemple de programme utilisant plusieurs fonctions logiques

Sur le schéma ci-dessous, le RLO en sortie de la fonction logique AND est affecté à la sortie S via le symbole d'affectation (dernier bloc).



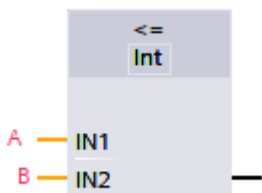
II) Les fonctions de comparaison

Elles permettent de comparer deux valeurs numériques. Leur sortie est une valeur booléenne. Pour insérer une fonction de comparaison sur TIA Portal, on spécifie d'abord l'opérateur de comparaison puis le type de données à comparer comme on peut le voir sur l'image ci-dessous.



Exemple :

Le bloc de fonction ci-dessous permet de comparer si l'entier A est inférieur ou égale à l'entier B.



III) Les fonctions mémoires

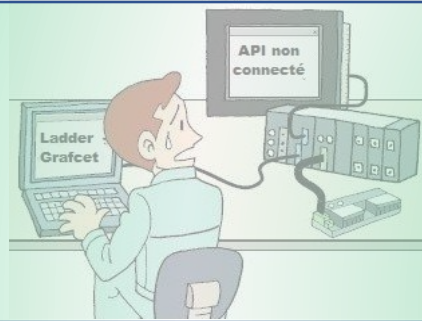
a) La fonction SET

Elle permet de mettre à 1 une variable si elle reçoit comme entrée un RLO égale à 1. Ici la variable S sera mise à 1, si le RLO à la sortie de la porte logique AND est à 1.



b) La fonction RESET

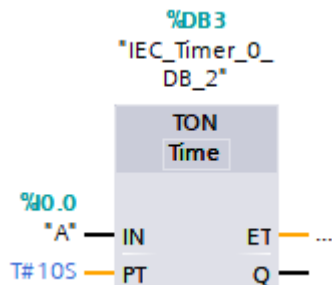
Elle permet de mettre à 0 une variable si elle reçoit comme entrée un RLO égale à 1. Ici la variable S sera mise à 0 si le RLO à la sortie de la porte logique AND est à 1.



IV) Les timers IEC

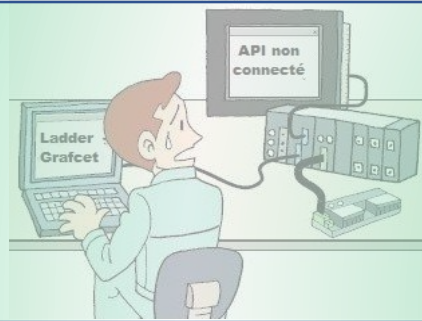
Les timers IEC sont des timers normalisés définis par l'IEC (commission internationale d'électrotechnique). Par opposition aux timers Simatic, ils font appel à des blocs systèmes. Il existe trois types de timers : **TON, TOFF et TP**. Chaque timer génère un DB d'instance.

a) Le TON



Name	Declaration	Data type	Description
IN	INPUT	BOOL	Start input
PT	INPUT	TIME	Defined duration
Q	OUTPUT	BOOL	Timer status
ET	OUTPUT	TIME	Elapsed time

Le timer TON démarre si l'état du signal de l'entrée IN passe de "0" à "1". Il expire quand la durée programmée à l'entrée PT est atteinte.

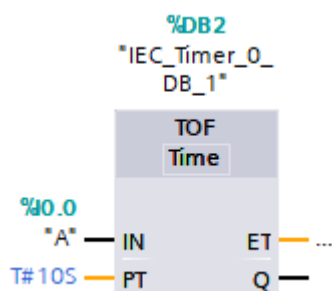


Sa sortie Q passe à "1" si la durée programmée est atteinte et que l'entrée IN est toujours à "1"

Le timer est reseté si le signal de l'entrée IN passe de « 1 » à « 0 » avant que la durée programmée ne soit terminée.

La sortie ET contient le temps écoulé. Il démarre à T#0s et se termine à PT.

b) Le TOF

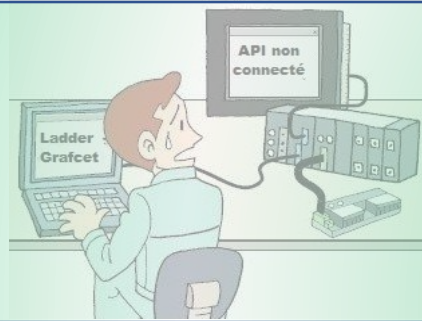


Name	Declaration	Data type	Description
IN	INPUT	BOOL	Start input
PT	INPUT	TIME	Defined duration
Q	OUTPUT	BOOL	Timer status
ET	OUTPUT	TIME	Elapsed time

Sa sortie Q passe à 1 quand l'état du signal de l'entrée IN passe de 0 à 1. Si le signal de l'entrée IN repasse à 0, la temporisation démarre jusqu'à atteindre la durée programmée PT. La sortie Q reste à 1 tant que la temporisation est en cours. La sortie Q est réinitialisée quand la temporisation expire.

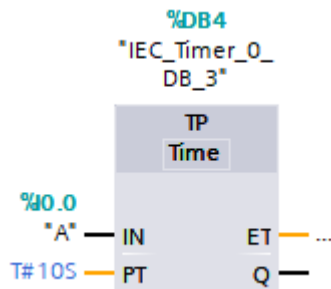
Si le signal de l'entrée IN repasse à 1 avant que la temporisation n'expire, la temporisation en cours sera réinitialisée et la sortie Q restera à 1.

La sortie ET contient le temps écoulé. Il démarre à T#0s et se termine à PT.



c) Le TP

La fonction TP permet de réduire ou d'étendre un signal d'entrée à la durée programmée.



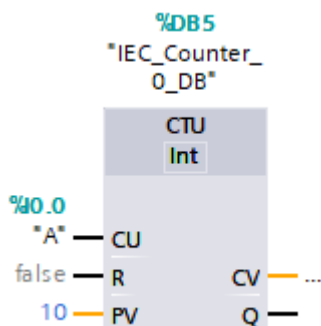
La temporisation démarre si l'état du signal de l'entrée IN passe de 0 à 1. Elle fonctionne pendant la durée programmée à l'entrée PT, indépendamment des variations de l'entrée IN. La sortie Q sera à 1 tant que la temporisation est en cours.

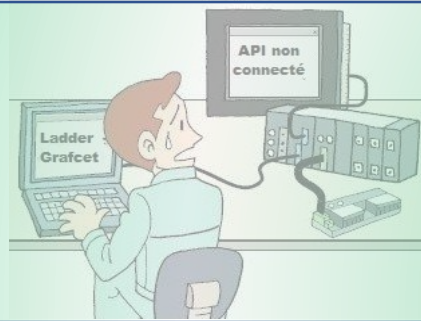
La sortie ET contient le temps écoulé. Il démarre à T#0s et se termine à PT.

V) Les compteurs IEC

Comme pour les timers, il existe des compteurs Simatic et des compteurs IEC. On peut utiliser les compteurs pour faire du comptage ou du décomptage. Il existe trois types de compteurs : le **CTU**, le **CTD** et le **CTUD**. Les fonctions de comptage font appel à des blocs systèmes, ainsi ils génèrent des DB d'instance.

a) La fonction CTU (comptage)



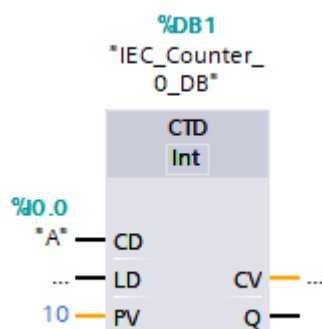


Name	Declaration	Data type	Description
CU	INPUT	BOOL	Count up input
R	INPUT	BOOL	Reset input
PV	INPUT	INT	Specified count value
Q	OUTPUT	BOOL	Counter status
CV	OUTPUT	INT	Actual count value

Si l'état du signal de l'entrée CU passe de 0 à 1, la valeur courante du compteur est incrémentée de 1 puis stockée dans CV. Si la valeur courante du compteur dépasse la limite définie au niveau de PV, l'incrémentacion est arrêtée.

Le compteur est réinitialisé si l'état du signal de l'entrée R est à 1. Quand l'état du signal de l'entrée R est à 1, le fait de mettre CU à 1 n'aura aucun effet.

b) La fonction CTD (décomptage)



Name	Declaration	Data type	Description
CD	INPUT	BOOL	Count down input
LD	INPUT	BOOL	Load input
PV	INPUT	INT	Specified count value
Q	OUTPUT	BOOL	Counter status
CV	OUTPUT	INT	Actual count value

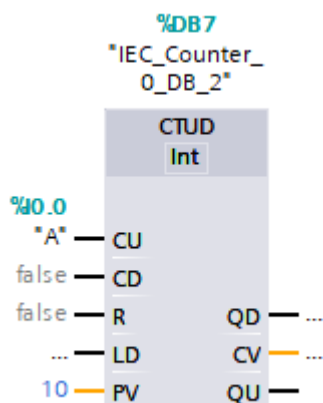


Si l'état du signal de l'entrée CD passe de 0 à 1, la valeur courante du décompteur est décrétementée de 1 puis stockée dans CV. Si la valeur courante du décompteur atteint la limite définie au niveau de PV, la décrémentation s'arrête.

La valeur de comptage CV sera définie à la valeur spécifiée PV si l'état du signal de l'entrée LD passe à 1. Quand l'état du signal de l'entrée LD est à 1, le fait de mettre CD à 1 n'aura aucun effet.

La sortie Q sera de 1 si la valeur courante de comptage CV est inférieure ou égale à 0.

c) La fonction CTUD (Compteur/décompteur)



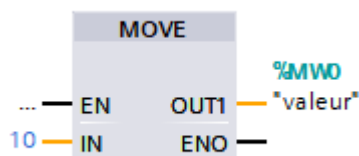
Name	Declaration	Data type	Description
CU	INPUT	BOOL	Count up input
CD	INPUT	BOOL	Count down input
R	INPUT	BOOL	Reset input
LD	INPUT	BOOL	Load input
PV	INPUT	INT	Specified count value
QU	OUTPUT	BOOL	Count up status
QD	OUTPUT	BOOL	Status down status
CV	OUTPUT	INT	Actual count value



Si l'état du signal de l'entrée CU passe de 0 à 1, le compteur est incrémenté de 1, puis sa valeur enregistrée dans CV. Si l'état du signal de l'entrée CD passe de 0 à 1, le décompteur est décrémenté de 1, puis sa valeur enregistrée dans CV. Si CU et CD passe tous les deux de 0 à 1, la valeur de CV ne sera pas modifiée.

VI) La fonction MOVE

La fonction Move permet de transférer la donnée qui se trouve sur son entrée IN vers OUT1

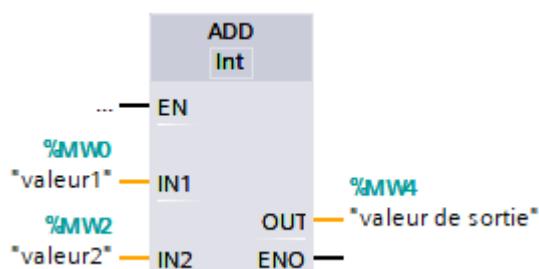


La fonction MOVE peut prendre en paramètre les principaux types de données (INT, DINT, REAL etc..). Cependant, le type Bool n'est pas accepté.

VII) Les fonctions arithmétiques

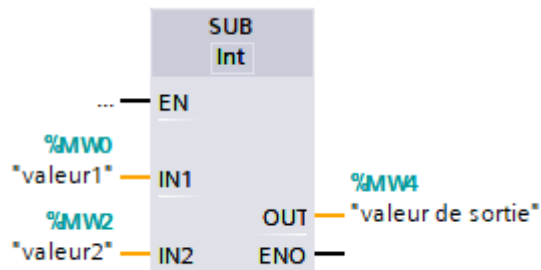
Les fonctions arithmétiques permettent d'effectuer des opérations arithmétiques. Il en existe principalement quatre : ADD, SUB, MUL et DIV.

La fonction ADD permet d'additionner deux variables qui peuvent être de type INT, DINT ou REAL.

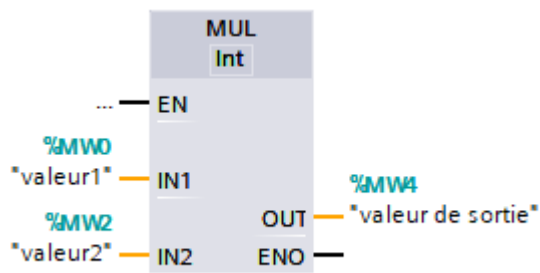




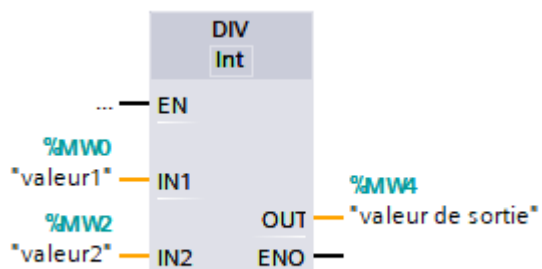
La fonction SUB permet de soustraire deux variables qui peuvent être de type INT,DINT,REAL.

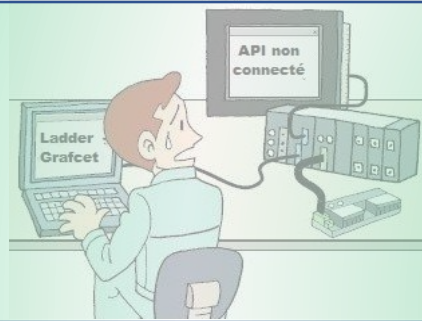


La fonction MUL permet de multiplier deux variables qui peuvent être de type INT,DINT ou REAL.



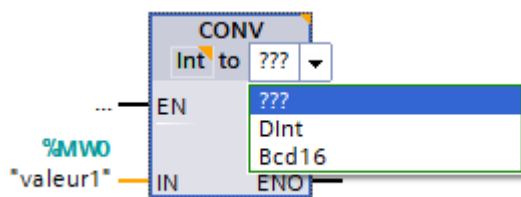
La fonction DIV permet de diviser deux variables qui peuvent être de type INT,DINT ou REAL.





VIII) Les fonctions de conversion

Elles permettent de convertir une variable en un autre type (INT,DINT,REAL,BCD etc..).Il suffit de spécifier le type de la variable à convertir et le type vers lequel cette variable doit être convertie. Voir ci-dessous.



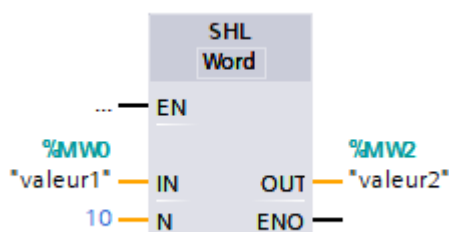
IX) Les fonctions de décalage

Les fonctions de décalage permettent de décaler bit par bit le contenu d'une variable vers la droite ou vers la gauche.

On distingue principalement la fonction de décalage vers la gauche **SHL** et la fonction de décalage vers la droite **SHR**. Avec ces deux fonctions on pourra décaler un WORD ou un DWORD

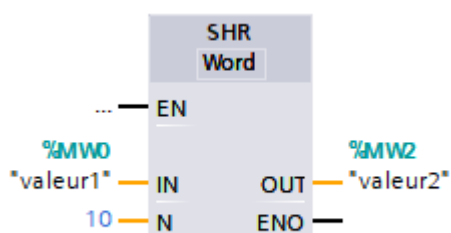
a) La fonction SHL

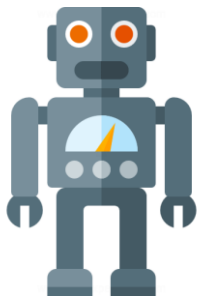
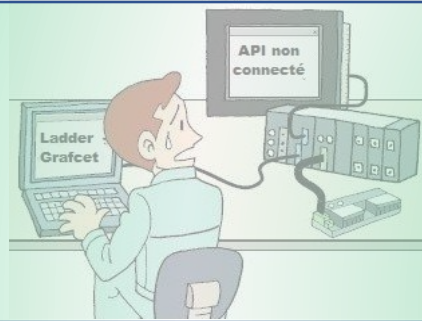
Elle permet de décaler une variable d'entrée de N bits vers la gauche.



b) La fonction SHR

Elle permet de décaler une variable d'entrée de N bits vers la droite





Dans cette fiche, vous avez pu découvrir le langage FBD

Vous pourrez désormais concevoir avec un peu de pratique des programmes en FBD sur TIA Portal ou sur Simatic Manager Step7

