

# Le Millenium 3 pour les nuls !!

## Phase 2 : Les blocs fonction

**Document rédigé par Pascal Bigot pour le site et le forum APPER**

### **Introduction :**

Le premier tutoriel vous a normalement permis de prendre en main les diverses manipulations de base de l'atelier M3, et il est maintenant le moment de comprendre à quoi sert et comment marche chaque bloc fonction afin de les utiliser au mieux dans votre programme.

Pour attaquer sereinement cette partie du tutoriel, il est impératif d'avoir acquis tout ce qui est expliqué dans la phase 1 (sauf partie raccourci clavier). Cette partie a uniquement pour but de décrire l'ensemble des fonctions à disposition dans l'atelier Millenium 3, afin que vous sachiez ce qui existe sans avoir à tester une par une chacune des fonctions, et que vous puissiez faire le bon choix de bloc selon vos besoins. Notons tout de même que le niveau global du document est un peu plus technique que dans la phase précédente.

Dans ce tutoriel, j'introduis également des explications spécifiques à la logique booléenne car c'est une notion qu'il faut impérativement maîtriser si l'on souhaite créer des programmes un minimum évolués (comme ceux d'une régulation solaire).

La dernière partie qui est un peu détaillée est la partie sur le SFC. Bien que le graphe SFC soit assez peu utile dans le cadre de la régulation solaire, il m'a semblé important d'en expliquer les grandes lignes pour ceux qui ne feraient pas que de la régulation avec leur M3.

# Table des matières

<b>Les blocs fonction en général</b> .....	<b>4</b>
<b>Les blocs IN</b> .....	<b>4</b>
<b>Les entrées TOR (DI et Filtered DI)</b> .....	<b>4</b>
<b>Les entrées ANALOGIQUES/NUMERIQUES (AI, Filtered AI, NUM IN, XA AI)</b> .....	<b>5</b>
Les entrées Analogiques.....	5
Les entrées Numériques.....	7
<b>Le bloc MARCHÉ et ARRÊT (CT ON et CT OFF)</b> .....	<b>8</b>
<b>Le bloc constante NUMERIQUE (constante)</b> .....	<b>8</b>
<b>Le bloc Timer 1s</b> .....	<b>8</b>
<b>Les boutons</b> .....	<b>8</b>
<b>Autres (ETE/HIVER)</b> .....	<b>9</b>
<b>Les blocs OUT</b> .....	<b>9</b>
<b>La sortie TOR (DO)</b> .....	<b>9</b>
<b>Le rétro éclairage</b> .....	<b>9</b>
<b>La sortie PWM</b> .....	<b>9</b>
<b>La sortie NUM OUT</b> .....	<b>10</b>
<b>La sortie XA ANA/PWM</b> .....	<b>10</b>
<b>Les blocs LOGIC et la logique booléenne</b> .....	<b>11</b>
<b>Le ET Logique (bloc AND)</b> .....	<b>11</b>
<b>Le OU Logique (bloc OR)</b> .....	<b>12</b>
<b>Le NON Logique (bloc NOT)</b> .....	<b>14</b>
<b>Le NOR (NON OU) et le NAND (NON ET)</b> .....	<b>16</b>
<b>Le OU EXCLUSIF (bloc XOR ... le shérif de l'espace)</b> .....	<b>18</b>
<b>La fonction BOOLEENNE !!</b> .....	<b>18</b>
<b>La fonction super BOOLEENNE !!</b> .....	<b>22</b>
<b>Les blocs FBD</b> .....	<b>24</b>
<b>Les TIMERS</b> .....	<b>24</b>
Timer AC (retardateur).....	24
Timer LI (clignoteur).....	26
Timer BW (impulsionnel) .....	29
Timer BH (temporisateur) .....	30
Timer TOT (Totalisateur).....	31
<b>Les Afficheurs</b> .....	<b>34</b>
<b>Les Fonctions Mathématiques</b> .....	<b>35</b>
Addition/Soustraction (ADD/SUB).....	35
Multiplication/Division (MUL/DIV).....	35
GAIN .....	36
<b>Les Compteurs</b> .....	<b>37</b>
Compteur à présélection (preset counter) .....	37
Compteur (counter) .....	38

Compteur horaire (H-meter).....	39
<b>Les Comparateurs.....</b>	<b>40</b>
Comparateur simple.....	40
Comparateur à fenêtre (compare in zone).....	40
Trigger de Schmitt.....	41
<b>Les liaisons série.....</b>	<b>43</b>
SL IN (entrées liaison série).....	43
SL OUT (sorties liaison série).....	44
A propos des SL IN/SL OUT.....	44
<b>Les « Mémoires ».....</b>	<b>44</b>
Archives.....	44
Min/Max.....	46
<b>Les Autres.....</b>	<b>47</b>
Bascule RS (Set/Reset).....	47
Télérupteur (BISTABLE).....	47
Multiplexeur (MUX).....	48
Arbre à Cam (CAM bloc).....	49
Status.....	50
Programmateur horaire (Time Prog).....	51
Convertisseur DEC=>BIN.....	54
Convertisseur BIN=>DEC.....	55
<b><i>Les blocs FBD_C.....</i></b>	<b><i>56</i></b>
<b>Généralités.....</b>	<b>56</b>
<b>Les plus utiles.....</b>	<b>56</b>
Super booléen.....	56
Timer A.....	57
Porte logique ET à deux entrées.....	58
Porte logique OU à deux entrées.....	58
<b><i>Les blocs SFC.....</i></b>	<b><i>59</i></b>
<b>L'étape initiale, et l'étape initiale ré-initialisable.....</b>	<b>59</b>
<b>L'étape simple.....</b>	<b>60</b>
<b>La divergence en OU et la convergence en OU.....</b>	<b>60</b>
<b>La divergence en ET et la convergence en ET.....</b>	<b>60</b>
<b>Un graphe SFC « simple ».....</b>	<b>60</b>

## Les blocs fonction en général

Un bloc fonction c'est un bloc que l'on pose sur la surface de câblage de l'atelier M3 et qui a un comportement spécifique. Les blocs sont regroupés selon différentes catégories de leur type.

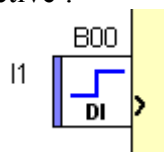
### Les blocs IN

Dans cette catégorie on trouve tous les blocs qui servent d'entrée dans vos programmes. Les entrées peuvent être tout ou rien (TOR) ou analogique (ANA).

#### Les entrées TOR (DI et Filtered DI)

Alors, notre automate à des entrées... et notre programme aussi, à commencer par les entrées TOUT ou RIEN (TOR) standards et filtrées.

Pour ce qui est de l'entrée TOR, rien de très spécial à raconter, puisque si l'entrée physique est active, la sortie de ce bloc sera également active !

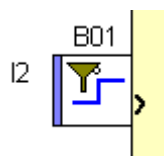


La seule petite chose spéciale de ce bloc, c'est que vous pouvez modifier l'icône qu'il affiche, que ce soit à partir d'une liste prédéfinie, ou à partir de vos propres images.

**AVERTISSEMENT :** Soyez intelligent lorsque vous voulez utiliser des images dans vos programmes M3, car ceci peut vite provoquer des gros ralentissements de l'atelier.

Prenons un exemple courant : vous avez pris une photo de votre installation pour que ce soit tout joli, et vous la placez dans votre programme. Votre photo issue de votre super appareil photo numérique entre 3 et 8 Méga Pixels (voir plus) selon son âge va vous faire des clichés qui pèsent plusieurs méga octets pour chacun. Si vous utilisez tel quel vos clichés dans l'atelier, c'est la brouette assurée car vous demandez à un atelier de programmation de traiter des grosses images alors qu'il n'est pas du tout fait pour ça. Il est donc recommandé de redimensionner vos images avant de les utiliser dans l'atelier. Par exemple pour l'image d'un bloc d'entrée, de sortie, ou d'une MACRO, le format idéal de votre photo doit être de 36 pixels par 36 pixels.

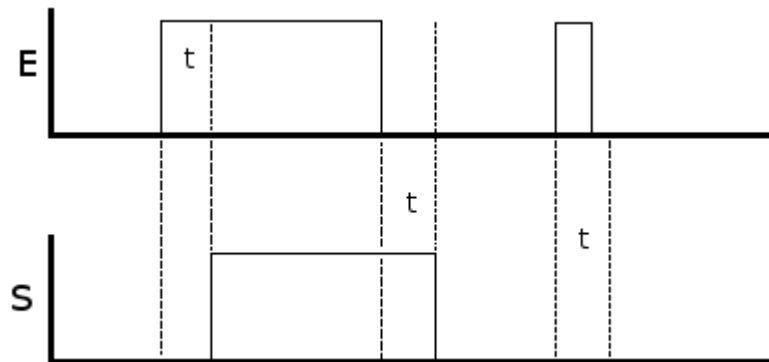
Pour ce qui est de la fonction TOR filtrée, celle-ci à un peu plus d'intelligence que la fonction TOR standard.



Cette fois ci, nous avons une entrée TOR qui ne passera à 1 que si l'entrée physique de l'automate a été active suffisamment longtemps, et repassera à 0 que si elle a été inactive assez longtemps, en fonction de la durée du filtrage.

Voici un chronogramme qui sert à expliquer ceci :

E représente l'entrée physique de l'automate, S représente la sortie bloc DI filtré, et t la période de filtrage.



Ainsi on remarquera que :

1 : l'état de la sortie de notre bloc est en décalage par rapport à l'entrée physique

2 : si l'état de l'entrée physique passe fugitivement à 1, sur un temps  $t$  inférieur au filtrage, la sortie du bloc restera à 0

3 : si l'entrée physique passe à 0 fugitivement sur un temps  $t$  inférieur au filtrage, et que la sortie du bloc était à 1 (cas non dessiné dans le chronogramme)

L'intérêt de cette fonction est donc de pouvoir filtrer d'éventuels sursauts de l'entrée dus à des perturbations électriques ou à des états transitoires accidentels, sans que cela ne se ressente dans le programme.

Bon il faut être honnête, cette fonction sert rarement dans nos installations, mais il est bon de savoir qu'elle existe (il y a peu de capteur tout ou rien dans nos installation, et ils ont peu de raison d'avoir des sursauts).

### **Les entrées ANALOGIQUES/NUMERIQUES (AI, Filtered AI, NUM IN, XA AI)**

Alors j'ai placé les entrées analogiques et numériques dans le même sous chapitre car pour moi elles ont toute la même utilité, elles nous permettent d'utiliser autre chose que des 1 et des 0 dans nos programme.

#### **Les entrées Analogiques**

Dans cette catégorie on va ranger les blocs AI, Filtered AI, et XA AI.

Le bloc **AI** est l'entrée analogique de base, il est possible de le placer sur les plots d'entrées capables d'accepter une entrée analogique (différent selon le modèle de M3 choisi).



Elle possède deux modes de paramétrage :

- le mode 0-10V
- le mode potentiomètre

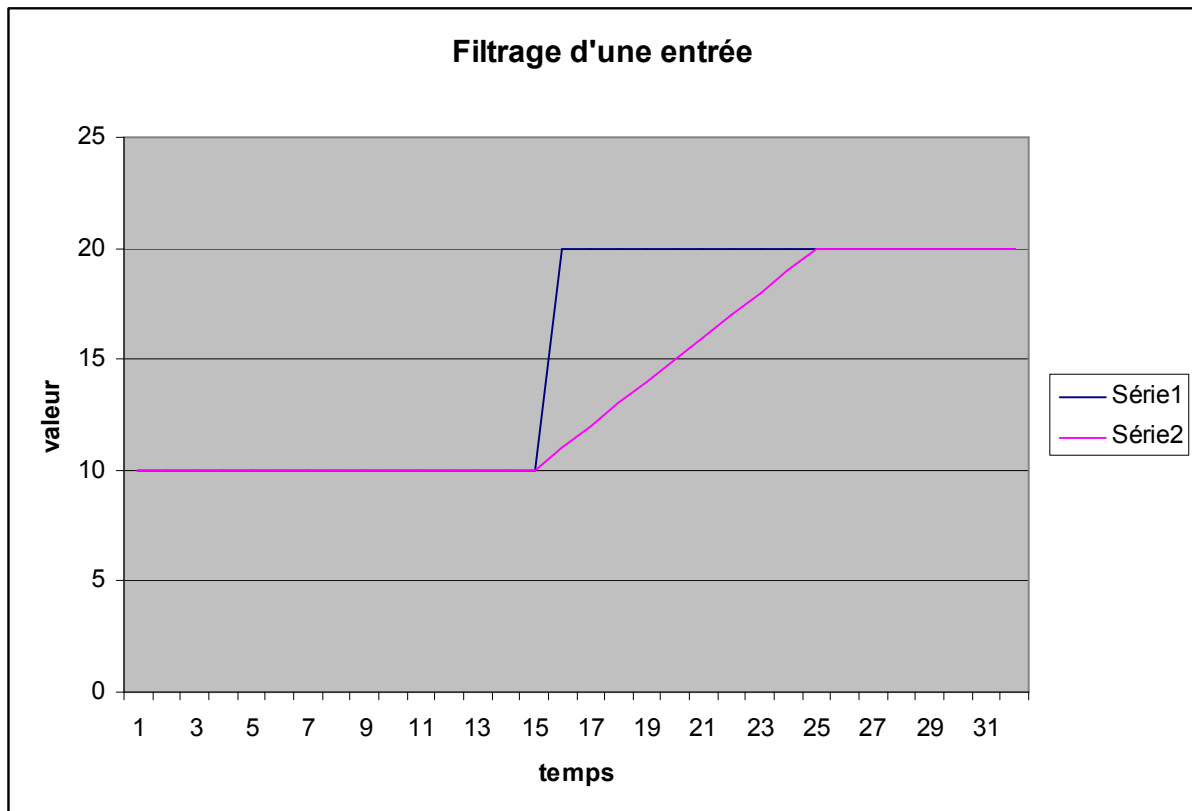
Le mode 0-10 volts offre une fausse résolution de 10 bits (valeur de 0 à 1023) sur la plage 0-10V. Alors pourquoi je parle de fausse résolution 10 bits? Eh bien parce que c'est le cas. En fait lorsqu'on est en mode 0-10V, la conversion tension d'entrée => valeur est effectuée logiciellement, et cela se traduit par une perte de résolution. D'après les tests faits par les membres du forum APPER, il semblerait que cette conversion fasse tomber la résolution en mode 0-10V à un équivalent de 8 bits (valeur de 0 à 255).

Le mode potentiomètre offre une vraie résolution de 10 bits (valeur de 0 à 1023) sur la plage 0-Tension d'alim (12V ou 24V selon les modèles de contrôleurs)

Le bloc **Filtered AI** possède les mêmes fonctions que l'entrée analogique de base, mais possède en plus une fonction de filtrage sur le temps lui permettant notamment de lisser les perturbations électriques qui peuvent perturber la mesure.



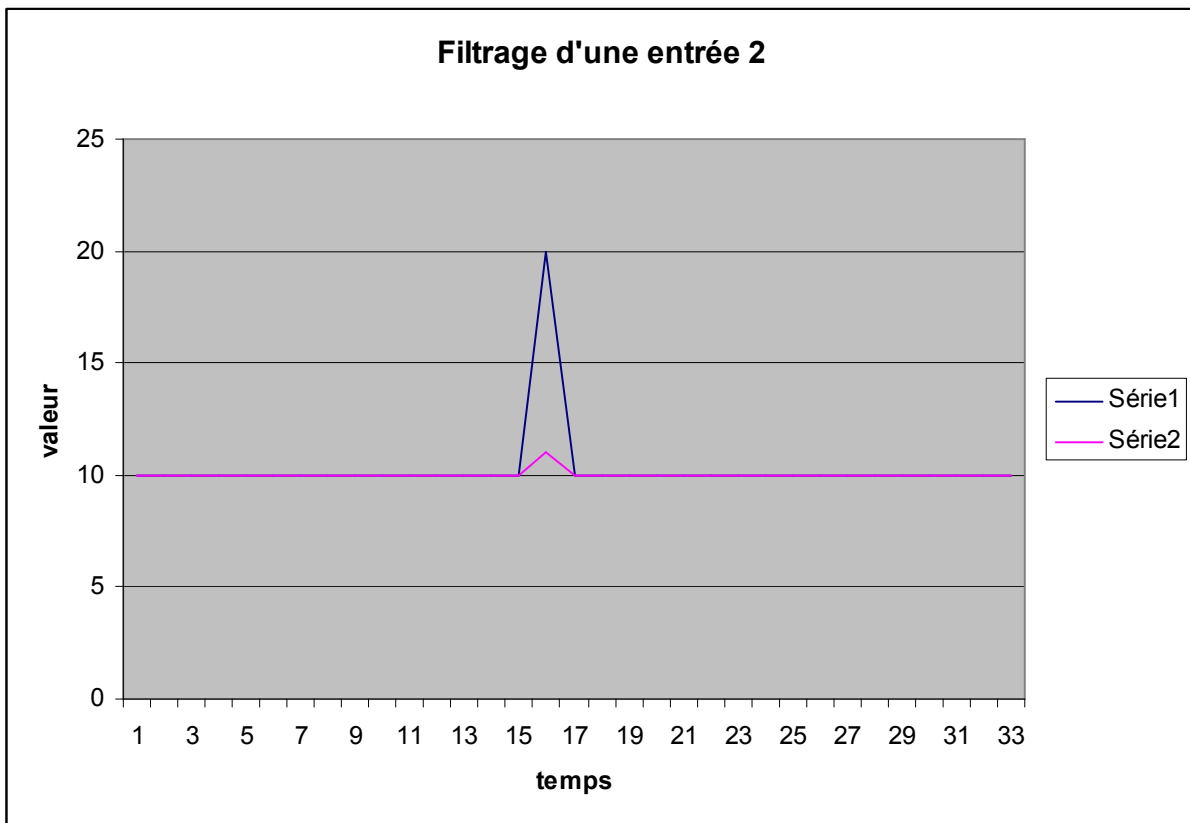
Imaginons un premier cas où l'entrée va subitement changer de valeur, nous obtiendrons le résultat suivant :



(En violet, la valeur de sortie de l'entrée analogique filtrée, et en bleu la valeur de l'entrée physique)

Dans ce cas, l'entrée analogique filtrée va permettre une variation progressive jusqu'à atteindre la valeur réelle de l'entrée.

Prenons maintenant le cas où l'entrée réelle va avoir un sursaut puis revenir à sa valeur initiale.

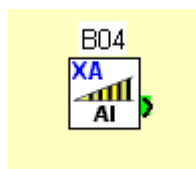


(En violet, la valeur de sortie de l'entrée analogique filtrée, et en bleu la valeur de l'entrée physique)

Dans ce cas, l'entrée analogique filtrée va permettre de limiter l'impact du pic dans le comportement du programme.

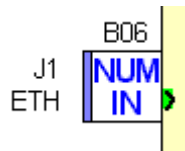
Bien entendu l'efficacité du filtrage va dépendre de son paramétrage. Dans le cas des installations solaires, une fréquence de coupure la plus petite possible sera tout à fait satisfaisante car ce n'est pas un système où les variations ont besoin d'être perçues rapidement. Un filtre avec une fréquence de coupure très faible lisse un passage de 0 à Vmax en 5 secondes environ, ce qui est tout à fait correcte vis-à-vis des vitesses de variations des températures pour le solaire.

Le bloc **XA AI**, est en fait un bloc spécial qui ne s'utilise qu'avec l'extension analogique XA 04 (et peut être aussi d'autres extensions analogiques que je ne connais pas). Il n'a aucun paramètre car c'est l'extension qui est paramétrée à la place du bloc.



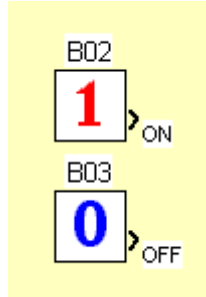
## Les entrées Numériques

Dans le M3 nous n'avons qu'un seul bloc que je considère comme étant une entrée numérique, c'est le NUM IN. Ce bloc ne vous sera utile que si vous utilisez des extensions de communication (XN03/XN06, XN05). C'est le seul bloc qui peut être posé sur les plots d'entrées ajoutés par ces extensions. **Ils n'ont pas de paramétrage et leur plage de valeur va de -32768 à +32767.**



### **Le bloc MARCHE et ARRET (CT ON et CT OFF)**

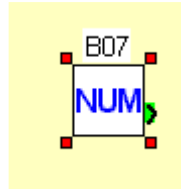
Pour ces blocs, rien de très extraordinaire à raconter, puisque leur sortie ne varie jamais. C'est soit ON, soit OFF en fonction du bloc choisi.



Seule petite chose à noter, il est possible de les forcer en simulation ou en monitoring. Dans tous les cas ces blocs vous seront rarement utiles en dehors de la mise au point de vos programmes.

### **Le bloc constante NUMERIQUE (constante)**

Alors en voila un bloc très utile, et qui sera un fidèle compagnon de vos longues heures de programmation, c'est un bloc très simple puisqu'il permet seulement de configurer une constante numérique. Eh oui, il faut des valeurs de référence dans les programmes (60° max pour le ballon... ou un peu plus, 0° pour le gel, etc)



Son paramétrage permet de configurer sa valeur entre -32768 à +32767.

### **Le bloc Timer 1s**

Voila encore un bloc très simple. Pas de paramétrage, une seule sortie TOR, il ne fait que changer d'état 2 fois par seconde (période complète de 1 seconde).



### **Les boutons**

Là encore aucun grand mystère, puisqu'il s'agit des entrées correspondants aux boutons de la face avant du M3.



La sortie de ces blocs est active lorsque le bouton correspondant de la face avant est appuyé. Les boutons sont principalement utilisés pour gérer des affichages ou des modifications de variables. Il est possible de poser plusieurs fois le même bouton dans la surface de câblage sans que cela ne pose problème.

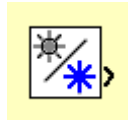


De manière générale, il y a quelques règles auxquelles il faut faire attention sur l'utilisation des boutons. Si vous utilisez la modification de variable via les blocs DISPLAY ou TEXT, il faut éviter d'utiliser le bouton Moins, plus OK ou Esc en même temps. Le bouton OK permet d'activer/valider la modification de variable, et le bouton Esc annule la modification effectuée alors que les boutons plus moins servent à modifier la valeur.

J'ai parlé ici des DISPLAY et des TEXT, car l'association boutons/afficheur est très courante. Ces blocs assez spéciaux ne sont pas abordés dans ce tutoriel car leur utilisation est détaillée dans un tutoriel dédié.

### **Autres (ETE/HIVER)**

La sortie du bloc été hiver s'active lorsque l'on est en hiver (la notion été hiver est géré dans les paramètres du M3, avec le changement d'heure automatique)

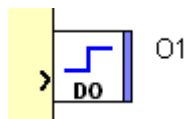


## **Les blocs OUT**

Dans cette catégorie, sans surprise on trouvera les blocs de sortie.

### **La sortie TOR (DO)**

C'est la sortie tout ou rien de base, et très certainement celle que vous utiliserez le plus étant donné que, à moins d'avoir une extension ou un modèle à sortie statique, les M3 n'ont que des sorties TOR. Il n'y a aucun paramétrage sur ce bloc, et comme vous vous en doutez (ou le savez déjà), lorsque l'entrée du bloc est activée, le relais du M3 correspondant au plot sur lequel il est posé est fermé.



### **Le rétro éclairage**

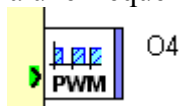
Ce bloc permet d'activer volontairement le rétro éclairage du LCD de la face avant. Juste un petit détail à propos d'une chose que j'ai trop souvent vue. Il est inutile d'activer volontairement le rétro éclairage lorsqu'on appuie sur un bouton, car le M3 le fait naturellement.

Une bonne utilisation du rétro éclairage est de l'allumer ou de le faire clignoter lorsqu'on souhaite attirer l'œil si une erreur ou une alerte est survenue dans le programme (à condition que le M3 soit à un endroit visible)



### **La sortie PWM**

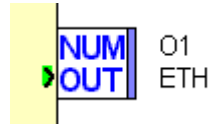
Cette sortie n'est utilisable qu'avec les M3 possédant des sorties statiques (ex : XD26S). Elle permet de sortir un signal PWM (Pulse Weight Modulation = Modulation de largeur d'impulsion) avec une résolution de 8 bits (256 valeurs) à une fréquence paramétrable dans le programme.



Je n'expliquerai pas ce qu'est le PWM, car si vous ne savez pas ce que c'est, c'est que vous n'en avez sans doute pas besoin. Pour les curieux je vous invite à aller visiter [cette page](#) sur Wikipedia.

### **La sortie NUM OUT**

C'est le seul bloc de sortie numérique. Ce bloc ne vous sera utile que si vous utilisez des extensions de communication (XN03/XN06, XN05). C'est le seul bloc qui peut être posé sur les plots de sortie ajoutés par ces extensions. Ils n'ont pas de paramétrables et leur plage de valeur va de -32768 à +32767.



### **La sortie XA ANA/PWM**

Le bloc **XA out**, est en fait un bloc spécial qui ne s'utilise qu'avec l'extension analogique XA 04 (et peut être aussi d'autres extensions analogiques que je ne connais pas). Il n'a aucun paramètre car c'est l'extension qui est paramétrée à la place du bloc.



## Les blocs LOGIC et la logique booléenne

Alors attention ce chapitre sous son air tranquille est loin d'être le plus simple à comprendre car la logique booléenne est un monde à part. Ici on va raisonner avec des 1 et des 0 (que l'on peut aussi traduire par « VRAI » et « FAUX »)

On va commencer doucement avec les fonctions ET (AND), OU (OR) et NON (NOT), qui sont les fonctions les plus simples mais également les plus importantes à connaître car ce sont les plus utilisées (les autres ont un intérêt pour l'optimisation ou la lisibilité ...pour ceux qui maîtrisent)

Ensuite, nous verrons les fonctions NOR (NON OU), et NAND (NON ET) qui n'apportent rien de très compliqué alors que la fonction XOR (OU Exclusif) est un peu plus spéciale.

En dernier, on verra un bloc qui se trouve dans l'onglet FBD mais qui à tout à fait sa place dans la partie logique, puisqu'il s'agit du bloc BOOLEEN.

### Le ET Logique (bloc AND)

Alors comme on est dans un tutoriel pour les nuls, je vais tenter de vulgariser car si je vous dis d'office  $1 \text{ ET } 1 = 1$ ...je vais en perdre plus d'un en route et ce n'est pas le but. Alors quand on parle de ET logique, on parle d'une association de deux éléments qui sont la condition à la réalisation d'un troisième.

Prenons le cas d'une installation solaire simple dont on souhaite piloter le circulateur :

Pour que le circulateur fonctionne il faut au minimum que :

- la T° du capteur soit supérieur à celle de l'eau contenue dans le ballon (on notera cette condition TCaptOK)
- que la T° de l'eau dans le ballon n'ait pas atteint la valeur maximum (mettons 60°) (on notera cette condition TBallonNonMax)

On aurait une « équation » du type :

TCaptOK ET TBallonNonMax = circulation en marche

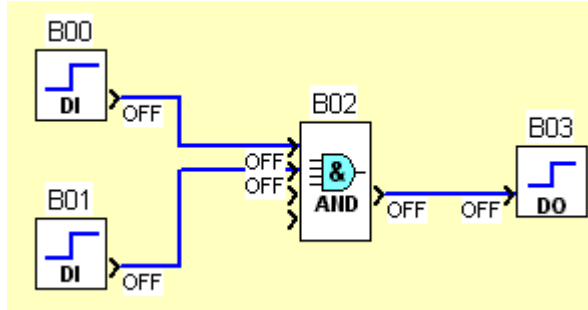
Le ET logique c'est donc ça, l'association de deux éléments pour en déclencher un nouveau. Les deux éléments condition doivent obligatoirement être présents, et bien sur ça marche pareil avec plus d'éléments en entrée (on le verra après, on va monter en puissance progressivement).

Entrons un peu plus dans le sujet du binaire avec ce qu'on appelle une table de vérité. Une table de vérité est une table qui permet de représenter l'état de la sortie en fonction des états des entrées. Ici je vais volontairement prendre le cas où on a 4 entrées pour coller au bloc AND de l'atelier M3

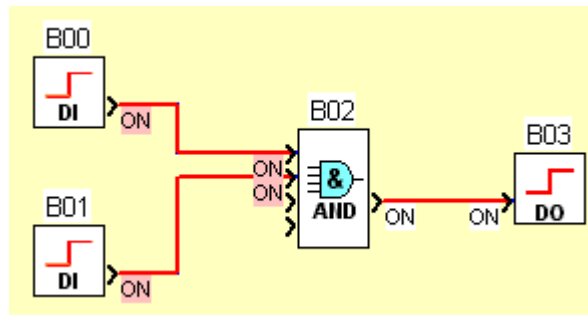
Entrée 1	Entrée 2	Entrée 3	Entrée 4	Sortie
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	0	0
0	0	1	0	0
1	0	1	0	0
0	1	1	0	0
1	1	1	0	0
0	0	0	1	0
1	0	0	1	0
0	1	0	1	0
1	1	0	1	0
0	0	1	1	0
1	0	1	1	0

0	1	1	1	0
1	1	1	1	<b>1</b>

Qu'observe t' on ici ? Seule la dernière ligne donne une sortie vraie et on voit bien qu'il faut que les 4 autres conditions soient vraies pour que ça marche. Nous allons donc tenter une petite expérience dans l'atelier M3. Recréez ce petit programme et passez en mode simulation :



En cliquant sur les blocs DI, changez leurs états et observez le comportement. Voilà ce qu'on obtient si les deux entrées sont à ON



Alors la, il y a de quoi se poser au moins une question... pourquoi la sortie est à ON si on n'a que 2 entrée sur 4 à ON ????

Ceci me permet de vous expliquer rapidement la notion de valeurs par défaut des entrées de blocs. En effet, le logiciel sait quelle est la valeur par défaut (et neutre) à « connecter » virtuellement si l'utilisateur ne place aucun lien. Ainsi, lorsque le programme est en fonctionnement, les entrées non connectées du bloc ET sont par défaut placées à ON. Si on reprend notre table de vérité, cela correspond à travailler uniquement sur les 4 dernières lignes (puisque les deux dernières sorties sont toujours à ON)

Entrée 1	Entrée 2	Entrée 3	Entrée 4	Sortie
0	0	1	1	0
1	0	1	1	0
0	1	1	1	0
1	1	1	1	<b>1</b>

### **Le OU Logique (bloc OR)**

Allez, continuons les joyusetés qui embrouillent tout le monde avec la porte logique OU (OR). Cette porte logique porte aussi bien son nom que la précédente, puisque son but est de réaliser un OU entre deux éléments.

Retrouvons notre exemple de tout à l'heure : une installation solaire et la gestion de son circulateur. Tout à l'heure nous avons :

Pour que le circulateur fonctionne il faut au minimum que :

- la  $T^\circ$  du capteur soit supérieur à celle de l'eau contenue dans le ballon (on notera cette condition *TCaptOK*)
- que la  $T^\circ$  de l'eau dans le ballon n'ait pas atteint la valeur maximum (mettons  $60^\circ$ ) (on notera cette condition *TBallonNonMax*)

Ajoutons à cela une condition pour le faire tourner si la  $T^\circ$  du capteur approche un peu trop de  $0^\circ\text{C}$

- La  $T^\circ$  du capteur est inférieur ou = à  $0^\circ$  (on notera cette condition *GelCapteur*)

Avant nous avions l'équation :

*TCaptOK ET TBallonNonMax* = circulation en marche à laquelle nous ajoutons le OU *GelCapteur* ce qui nous donne :

*(TCaptOK ET TBallonNonMax) OU GelCapteur* = circulation en marche.

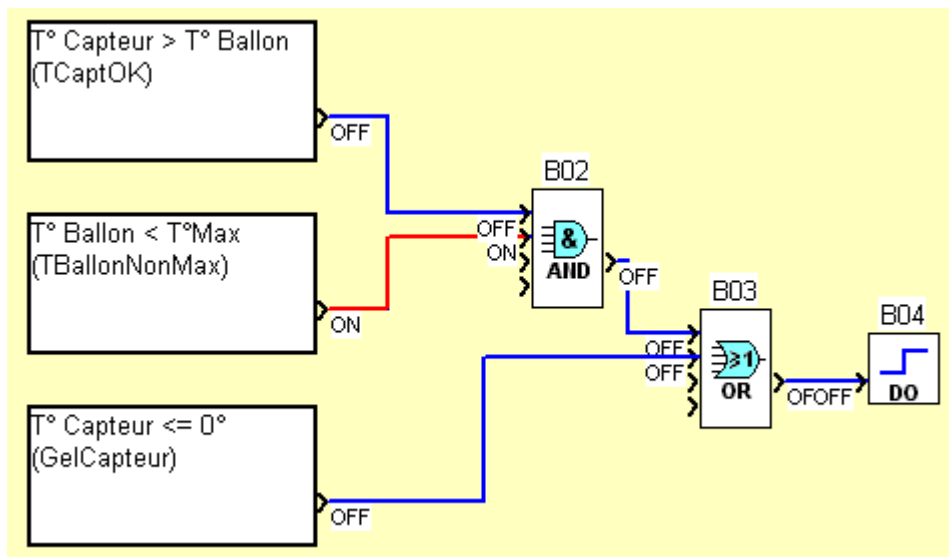
Comment on interprète tout ça? Comme cela se lit. Avec les conditions telles qu'elles sont écrites, notre circulateur sera forcément en marche si le capteur est plus chaud que le ballon de stockage et que ce ne dernier n'est pas trop chaud, ou si la température du capteur est proche, égale ou inférieur à  $0^\circ$ .

Revenons dans le domaine du binaire avec la table de vérité du bloc OU.

Entrée 1	Entrée 2	Entrée 3	Entrée 4	Sortie
0	0	0	0	<b>0</b>
1	0	0	0	1
0	1	0	0	1
1	1	0	0	1
0	0	1	0	1
1	0	1	0	1
0	1	1	0	1
1	1	1	0	1
0	0	0	1	1
1	0	0	1	1
0	1	0	1	1
1	1	0	1	1
0	0	1	1	1
1	0	1	1	1
0	1	1	1	1
1	1	1	1	1

Alors oui cette fois ci nous n'avons qu'un seul 0, car la sortie du bloc est active dès qu'au moins une des entrées est active.

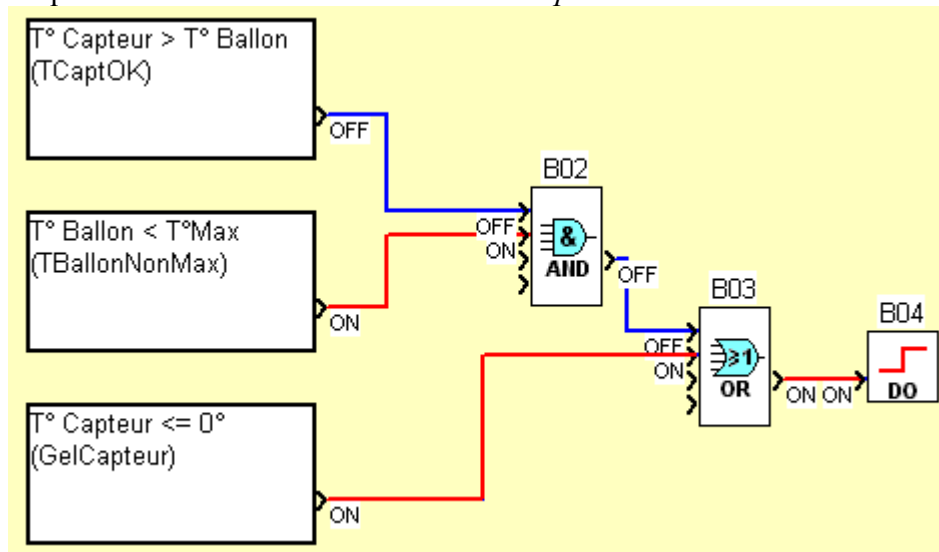
Allons maintenant jouer avec l'atelier M3 pour faire quelques tests :



Dans cet exemple, j'ai dessiné des « boites noires » qui correspondent à nos différentes conditions afin de mieux se représenter la fonction en rapport avec l'exemple que j'ai choisi tout en gardant un programme simple à comprendre.

Dans la capture que j'ai prise juste au dessus, on peut imaginer qu'on se situe dans un cas où le ballon n'est pas à température maximum, mais que l'état actuel ne permet pas de le chauffer. Ceci est vrai la nuit ou si le ciel est couvert.

Simulons maintenant une baisse de la température extérieure qui va refroidir le capteur en dessus de 0°, ce qui provoquera l'activation de la condition *GelCapteur*.

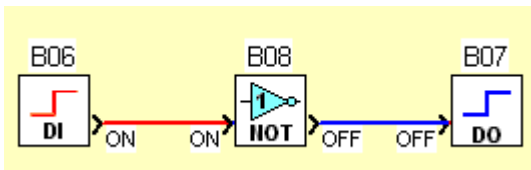
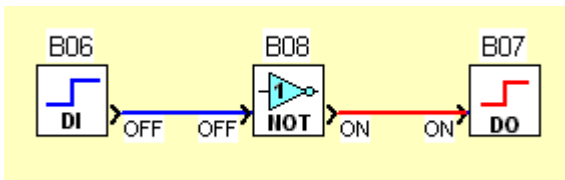


Et voilà, notre OU fait son office, puisque le circulateur est en marche alors qu'on n'est pas en train de chauffer, ceci étant une solution pour éviter le gel de l'installation (il en existe bien d'autres hein, ce n'est qu'un exemple de travail)

### **Le NON Logique (bloc NOT)**

Alors en voilà un bloc qui est contrariant par nature, puisque son seul et unique objectif est de dire NON (OFF) lorsqu'on a OUI (ON), et de dire OUI lorsqu'on a NON.

L'exemple ici est très simple :



Voici le bloc dans ses deux états possibles, et vous constatez donc qu'il porte bien son nom puisque sa sortie est toujours l'opposé de son entrée. Sa table de vérité est donc forcément très simple elle aussi :

Entrée 1	Sortie
0	1
1	0

Essayons maintenant de le mettre en situation.

Imaginons que nous ayons une vanne 3 voies pilotée par notre programme, dont la position est déterminée par le fait que l'on soit en surchauffe (vanne qui aura donc le rôle de router le fluide caloporteur vers un système de décharge. Ceci va quelque peu modifier notre programme puisque pour le coup, notre circulateur devra continuer de fonctionner même si la condition *TBallonNonMax* est atteinte.

Commençons par réfléchir à notre équation logique :

Le circulateur devra fonctionner si :

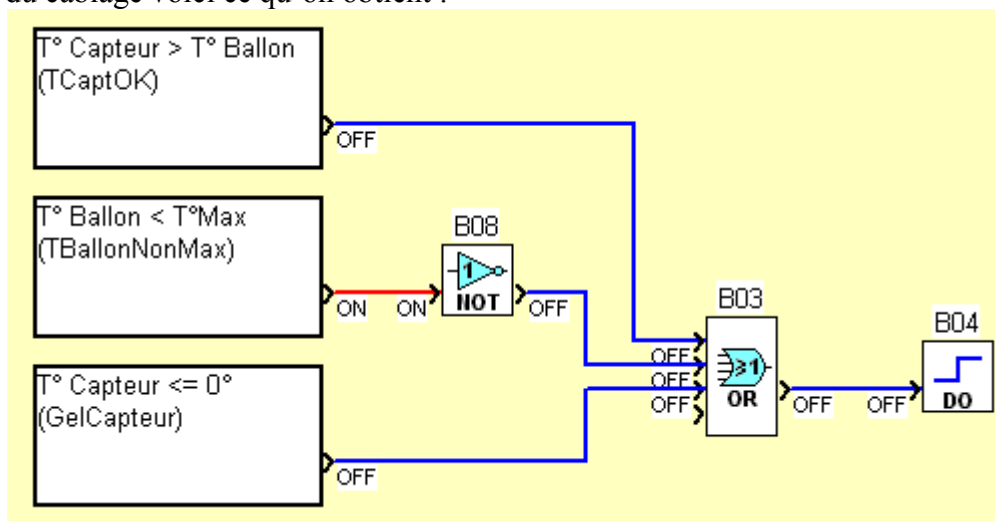
- TCaptOK ( $T^{\circ}$  capteur >  $T^{\circ}$  ballon)
- Ou NON TBallonNonMax (équivalent à  $T_{Ballon} > Maximum$ )
- Ou GelCapteur

Dans notre cas nous n'avons plus de condition ET, puisque chacune de nos conditions suffit à faire marcher le circulateur.

L'équation logique sera donc :

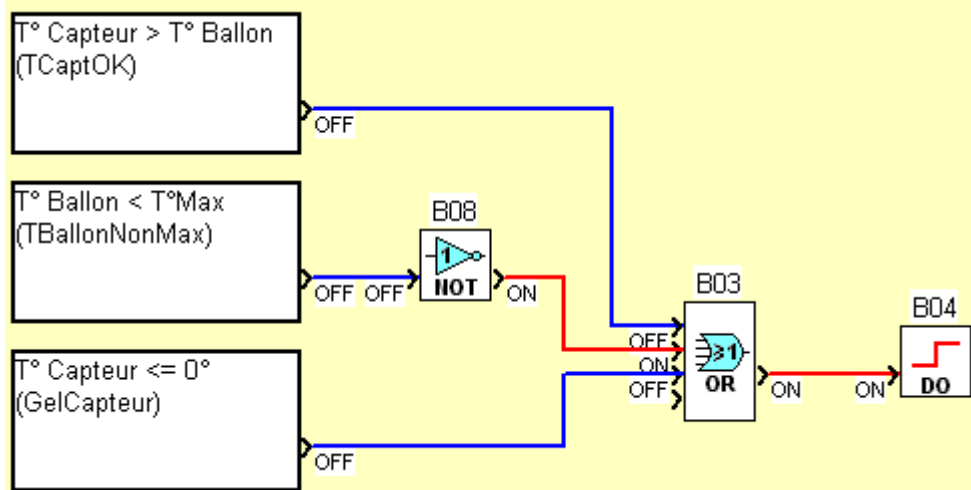
$TCaptOK \text{ OU } (NON \ TBallonNonMax) \text{ OU } GelCapteur = \text{marche Circulateur.}$

Au niveau du câblage voici ce qu'on obtient :

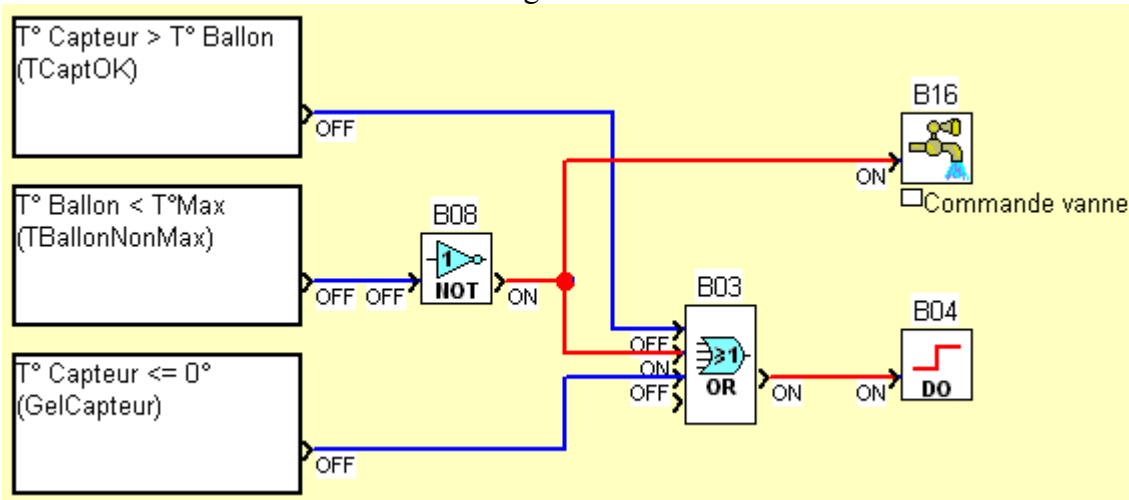


Ici nous sommes dans l'état initial, ou l'on a pas de quoi chauffer notre ballon, que ça ne gèle pas, et qu'on n'est pas en surchauffe (la  $T^{\circ}$  du ballon est bien inférieur au max autorisé).

Si maintenant on réalise la condition NON TBallonNonMax (ou TBallonMax) on obtient la mise en marche du circulateur :



Avec ce schéma, si nous voulons commander une vanne de décharge, nous utiliserons directement la sortie du bloc NOT B08 comme sur le câblage suivant

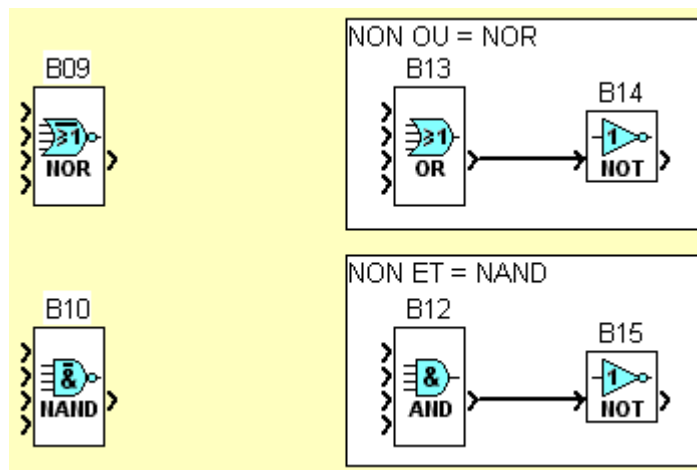


Bien entendu ce schéma n'est donné que pour l'exemple, et cette situation n'en n'est qu'une parmi des dizaines possibles, mais il permet de montrer différentes associations de bloc logiques.

### **Le NOR (NON OU) et le NAND (NON ET)**

Ici, aucune surprise si vous avez déjà compris les 3 blocs précédents, puisque le NOR revient à l'association d'un bloc OU et d'un bloc NON, alors que le NAND est l'association d'un bloc ET et d'un bloc NON.





Pour avoir leurs tables de vérité, rien de plus simple, il suffit de prendre celle du bloc de base, et d'inverser l'état de la sortie :

### Le NAND

Entrée 1	Entrée 2	Entrée 3	Entrée 4	Sortie
0	0	0	0	1
1	0	0	0	1
0	1	0	0	1
1	1	0	0	1
0	0	1	0	1
1	0	1	0	1
0	1	1	0	1
1	1	1	0	1
0	0	0	1	1
1	0	0	1	1
0	1	0	1	1
1	1	0	1	1
0	0	1	1	1
1	0	1	1	1
0	1	1	1	1
1	1	1	1	0

### Le NOR

Entrée 1	Entrée 2	Entrée 3	Entrée 4	Sortie
0	0	0	0	1
1	0	0	0	0
0	1	0	0	0
1	1	0	0	0
0	0	1	0	0
1	0	1	0	0
0	1	1	0	0
1	1	1	0	0
0	0	0	1	0
1	0	0	1	0
0	1	0	1	0
1	1	0	1	0
0	0	1	1	0
1	0	1	1	0

0	1	1	1	0
1	1	1	1	0

Notons que ces blocs vous seront certainement plus utiles lorsque vous en serez à l'optimisation poussée de votre programme, ou alors si vous êtes un grand maître de la logique booléenne et que vous ne confondrez jamais un AND et un NAND.

### **Le OU EXCLUSIF (bloc XOR ... le shérif de l'espace)**

Alors je m'excuse pour tous ceux qui ne comprennent pas la blague du shérif de l'espace, mais pour moi c'est une grande référence à ma petite enfance et je ne pouvais simplement pas m'empêcher de la faire (la référence vient [d'ici](#), avec notre grand chanteur à texte Bernard Minet)

Bref, le XOR est donc un ou exclusif, en gros c'est un OU, mais qui n'accepte pas que toutes ses entrées soient actives à la fois... c'est soit l'un, soit l'autre mais PAS les deux !!

Voyons sa table de vérité :

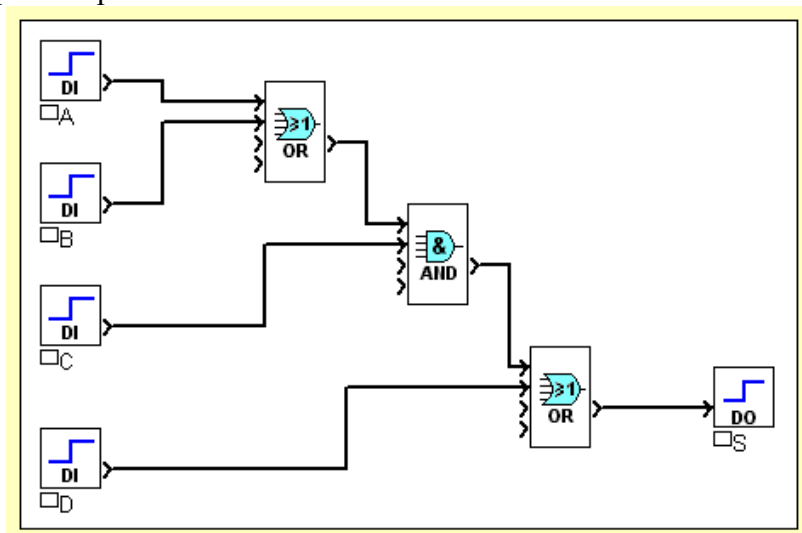
Entrée 1	Entrée 2	Sortie
0	0	0
1	0	1
0	1	1
1	1	0

Très honnêtement, j'ai rarement vu ce bloc utilisé dans des programmes, et je ne vais pas pousser son étude plus loin que ça, car il reste un bloc à étudier qui à lui tout seul permet de réaliser (presque) tout ce que font les blocs logiques standard.

### **La fonction BOOLEENNE !!**

Alors pour le coup, celle la va sûrement en perdre quelques un au début, mais accrochez vous un peu car elle est très utile !!

Prenons cet exemple complètement abstrait :



Et écrivons son équation logique en commençant par le haut :

A et B sont connectés à un bloc OU nous avons donc pour commencer :

$A \text{ OU } B$

Ensuite la sortie du OU est connecté avec C à un bloc ET, nous obtenons donc :

$(A \text{ OU } B) \text{ ET } C$

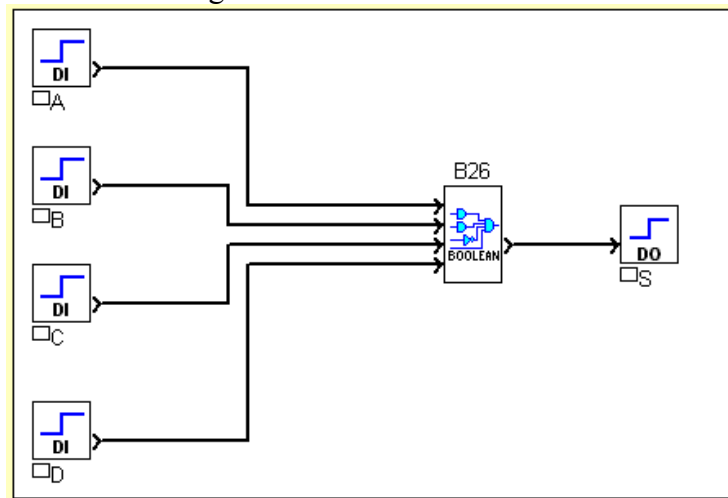
Enfin la sortie du bloc est connectée à un OU avec D, ce qui donne au final :

$$S = ((A \text{ OU } B) \text{ ET } C) \text{ OU } D$$

Comme en mathématique, ce genre d'opération peut se développer ou se factoriser et souvent se simplifier, mais je ne me chargerai pas d'expliquer profondément ceci car on trouve tout ce qu'il faut sur internet, et certainement bien mieux présenté que ce que je pourrai le faire ici (pour ceux qui souhaitent pousser le truc à fond, je les invite à aller visiter cette page [wikipedia](http://wikipedia) ... attention tout de même, cela demande de s'y pencher sérieusement)

Bref, nous, puisqu'on ne fait pas des trucs trop compliqués, on va se limiter à transposer ceci dans une fonction booléenne.

Voici ce que cela donne niveau câblage :



Déjà c'est sur c'est nettement plus lisible graphiquement, mais notons tout de même que pour le coup on ne sait pas trop comment ça marche du premier coup d'œil, et il va donc nous falloir accéder au paramétrage du bloc BOOLEEN pour y voir plus clair. C'est l'avantage et l'inconvénient de ce bloc, il permet de simplifier les programmes, mais il en cache la logique de base.

Bref, attelons nous à le paramétrer pour avoir le même résultat que précédemment.

Pour une fois je ne vais pas donner la solution de suite, mais je vais plutôt vous proposer de résoudre l'exercice ensemble, avec ce qu'il faut d'explication.

Notre équation était donc :

$$S = ((A \text{ OU } B) \text{ ET } C) \text{ OU } D$$

Commençons par un peu de raisonnement simple en lisant ceci. On voit que la dernière condition est OU **D**, ce qui veut dire que dans notre table de vérité, chaque fois que **D** sera à 1, notre sortie sera forcément à 1.

Vous pouvez donc déjà remplir les 8 dernières lignes de cette table de vérité, car sur ces 8 lignes **D** est à 1.

A	B	C	D	S
0	0	0	0	?
1	0	0	0	?
0	1	0	0	?
1	1	0	0	?
0	0	1	0	?
1	0	1	0	?
0	1	1	0	?
1	1	1	0	?
0	0	0	1	1

1	0	0	1	<b>1</b>
0	1	0	1	<b>1</b>
1	1	0	1	<b>1</b>
0	0	1	1	<b>1</b>
1	0	1	1	<b>1</b>
0	1	1	1	<b>1</b>
1	1	1	1	<b>1</b>

Ensuite, il nous reste la sous expression  $(A \text{ OU } B) \text{ ET } C$ .

Dans cette sous expression, on voit que **C** est une condition obligatoire (puisque c'est un ET), et que par conséquent cette sous expression sera vraie SI et seulement SI **C** est à 1. En reprenant notre table de vérité, on peut donc remplir les 4 premières lignes de 0 puisque **C** est à 0.

A	B	C	D	S
0	0	0	0	<b>0</b>
1	0	0	0	<b>0</b>
0	1	0	0	<b>0</b>
1	1	0	0	<b>0</b>
0	0	1	0	<b>?</b>
1	0	1	0	<b>?</b>
0	1	1	0	<b>?</b>
1	1	1	0	<b>?</b>
0	0	0	1	<b>1</b>
1	0	0	1	<b>1</b>
0	1	0	1	<b>1</b>
1	1	0	1	<b>1</b>
0	0	1	1	<b>1</b>
1	0	1	1	<b>1</b>
0	1	1	1	<b>1</b>
1	1	1	1	<b>1</b>

Notre table de vérité est déjà bien complétée, et on va donc s'intéresser aux 4 lignes qu'il reste.

A	B	C	D	S
0	0	1	0	<b>?</b>
1	0	1	0	<b>?</b>
0	1	1	0	<b>?</b>
1	1	1	0	<b>?</b>

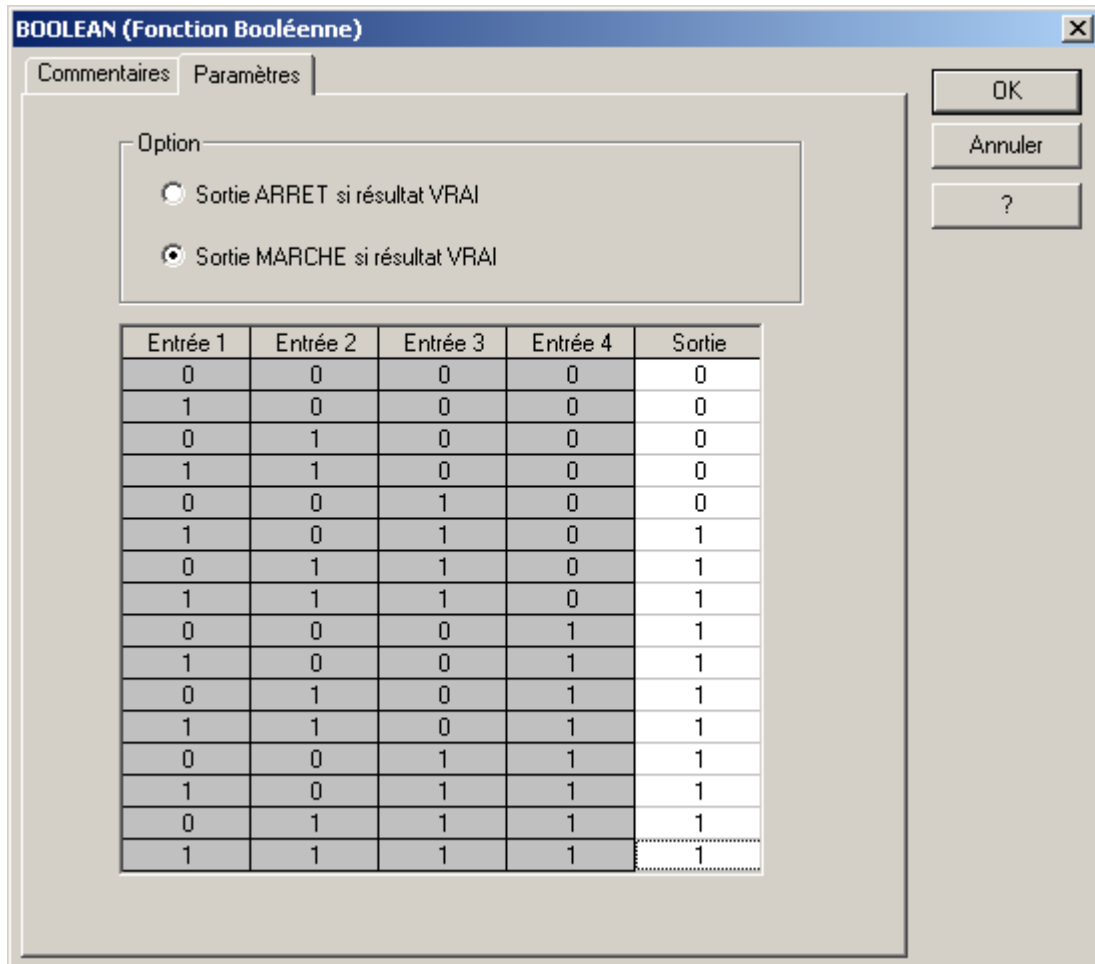
On reprend notre sous expression  $(A \text{ OU } B) \text{ ET } C$  et en relisant ceci, avec un petit tableau bien simple, on comprend qu'il nous faut les cas où on a **C**, et où on a soit **A**, soit **B**, soit **les deux**, ce qui permet de déduire que la sortie de la première de ces 4 lignes sera un 0, et les trois autres un 1.

Et voilà le résultat final.

A	B	C	D	S
0	0	0	0	<b>0</b>
1	0	0	0	<b>0</b>
0	1	0	0	<b>0</b>
1	1	0	0	<b>0</b>
0	0	1	0	<b>0</b>
1	0	1	0	<b>1</b>
0	1	1	0	<b>1</b>
1	1	1	0	<b>1</b>
0	0	0	1	<b>1</b>

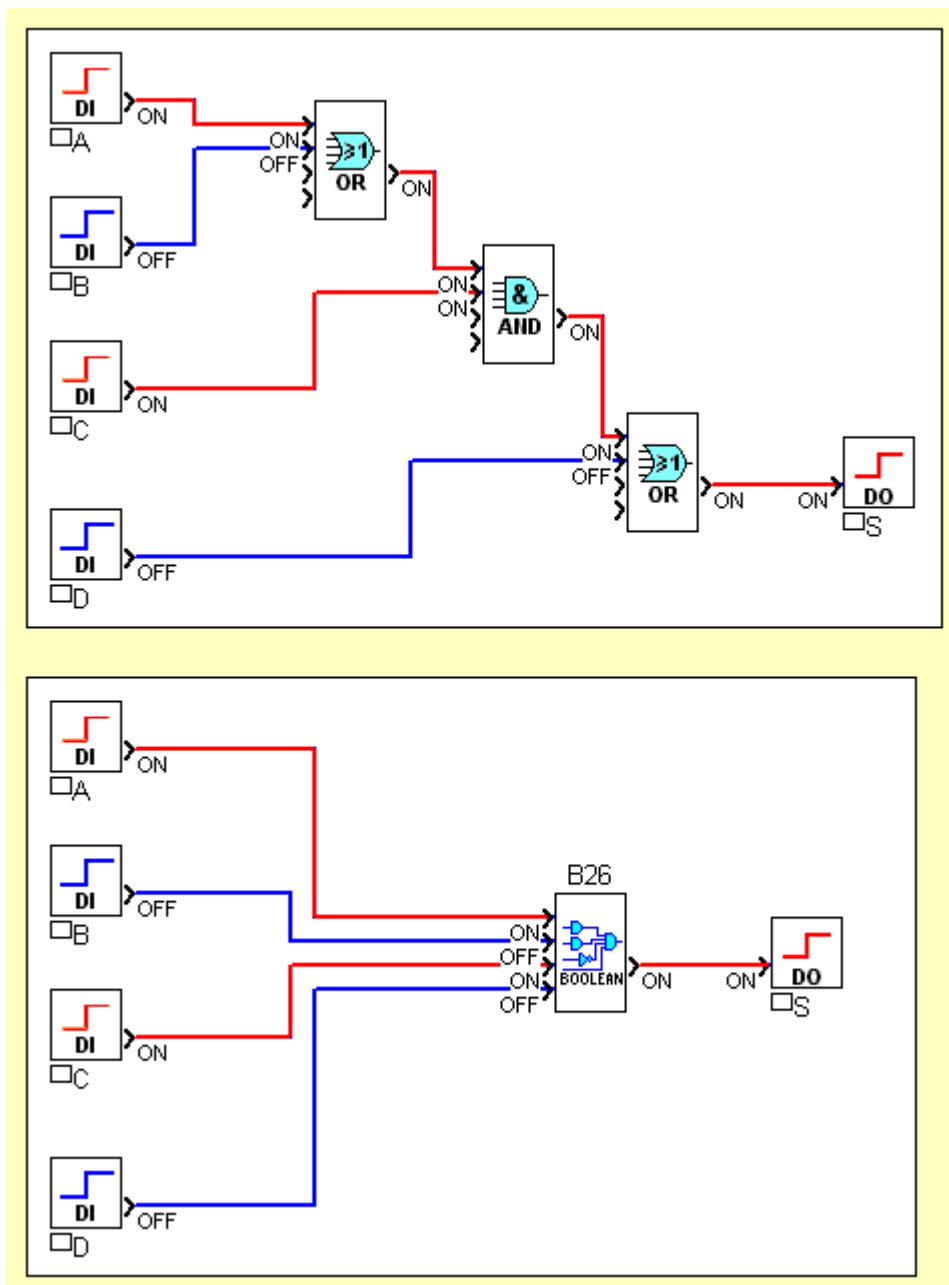
1	0	0	1	<b>1</b>
0	1	0	1	<b>1</b>
1	1	0	1	<b>1</b>
0	0	1	1	<b>1</b>
1	0	1	1	<b>1</b>
0	1	1	1	<b>1</b>
1	1	1	1	<b>1</b>

Si je vous ai fait travailler un peu les tables de vérité, ce n'est pas pour rien, regarder le paramétrage du bloc BOOLEEN :



Quelle surprise ça alors... c'est une table de vérité (que j'ai paramétré ici avec le résultat de notre petit exercice)

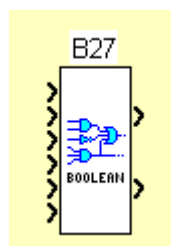
Allez pour le plaisir, un petit passage en simulation pour comparer le résultat de ces deux programmes :



Et voila on obtient le même résultat !!

### ***La fonction super BOOLEENNE !!***

Haha, quand y'en a plus y'en a encore !!!



Dans l'onglet FBD\_C, vous pouvez trouver une petite coquine que je nomme super booléenne, car elle fonctionne comme la fonction booléenne normale, mais avec 6 entrée... et donc une table de vérité de **64 lignes, et 2 sorties**. Alors non je ne vais pas vous faire faire encore un exercice avec celle-ci, car cela n'apportera rien de plus qu'avec le précédent. Je n'ai qu'un seul conseil à vous

donner, Si vous vous lancez dans l'utilisation de ce bloc, utiliser un tableur quelconque pour mettre d'abord en forme votre table de vérité avec des mots, avant de mettre des 1 n'importe comment dans l'une des deux colonnes sorties.

Pour ma part j'ai eu besoin de ce bloc pour écrire la logique des vannes du programme de Gallo  
Jean Paul que vous trouverez sur [le site APPER](#)

## Les blocs FBD

Pour les blocs FBD, je vais décrire les blocs selon plusieurs caractéristiques qui sont :

- Les entrées (noté E)
- Les sorties (noté S)
- Les paramètres sortis (noté PS)
- Les paramètres (noté P)
- Le fonctionnement

Ceci sera présenté sous forme de tableau. La colonne N° indique le numéro de l'entrée ou de la sortie (on compte en partant du haut) selon le contexte.

Alors voilà déjà un truc qui doit en surprendre plus d'un... y'a des paramètres et des paramètres sortis...mais c'est quoi ces trucs la encore !!!

Alors non ne vous sauvez pas car en fait c'est très simple :

Un paramètre se trouve...dans la feuille de paramétrage... oui logique me direz vous, mais le paramètre sortis aussi s'y trouve...alors c'est quoi donc la différence entre les deux...

La vraie différence entre ces deux types de paramètres que je me permets de distinguer, c'est que le paramètre sortie se trouve non seulement dans la page de paramétrage, mais aussi sur une sortie du bloc fonction. L'intérêt de ceci peut sembler limité, mais lorsque vous aurez passé le tutoriel sur la programmation de la face avant, vous comprendrez toute la subtilité de la chose. Pour le moment je ne vais pas vous embêter plus longtemps avec, car ceci n'a pas d'importance sans les fonctions d'affichage.

## Les TIMERS

Les timers servent à effectuer des opérations sur le temps, et il en existe suffisamment de sortes différentes pour toujours trouver celui qui nous convient. Tout les timers (sauf BW impulsion sur front) ont un homologue avec consignes externes, que je ne décrirai pas en détail car le fonctionnement est identique au bloc standard à la seule différence que les consignes sont des valeurs d'entrées.

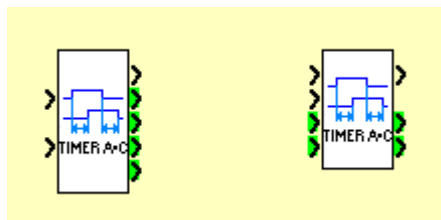
### Timer AC (retardateur)

Ce timer à pour but de retarder sa sortie par rapport à son entrée.

**Icône :**

A gauche la version normale, à droite la version avec consignes externes.





### Caractéristiques :

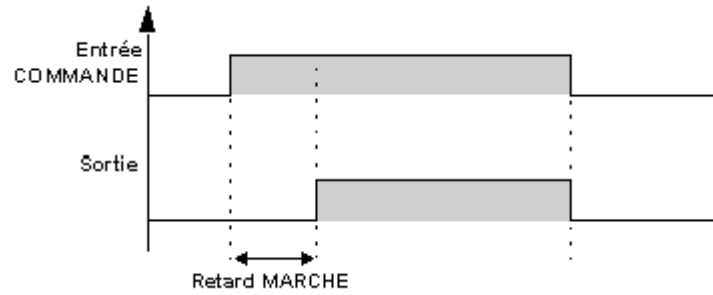
Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Commande	Active le timer lorsque cette entrée est validée (à ON). Le timer commence alors à compter. Si l'entrée est désactivée le comptage s'arrête et revient à 0
	2	TOR	Reset	Remet à zéro le timer (reset des valeurs courante retard marche et arrêt)
S	1	TOR	Sortie	S'active lors que la valeur courante retard marche atteint la consigne retard marche, et s'arrête lorsque valeur courante retard arrêt atteint la consigne retard arrêt
	3	ANA	Valeur courante retard marche	Temps écoulé depuis l'activation de l'entrée « Commande »
	5	ANA	Valeur courante retard arrêt	Temps écoulé depuis l'activation de la sortie du bloc
PS	2	ANA	Consigne retard marche	Consigne de retard avant l'activation de la sortie du bloc.
	4	ANA	Consigne retard arrêt	Consigne de retard avant l'arrêt de la sortie du bloc
P	NA	NA	Base de temps	Base de temps utilisé par le timer, peut être : <ul style="list-style-type: none"> <li>- 1 seconde</li> <li>- 100ms</li> <li>- Période du programme</li> </ul>
	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur
	NA	TOR	Modif. Autorisée	Indique si le bloc est paramétrable par le menu générique de la face avant du M3

### Fonctionnement :

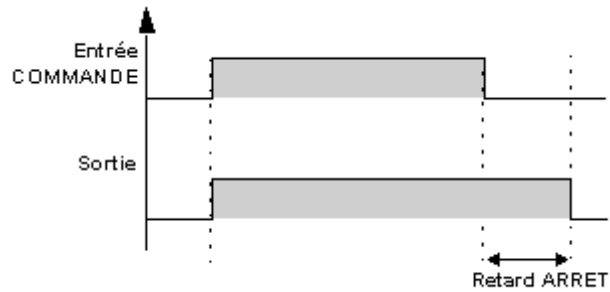
Lorsqu'on active l'entrée 1 de ce bloc, celui-ci se met à compter le temps. Une fois que la valeur courante retard marche à atteint la consigne, la sortie du bloc s'active et c'est la valeur courante retard arrêt qui est incrémenté. Une fois que cette dernière valeur à atteint la consigne retard arrêt, la sortie du bloc se désactive. Le timer ne fait rien d'autre tant que son entrée 1 n'a pas été désactivée ou que l'entrée reset n'a pas été activée.

### Chronogrammes (issues de l'aide en ligne de l'atelier) :

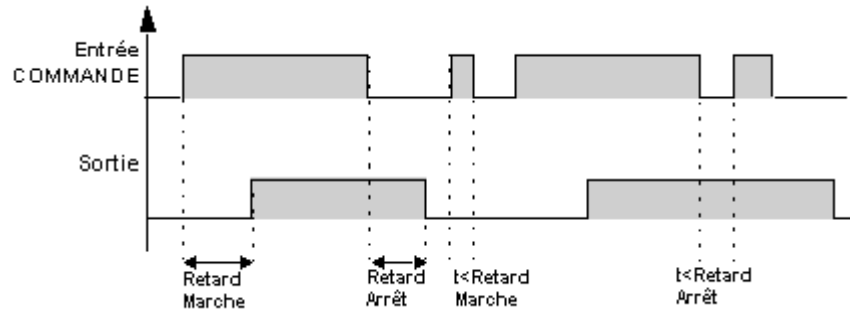
Fonction A :



Fonction C :



Fonction A-C :

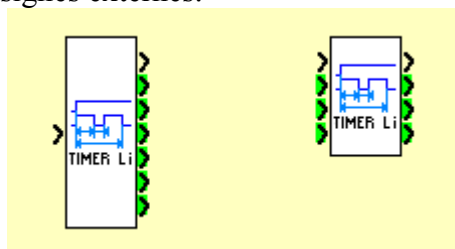


### Timer LI (clignoteur)

Ce timer à pour but d'envoyer un signal périodique de façon répétitive.

**Icône :**

A gauche la version normale, à droite la version avec consignes externes.

**Caractéristiques :**

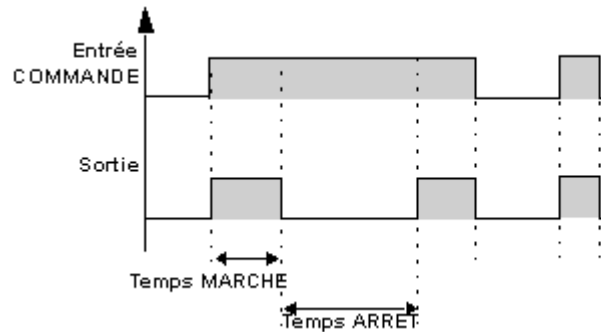
Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Commande	Active le timer lorsque cette entrée est validée (à ON). Le timer commence alors à compter. Si l'entrée est désactivée le comptage s'arrête et revient à 0
S	1	TOR	Sortie	S'active périodiquement lorsque l'entrée est active.
	3	ANA	Valeur courante retard marche	Temps écoulé avec la sortie à l'état ON
	5	ANA	Valeur courante retard arrêt	Temps écoulé avec la sortie à l'état OFF
	7	ANA	Valeur courante nombre de cycle/durée	Nombre de cycles/durée de clignotement passé
PS	2	ANA	Consigne retard marche	Consigne de temps de maintien de la sortie à ON
	4	ANA	Consigne retard arrêt	Consigne de temps de maintien de la sortie à OFF
	6	ANA	Consigne Nombre de cycle/durée	Consigne du nombre de cycle/durée de clignotement
P	NA	NA	Base de temps	Base de temps utilisé par le timer, peut être : <ul style="list-style-type: none"> <li>- 1 seconde</li> <li>- 100ms</li> <li>- Période du programme</li> </ul>
	NA	NA	Mode de clignotement	Définit si le clignotement est continu, définit sur une durée, ou sur un nombre de cycles.
	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur
	NA	TOR	Modif. Autorisée	Indique si le bloc est paramétrable par le menu générique de la face avant du M3

**Fonctionnement :**

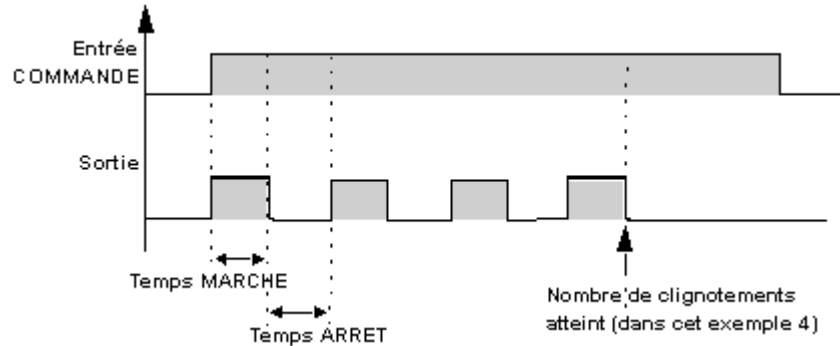
Lorsqu'on active l'entrée 1 de ce bloc, celui-ci se met à compter le temps. Une fois que la valeur courante retard marche à atteint la consigne, la sortie du bloc s'active et c'est la valeur courante retard arrêt qui est incrémenté. Une fois que cette dernière valeur à atteint la consigne retard arrêt, la sortie du bloc se désactive. Le timer ne fait rien d'autre tant que son entrée 1 n'a pas été désactivée ou que l'entrée reset n'a pas été activée.

**Chronogrammes (issues de l'aide en ligne de l'atelier) :**

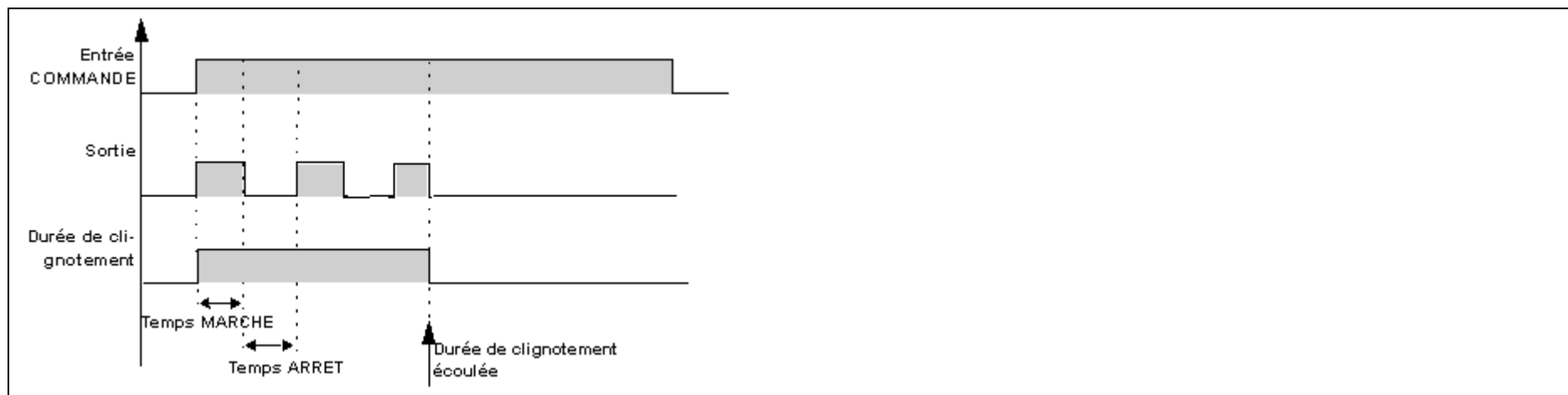
Mode Clignotement continu :



Mode Nombre de clignotements :



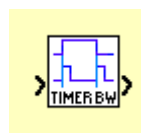
Mode Durée de clignotement :



### Timer BW (impulsionnel)

Ce timer ne travail pas vraiment sur le temps, puisque son seul et unique but est de générer une impulsion sur sa sortie lors du changement d'état de son entrée.

**Icône :**



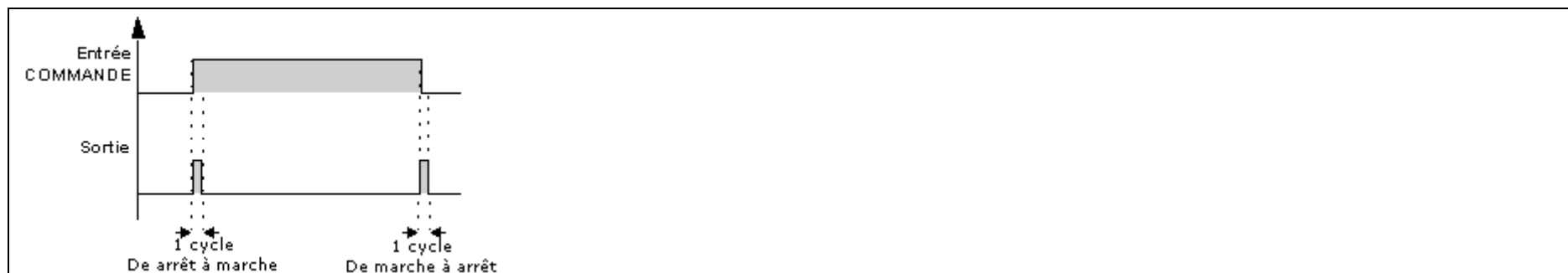
**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Commande	Activation du timer
S	1	TOR	Sortie	Sortie impulsion
P	NA	NA	Mode de fonctionnement	Paramètre permettant de définir si les impulsions sont envoyées sur front montant (d'arrêt à marche), sur front descendant (de marche à arrêt) ou les deux.

**Fonctionnement :**

Le timer BW se contente de détecter un changement de sa commande, et d'envoyer une impulsion durant un cycle d'automate (temps entre 10 et 20ms selon les programmes) en fonction du paramétrage qui a été fait.

**Chronogramme :**

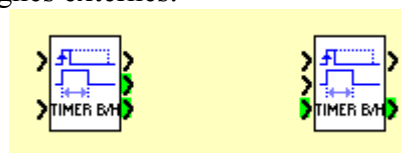


### Timer BH (temporisateur)

Ce timer a pour but de fixer une durée à la sortie quelque soit la durée de son entrée.

#### Icône :

A gauche la version normale, à droite la version avec consignes externes.



#### Caractéristiques :

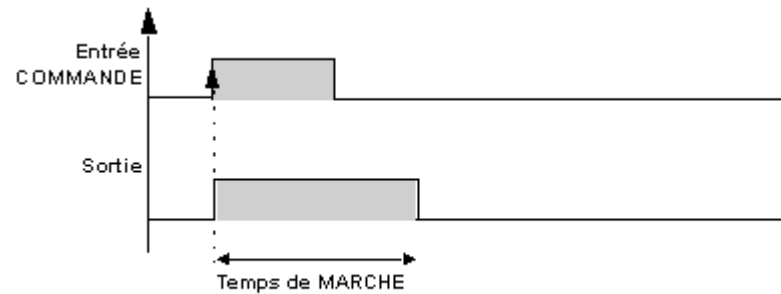
Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Commande	Active le timer lorsque cette entrée est validée (à ON).
	2	TOR	Reset	Remet à zéro le timer.
S	1	TOR	Sortie	S'active lorsque l'entrée est activée.
	3	ANA	Valeur courante temps de marche	Temps écoulé depuis l'activation
PS	2	ANA	Consigne temps de marche	Consigne de durée de marche de la sortie
P	NA	NA	Base de temps	Base de temps utilisé par le timer, peut être : - 1 seconde - 100ms - Période du programme
	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur
	NA	TOR	Modif. Autorisée	Indique si le bloc est paramétrable par le menu générique de la face avant du

**Fonctionnement :**

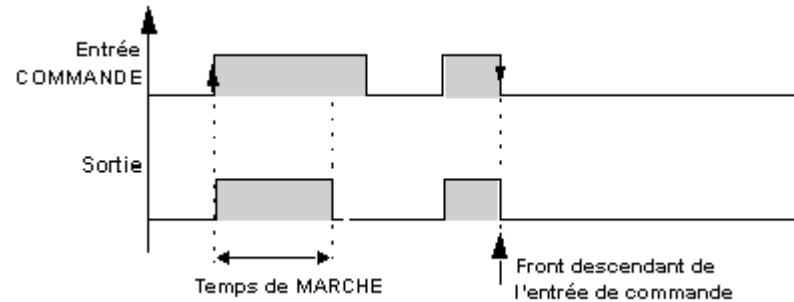
Lorsqu'on active l'entrée commande de ce bloc, la sortie est immédiatement activée. En mode « fonction B », sa sortie aura une durée fixe quelque soit la durée de la commande (plus longue ou plus courte). En mode « fonction H », la sortie sera désactivée si la commande est désactivée avant la fin du temps défini par le paramètre « Consigne temps de marche ».

**Chronogrammes (issues de l'aide en ligne de l'atelier) :**

Fonction B :



Fonction H :

**Timer TOT (Totalisateur)**

Ce timer à pour but de totaliser (compter) son temps d'activation

 **Icône :**

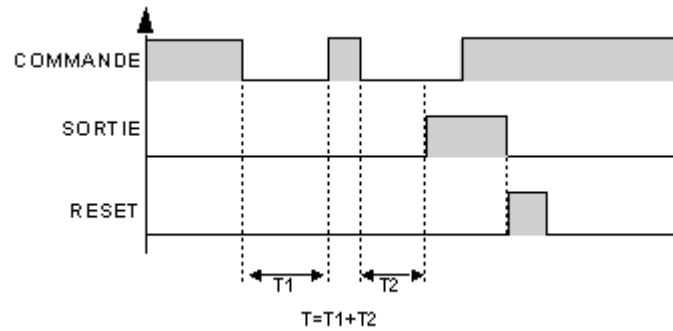
A gauche la version normale, à droite la version avec consignes externes.



**Caractéristiques :**

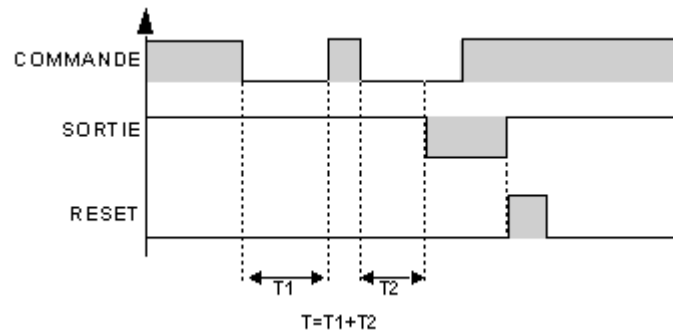
Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Commande	Active le timer lorsque cette entrée est validée (à ON).
	2	TOR	Reset	Remet à zéro le timer.
S	1	TOR	Sortie	S'active selon le mode paramétré
	3	ANA	Valeur courante	Temps comptabilisé
PS	2	ANA	Consigne de temps	Consigne de temps utilisé différemment selon le mode
P	NA	NA	Base de temps	Base de temps utilisé par le timer, peut être : - 1 seconde - 100ms - Période du programme
	NA	NA	Mode	Choisi le mode de fonctionnement du totalisateur.
	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur
	NA	TOR	Modif. Autorisée	Indique si le bloc est paramétrable par le menu générique de la face avant du M3
<p><b>Fonctionnement :</b>            Voir sur les chronogrammes :  <b>Chronogrammes (issues de l'aide en ligne de l'atelier) :</b>  <b>Fonction At :</b></p>				





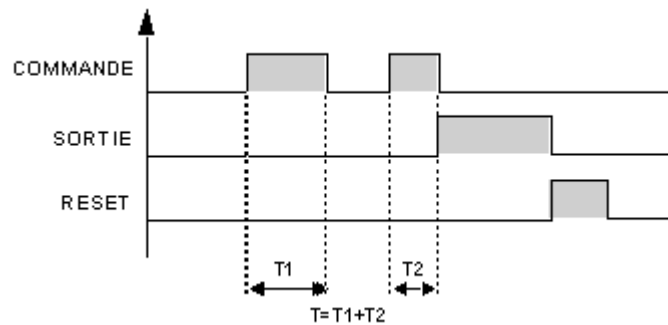
En mode AT, le timer fait la somme des temps de non activation de son entrée, et active sa sortie une fois que cette somme à atteint la consigne. La sortie est remise à zéro uniquement lorsque l'entrée reset est activée.

**Fonction Ht :**



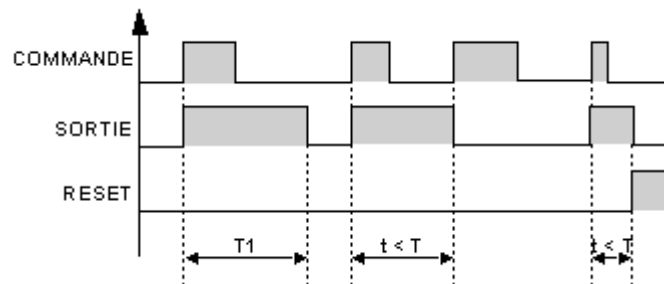
Le mode HT est identique au mode AT, sauf que la sortie est inversée (sortie à 1 jusqu'à que la somme des temps de la commande atteigne la consigne).

**Fonction T :**



En mode T, le fonctionnement est similaire au mode AT, sauf que le timer compte le temps d'activation de sa commande au lieu du temps de non activation.

### Fonction Tt :



En mode TT, lorsque la commande est activée la sortie s'active pendant la durée donnée par la consigne (indépendamment de la durée d'activation de la commande). Si la commande est désactivée puis réactivée avant la fin de la consigne, la sortie est désactivée. Dans tous les cas, si l'entrée reset est activée avant la fin de la consigne, la sortie est désactivée.

## Les Afficheurs

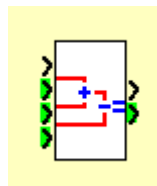
Alors pour les afficheurs, je n'expliquerai pas leur fonctionnement car celle-ci et leur utilisation à déjà été détaillée et poussée dans [ce tutoriel](#) disponible sur le site.

## Les Fonctions Mathématiques

### Addition/Soustraction (ADD/SUB)

La fonction ADD/SUB permet de faire une soustraction et/ou une addition entre des valeurs numériques. Dans une régulation solaire cette fonction est généralement utilisée pour introduire les hystérésis sur les trigger.

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Propagation d'erreur	Permet de propager l'erreur ou la saturation d'un calcul
	2	ANA	Entrée 1	Entrée valeur
	3	ANA	Entrée 2	Entrée addition
	4	ANA	Entrée 3	Entrée soustraction
S	1	TOR	Erreur/débordement	S'active en cas d'erreur ou de saturation
	2	ANA	Sortie calcul	Résultat de l'addition/soustraction

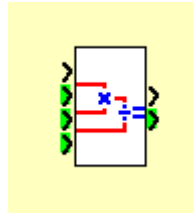
**Fonctionnement :**

La sortie résultat calcul donne le résultat de la somme/soustraction effectuée. Il est possible de faire une addition et une soustraction en même temps. Le millenium 3 effectue l'ensemble de ses calculs sur des entiers 16 bits : plage de valeur de -32768 à 32767, en conséquence, si le résultat d'une opération donne un résultat inférieur au minimum, ou supérieur au maximum, la sortie propagation d'erreur est activée pour indiquer la saturation.

### Multiplication/Division (MUL/DIV)

La fonction MUL/DIV permet de faire une multiplication et/ou une division entre des valeurs numériques.

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Propagation d'erreur	Permet de propager l'erreur ou la saturation d'un calcul
	2	ANA	Entrée 1	Entrée valeur
	3	ANA	Entrée 2	Entrée multiplication
	4	ANA	Entrée 3	Entrée division
S	1	TOR	Erreur/débordement	S'active en cas d'erreur ou de saturation
	2	ANA	Sortie calcul	Résultat de la multiplication/division

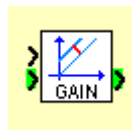
**Fonctionnement :**

La sortie résultat calcul donne le résultat de la multiplication/division effectuée. Il est possible de faire une multiplication et une division en même temps. Le millenium 3 effectue l'ensemble de ses calculs sur des entiers 16 bits : plage de valeur de -32768 à 32767, en conséquence, si le résultat d'une opération donne un résultat inférieur au minimum, ou supérieur au maximum, ou dans le cas d'une division par 0, la sortie propagation d'erreur est activée pour indiquer la saturation/l'erreur.

**GAIN**

La fonction GAIN permet d'appliquer un gain linéaire avec offset à une valeur d'entrée.

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Validation fonction	Active la fonction
	2	ANA	Entrée valeur	Valeur sur laquelle le calcul de gain va être appliqué
S	1	ANA	Sortie valeur	Résultat du gain
P	NA	ANA	Numérateur A	Paramètre A de l'équation $S = (Entrée \times A) / B + C$

	NA	ANA	Dénominateur B	Paramètre B de l'équation $S = (\text{Entrée} \times A) / B + C$
	NA	ANA	Offset C	Paramètre C de l'équation $S = (\text{Entrée} \times A) / B + C$
<b>Fonctionnement :</b>				
La sortie S est égale au résultat de l'équation $S = (\text{Entrée} \times A) / B + C$ . Le calcul n'est effectué que si l'entrée validation n'est pas connecté ou qu'elle est à ON.				

## Les Compteurs

### Compteur à présélection (preset counter)

Ce compteur permet d'activer une commande après un comptage prédéfini

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Comptage	Incrémente le compteur
	2	TOR	Décomptage	Décrémente le compteur
	3	TOR	Reset	Remet à zéro le compteur.
S	1	TOR	Sortie	S'active lorsque la valeur comptée atteint la présélection.
	3	ANA	Valeur courante	Donne la valeur courante du compteur
	4	ANA	Valeur courante temporisation	Temps écoulé depuis l'activation de la sortie (en mode cyclique uniquement)
PS	2	ANA	Valeur de présélection	Présélection du compteur
P	NA	NA	Mode comptage	Deux modes possibles : - Compte vers la présélection - Décompte à partir de la présélection
	NA	NA	Cycle	Deux modes possibles - Cycle unique

				- Cycle répétitif
	NA	ANA	Durée de l'impulsion	Durée de l'impulsion de la sortie en mode Cycle répétitif
	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur
	NA	TOR	Modif. Autorisée	Indique si le bloc est paramétrable par le menu générique de la face avant du M3

**Fonctionnement :**

Lorsque le compteur est en mode « Compte vers la présélection », la valeur initiale du compteur est 0 et la sortie s'active lorsque le compteur atteint la présélection. Lorsque le compteur est en mode « Décompte à partir de la présélection », la valeur initiale du compteur est égale à la présélection et la sortie s'active lorsque le compteur atteint 0.

En mode cycle unique, lorsque la sortie s'active, celle-ci le reste jusqu'à que la valeur de comptage passe en dessous de la présélection (ou au dessus de 0 en mode « décompte à partir de la présélection »), ou que le compteur soit réinitialisé.

En mode cycle répétitif, la sortie s'active durant la période définie par le paramètre « durée de l'impulsion » et la valeur courante de comptage est remise à 0 dès que la sortie est activée.

**Compteur (counter)**

Ce compteur permet d'activer une commande après un comptage prédéfini

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Comptage	Incrémente le compteur
	2	TOR	Décomptage	Décrémente le compteur
	3	TOR	Reset	Remet à zéro le compteur.
	4	TOR	Forçage présélection	Forçage de la valeur courante à la valeur de présélection.
	5	ANA	Présélection	Présélection du compteur
S	1	TOR	Sortie	S'active lorsque la valeur comptée atteint la présélection.

	2	ANA	Valeur courante	Donne la valeur courante du compteur
P	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur
<b>Fonctionnement :</b> La valeur initiale du compteur est 0 et la sortie s'active lorsque le compteur atteint la présélection en entrée. Lorsque l'entrée « Forçage présélection » est activée, la valeur de comptage est forcée à la valeur de présélection, et la sortie est activée.				

### Compteur horaire (H-meter)

Ce compteur permet d'activer une commande après un comptage prédéfini. Sans utiliser les consignes, il est très utile pour mesurer le temps de fonctionnement d'un appareil.

**Icône :**



**Caractéristiques :**

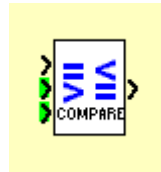
Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Commande	Active le compteur
	2	TOR	Reset	Remet à zéro le compteur.
S	1	TOR	Sortie	S'active lorsque le compteur atteint à la consigne
	3	ANA	Valeur courante Heure	Valeur courante des heures cumulée
	5	ANA	Valeur courante Minute	Valeur courante des minutes cumulée
PS	2	ANA	Présélection heures	Présélection des heures
	4	ANA	Présélection minutes	Présélection des minutes
P	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur
	NA	TOR	Modif. Autorisée	Indique si le bloc est paramétrable par le menu générique de la face avant du M3
<b>Fonctionnement :</b> Lorsque la commande est activée, ce bloc compte le temps (en heure et minutes). Une fois que le temps compté cumulé atteint la présélection, la sortie s'active jusqu'à ce que l'entrée reset soit activée.				

## Les Compérateurs

### Compérateur simple

Le but du compérateur est de comparer deux valeurs numériques

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Validation fonction	Valide la fonction
	2	ANA	Valeur 1	Valeur 1 à comparer
	3	ANA	Valeur 2	Valeur 2 à comparer
S	1	TOR	Sortie	Résultat de la comparaison
P	NA	NA	Type de comparaison	Choix de la comparaison entre : <ul style="list-style-type: none"> <li>- Strictement supérieur (&gt;)</li> <li>- Supérieur ou égal (<math>\geq</math>)</li> <li>- Egal (=)</li> <li>- Différent (<math>\neq</math>)</li> <li>- Inférieur ou égal (<math>\leq</math>)</li> <li>- Strictement inférieur (&lt;)</li> </ul>

**Fonctionnement :**

La sortie du bloc s'active si le résultat de la comparaison est vrai. Si la fonction est dé-validé (entrée Validation connecté et à OFF), sa sortie ne change pas d'état.

### Compérateur à fenêtre (compare in zone)

Le but du compérateur est de comparer une valeur numérique à deux autres, et d'indiquer si la valeur comparé se situe entre ou en dehors des deux autres valeurs.



**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Validation fonction	Valide la fonction
	2	ANA	Valeur à comparer	Valeur à comparer aux deux autres
	3	ANA	Valeur mini	Valeur minimum de la plage de comparaison
	4	ANA	Valeur maxi	Valeur maximum de la plage de comparaison
S	1	TOR	Sortie	Résultat de la comparaison
P	NA	NA	Mode	Permet de choisir si la sortie est active lorsque $V_{Min} \leq \text{Valeur} \leq V_{Max}$ , ou lorsque $\text{Valeur} < V_{Min}$ OU $\text{Valeur} > V_{Max}$

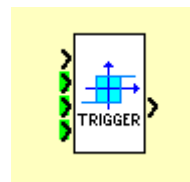
**Fonctionnement :**

La sortie du bloc devient active si  $V_{Min} \leq \text{Valeur} \leq V_{Max}$ , ou si  $\text{Valeur} < V_{Min}$  OU  $\text{Valeur} > V_{Max}$  en fonction du mode paramétré. Si la fonction est dé-validé (entrée Validation connecté et à OFF), sa sortie ne change pas d'état.

**Trigger de Schmitt**

Le but du trigger est de comparer une valeur numérique à deux autres, avec un fonctionnement par hystérésis.

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Validation fonction	Valide la fonction

	2	ANA	Valeur à comparer	Valeur à comparer aux deux autres
	3	ANA	Consigne de marche à arrêt	Seuil définissant le passage de marche à arrêt
	4	ANA	Consigne d'arrêt à marche	Seuil définissant le passage d'arrêt à marche
S	1	TOR	Sortie	Sortie de la fonction

**Fonctionnement :**

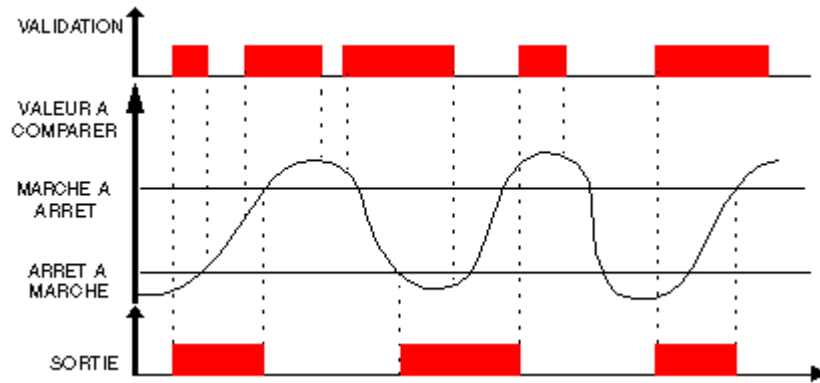
La sortie change d'état si :

- la valeur d'entrée est dépassée la valeur d'arrêt à marche
- la valeur d'entrée est passée en dessous de la valeur de marche à arrêt

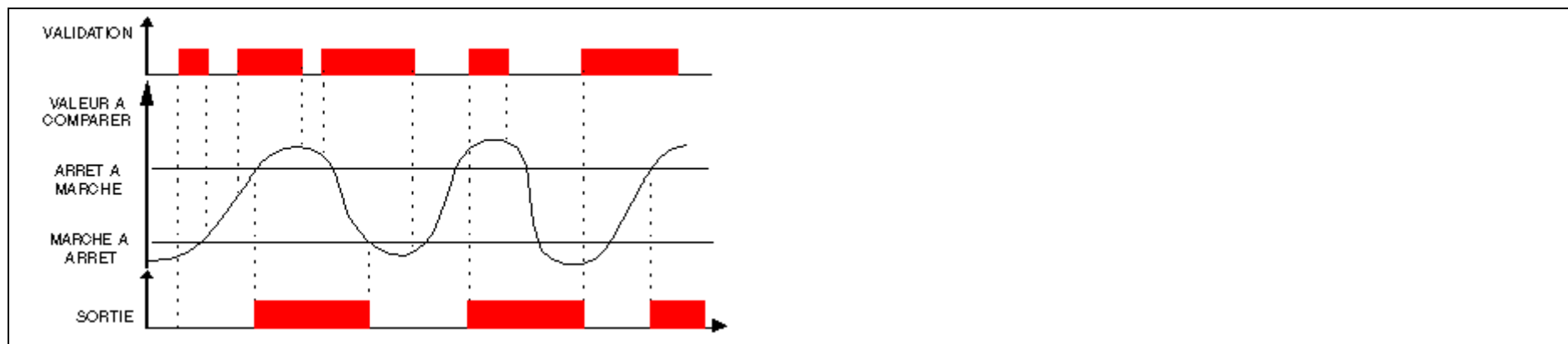
Si l'entrée est comprise entre les deux, la sortie ne change pas d'état.

Chacune des consignes DE MARCHÉ À ARRÊT et D'ARRÊT À MARCHÉ peuvent être aussi bien la valeur minimale ou la valeur maximale. Ceci implique un fonctionnement inversé de la fonction. Ces deux fonctionnements sont présentés sur les chronogrammes.

Cas où la **CONSIGNE DE MARCHÉ À ARRÊT** > à la **CONSIGNE D'ARRÊT À MARCHÉ** :



Cas où la **CONSIGNE D'ARRÊT À MARCHÉ** > **CONSIGNE DE MARCHÉ À ARRÊT** :

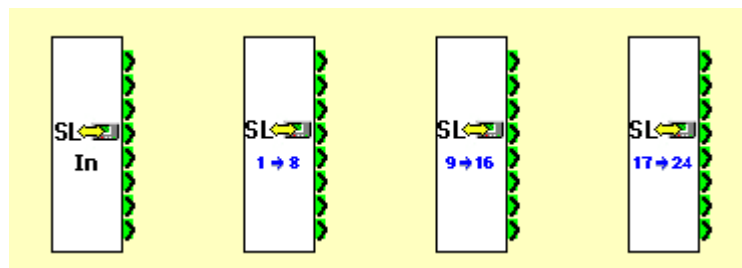


## Les liaisons série

### SL IN (entrées liaison série)

Le bloc SL IN permet d'injecter des valeurs dans le programme en utilisant la liaison série du Millennium 3.

**Icônes (variable selon le paramétrage) :**



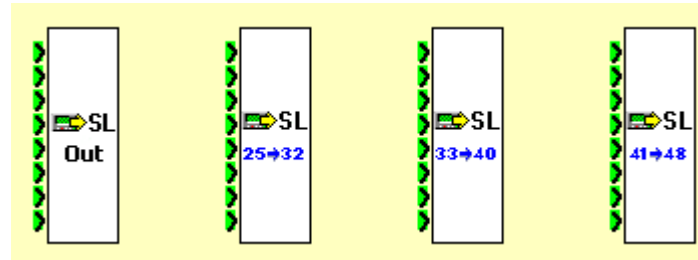
**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
S	1 à 8	ANA	Sorties 1 à 8	Valeurs issues de la communication par liaison série
P	NA	NA	Plage d'adresse du bloc	Configure la plage d'adresse du bloc (3 plages disponibles)
<b>Fonctionnement :</b>				
La fonction SL IN n'a aucun comportement spécifique, si ce n'est que ses sorties sont égales aux valeurs envoyées par un système externe utilisant la liaison série.				

## SL OUT (sorties liaison série)

Le bloc SL OUT permet de récupérer des valeurs du programme en utilisant la liaison série du Millenium 3.

**Icônes (variable selon le paramétrage) :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1 à 8	ANA	Entrées 1 à 8	Valeurs rendu accessibles via la communication par liaison série
P	NA	TOR	Plage d'adresse du bloc	Configure la plage d'adresse du bloc (3 plages disponibles)
<b>Fonctionnement :</b> La fonction SL OUT n'a aucun comportement spécifique, si ce n'est qu'elle met a disposition en lecture les valeurs connectées à ses entrées.				

## A propos des SL IN/SL OUT

Ces bloc n'ont aucune utilité dans un programme sauf si vous avez un logiciel/appareil capable de recevoir et/ou d'envoyer des valeurs via la liaison série. Le protocole de la liaison série est un dérivé du modbus avec quelques défauts en plus, principalement le mutisme du M3 si les trames envoyées ne sont pas valides. La base du protocole est donnée dans l'aide en ligne de l'atelier, et une note sur les problèmes courant est présente sur le site à [cet endroit](#).

## Les « Mémoires »

### Archives

Le bloc Archive permet de sauvegarder 2 valeurs analogiques tout en les datant.

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Mémorisation	Provoque la mémorisation des valeurs courantes de Valeur 1 et Valeur 2.
	2	TOR	Reset	Remet à zéro les valeurs mémorisées
	3	ANA	Valeur 1	Valeur 1 à mémoriser
	4	ANA	Valeur 2	Valeur 2 à mémoriser
S	1	TOR	Archive Valide	Indique si l'archive est valide
	2	ANA	Minute	Indique les minutes de la date de l'archivage
	3	ANA	Heure	Indique l'heure de la date de l'archivage
	4	ANA	Jour	Indique le jour de la date de l'archivage
	5	ANA	Mois	Indique le mois de la date de l'archivage
	6	ANA	Année	Indique l'année de la date de l'archivage
	7	ANA	Archive 1	Archive de la valeur 1
	8	ANA	Archive 2	Archive de la valeur 2
P	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur

**Fonctionnement :**

Ce bloc est très simple. A chaque impulsion de l'entrée mémorisation, le bloc sauvegarde la valeur courante de Valeur 1 et Valeur 2 dans les sorties Archive 1 et Archive 2. Les sorties donnant la date sont mises à jour à ce moment.

Tout est remis à 0 lors d'une impulsion sur l'entrée reset.

## Min/Max

Le bloc Min/Max permet de mémoriser les minimums et maximums atteint par une valeur.

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Reset	Remet les min/max mémorisé à la valeur courante de l'entrée valeur.
	2	ANA	Valeur	Valeur dont on veut mémoriser les min/max
S	1	ANA	Maximum	Maximum atteint par l'entrée valeur
	2	ANA	Minimum	Minimum atteint par l'entrée valeur
P	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur

**Fonctionnement :**

Le bloc mémorise automatiquement les valeurs min et max atteintes par l'entrée valeur. Une impulsion sur l'entrée reset remet les min/max mémorisé à la valeur courante de l'entrée valeur.

## Les Autres

### Bascule RS (Set/Reset)

La bascule RS est une sortie de mémoire de mise à 1 d'un signal TOR, avec une fonction reset.

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Set	Met la sortie à 1
	2	TOR	Reset	Met la sortie à 0
S	1	TOR	Sortie	Sortie du bloc
P	NA	NA	Priorité	Permet de définir laquelle des deux entrées est prioritaire sur l'autre.
	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur

#### Fonctionnement :

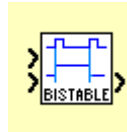
La sortie est égale à :

- 1 si l'entrée Set est active et que l'entrée reset n'est pas active
- 1 si l'entrée Set à été activée et que l'entrée reset n'a pas été activée
- 1 si l'entrée Set est active, que l'entrée reset est active, et que le paramètre de priorité est défini sur « Set prioritaire »
- Dans tout les autres cas, la sortie vaut 0 dès que l'entrée Reset est active ou à été activée sans que l'entrée Set n'ai été activée par la suite.

### Télérupteur (BISTABLE)

Le télérupteur fonctionne comme un télérupteur électrique. Son état change à chaque impulsion sur son entrée.

**Icône :**



### Caractéristiques :

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Commande	Met la sortie à 1 ou à 0 en fonction de son état précédent
	2	TOR	Reset	Met la sortie à 0
S	1	TOR	Sortie	Sortie du bloc
P	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur

#### Fonctionnement :

La sortie change de 0 à 1 ou de 1 à 0 à chaque impulsion de l'entrée commande.  
Une impulsion sur l'entrée Reset remet la sortie à 0.

### Multiplexeur (MUX)

Le multiplexeur permet de faire passer deux valeurs numériques vers une seule sortie. Ce bloc est rarement utile.

#### Icône :



### Caractéristiques :

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Sélection	Sélectionne la valeur à envoyer en sortie
	2	ANA	Voie B	Entrée valeur 1
	3	ANA	Voie A	Entrée valeur 2
S	1	ANA	Sortie	Prend la valeur de l'entrée Voie A ou de l'entrée Voie B en fonction de l'entrée sélection.



**Fonctionnement :**

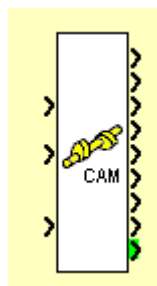
La sortie est égale à :

- l'entrée voie A lorsque sélection est a OFF
- l'entrée voie B lorsque sélection est a ON

**Arbre à Cam (CAM bloc)**

Le CAM Bloc permet de programmer l'état de ses sorties sous forme de séquence jusqu'à 50 cycles. Ce bloc très souvent utilisé en combinaison avec les blocs d'affichage pour gérer un système rudimentaire de menus.

**Icône :**



**Caractéristiques :**

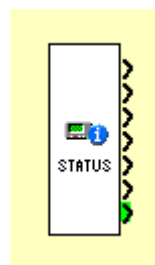
Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Avancer	Fait avance le CAM d'un cycle
	2	TOR	Reculer	Fait reculer le CAM d'un cycle
	3	TOR	Reset	Remet le CAM au cycle 0.
S	1 à 8	TOR	Sorties 1 à 8	Sorties correspondant aux états configurés dans les cycles.
	9	ANA	Position	Numéro du cycle courant.
P	NA	NA	Nombre de cycle et état de chaque cycle.	Le nombre de cycle permet de choisir le nombre total de cycle qui sera exécuté sur les 50 disponibles. Chaque cycle permet de configurer l'état des 8 sorties.
	NA	TOR	Sauvegarde sur coupure secteur	Indique si le bloc sauvegarde les valeurs courantes en cas de coupure secteur
	NA	TOR	Modif. Autorisée	Indique si le bloc est paramétrable par le menu générique de la face avant du M3 ( <b>note personnelle : modifier les cycles d'un cam</b>

				<b>bloc via la face avant du M3 relève simplement du pure masochisme)</b>
<b>Fonctionnement :</b> Pour chacun des cycles configurés, les sorties prennent l'état qui a été configuré dans le paramétrage. Chaque impulsion sur l'entrée « avancer » provoque le passage au cycle suivant (ou au premier cycle si le dernier a été atteint). Chaque impulsion sur l'entrée « reculer » provoque le passage au cycle précédent (ou au dernier cycle si il se trouve sur le premier).				

## Status

Ce bloc permet de récupérer des informations relatives à l'état du M3.

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
S	1	TOR	Etat Alarme	S'active si une alarme ou une erreur est détectée sur le M3 ou le programme.
	2	TOR	Marche monitoring	Indique si on est en mode monitoring
	3	TOR	Marche paramétrage	Indique si une modification de paramètres est en cours (via le mode monitoring ou via la face avant)
	4	TOR	Init à froid	Emet une impulsion à la mise en marche manuelle du programme
	5	TOR	Init à chaud	Emet une impulsion lors de la reprise du programme après une coupure secteur.
	6	TOR	Clignotant cycle	Change d'état à chaque cycle d'exécution du programme.
	7	ANA	Numéro alarme	Indique le numéro de l'alarme (voir sortie 1)
<b>Fonctionnement :</b> Pour de plus amples informations, se référer à l'aide en ligne (il est inutile que je fasse de la recopie pour ce bloc rarement utilisé)				

## Programmeur horaire (Time Prog)

Voilà un bloc tout à fait intéressant, et pour cause c'est un programmeur horaire très complet qui permet de déclencher des actions journalière, hebdomadaire, mensuel, annuel et ponctuel.

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Sortie	
P	NA	TOR	Modif. Autorisée	Indique si le bloc est paramétrable par le menu générique de la face avant du M3.
	NA	NA	Ensemble des taquets de programmation.	Voir la suite.

**Fonctionnement :**

Alors voilà un bloc dont le fonctionnement est très simple, mais dont le paramétrage est inversement proportionnellement compliqué.  
D'abord les captures d'écran des 2 onglets de paramétrage.

**Onglet paramètres :**

**TIME PROG (Programmeur horaire, hebdomadaire et annuel)**

Commentaires Paramètres Résumé

Heures: 0 Minutes: 0

MARCHE  ARRET

Hebdomadaire

Semaine(s): 1 2 3 4 5

Lu Ma Me Je Ve Sa Di

Journalier

Périodique

Date An: 10 (0...99)

Annuel Mois: 1 (1...12)

Mensuel Jour: 14 (1...31)

Cycle en cours: 00

Modification autorisée

Alors juste un mot de vocabulaire avant de commencer : Je vais parler de « Taquet » sans arrêt pour expliquer le paramétrage de cette fonction. Un taquet est un évènement MARCHE ou ARRET avec une heure et une périodicité.

C'est dans cette page que tout se configure. Le bouton « Nouveau » permet de créer un taquet (maximum 50), une fois qu'un taquet est créé il est possible de le paramétrer.

Dans la partie heure on va définir si c'est un taquet de mise à MARCHE ou à ARRET de la sortie, ainsi que son heure.

Dans la partie centrale on va configurer, dans le cas d'un évènement périodique, les semaines du mois ou il sera actif, ainsi que les jours de la semaine.

Pour un taquet journalier, il faut impérativement que les semaines 1 à 5 et que les 7 jours de la semaine soient cochés.

Pour un taquet hebdomadaire, il faut que les 5 semaines soient cochées et que 1 jour de la semaine soit coché.

Avec juste cette partie on a déjà de quoi faire, car on peut donc créer des évènements bi hebdomadaire, ou n autres variantes de périodicité.

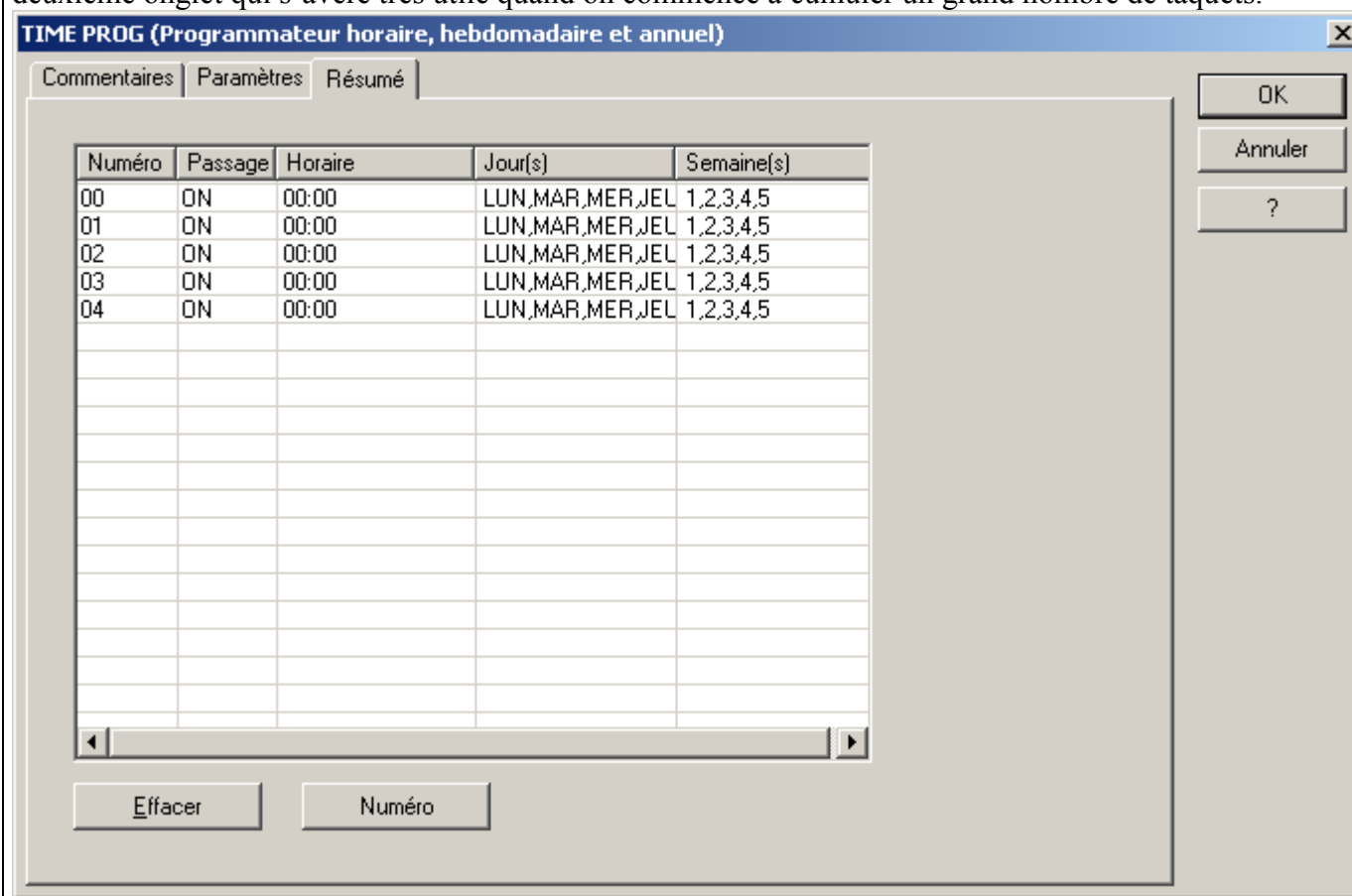
Une chose est importante à noter : Un TimeProg possède généralement un minimum de 2 taquets : un de mise en marche, et un de mise à arrêt. En effet, si aucun taquet « ARRET » n'est configuré, une fois la sortie activée par le taquet « MARCHE » celle-ci ne reviendra jamais à l'état ARRET (et pour cause, on ne le lui demande pas).

Le fait qu'il faille des taquets marche et arrêt permet de créer des sorties actives sur des plages horaire bien précises (par exemples les heures creuses EDF, ou je ne sais quoi d'autre du même style)

Ensuite il y a les taquets à date unique, pour cela il faut cliquer sur la case « Date » et configurer l'année, le mois, et le jour du mois.

En suivant ce qu'il y a dessous, vous pouvez configurer un taquet annuel (dans ce cas on ne configure plus que le mois et le jour du mois). En dernier lieu vous pouvez configurer un taquet mensuel en configurant uniquement le jour du mois.

Voilà un peu ce qu'on peut faire niveau configuration et vous l'aurez compris on peut vraiment configurer ceci comme on le souhaite. Voyons le deuxième onglet qui s'avère très utile quand on commence à cumuler un grand nombre de taquets.

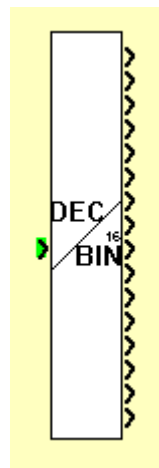


Pour l'exemple je n'ai pas configuré les taquets que j'ai créés, le but étant juste d'afficher une liste.  
 Nous avons donc sur cette page un résumé de tous les taquets configurés dans le TimeProg. En double cliquant sur l'un d'entre eux, vous revenez à la page de configuration du taquet cliqué.

### Convertisseur DEC=>BIN

Blablabla

**Icône :**



**Caractéristiques :**

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	ANA	Entier	Valeur que l'on souhaite convertir en bits
S	1 à 16	TOR	Bits 1 à 16	Bits issues de la valeur convertie

**Fonctionnement :**

Alors là, autant pour quelqu'un qui maîtrise le comptage en binaire, ce bloc paraît complètement trivial, autant pour un non initié, cela relève de l'improbable.

Je l'ai expliqué plusieurs fois, le M3 effectue ses calculs sur des entiers 16 bits signés. Il faut savoir que tout appareil numérique compte en binaire. Un ordinateur par exemple ne sait que manipuler des 1 et des 0, c'est le sens qu'on leur donne après qui fait que ce sont des caractères ou des chiffres (ou des images/vidéo, musique, etc.).

Pour comprendre comment compte un ordinateur (et donc aussi le M3), je vous invite à aller parcourir ces deux pages sur Wikipedia.

[Décimal codé en binaire](#)

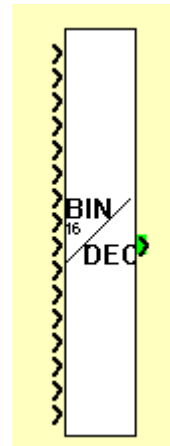
[Le complément à deux](#)

Dans tout les cas, ces blocs (dec/bin et bin/dec) sont rarement utiles, sauf si vous utilisez une supervision.

## Convertisseur BIN=>DEC

Blablaba

Icône :



Caractéristiques :

Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1 à 16	TOR	Bits 1 à 16	Bits à convertir en décimal
S	1	ANA	Entier	Valeur décimal issue de la conversion

**Fonctionnement :**

Voir le bloc DEC/BIN juste avant, le mécanisme étant le même mais en inversé.

## Les blocs FBD\_C

### Généralités

Alors, les FBD\_C qu'est ce que c'est ? La même chose que les FBD tout court nan ?

Alors ce n'est pas complètement faux, mais ce n'est pas complètement vrai non plus. Il faut savoir que tout les blocs présents dans les onglets autres que celui FBD\_C, sont des fonctions natives du firmware M3 (firmware = logiciel embarqué). Ceci signifie que lorsqu'on pose une fonction standard, le M3 sait déjà comment il doit l'exécuter.

Dans le cas des FBD\_C (aussi appelé fonctions métier) c'est un peu différent. En fait, il faut voir ces fonctions comme des logiciels supplémentaires qu'on peut installer dans le M3, avec bien sur une limite de mémoire (comme sur un PC, ou on ne peut plus rien installer si le disque dur est plein). Lorsque vous utilisez une de ces fonctions, et que vous chargez votre programme dans votre M3, l'atelier va détecter quel sont les fonctions métier déjà installé dans le M3, et il va ajouter ou supprimer ces « logiciels » en fonction de ce que contient votre programme.

### Comment connaît-on la place disponible pour installer des fonctions métier dans le M3 ?

Dans la phase 1, je vous avais présenté les barres d'outils et notamment la barre d'outil « contrôleur », souvenez vous de ça :



Lors de cette présentation je ne vous avais pas donné d'information sur cette jauge qui se trouve à gauche. Alors non, ceci ne donne pas, comme le veut la croyance populaire, la mémoire disponible pour votre programme, mais elle indique le nombre de slots fonction métier disponible dans l'automate. Cette jauge est automatiquement mise à jour chaque fois que ajoutez/supprimez une fonction métier dans votre programme. Un Millenium 3 extensible comme le XD26 possède 64 slots fonction métier. Une fonction métier peut consommer de 1 et 64 slots (dans les faits, la fonction la plus gourmande est le suivi solaire qui consomme à elle toute seul 56 slots).

Dernière chose à ce sujet, une fois que vous avez posé une fonction métier, vous pouvez la l'utiliser autant de fois que vous le souhaitez dans votre programme sans que cela consomme plus de slots. C'est comme un programme sur votre PC, vous l'installez une fois et vous pouvez le lancer autant de fois que vous le souhaitez.

### Les plus utiles

#### Super booléen

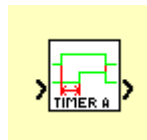
(Il a déjà été expliqué dans le chapitre sur les blocs logiques, voir aussi bloc BOOLEEN)



## Timer A

Sa fonction est similaire au TIMER AC déjà vu dans ce tutoriel, à la différence près qu'il possède beaucoup moins de sortie et un paramètre en moins, ce qui le rend plus économe en mémoire programme. Il est donc préconisé de l'utiliser si vous utilisez des timer AC, que sur ces derniers vous n'utilisez que la sortie TOR, et que vous n'utilisez pas le paramètre retard arrêt.

**Icône :**



**Caractéristiques :**

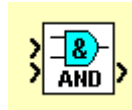
Catégorie	N°	Type (TOR/ANA)	Nom	Description
E	1	TOR	Commande	Active le timer lorsque cette entrée est validée (à ON). Le timer commence alors à compter. Si l'entrée est désactivée le comptage s'arrête et revient à 0
S	1	TOR	Sortie	S'active lors que la valeur courante retard marche atteint la consigne retard marche, et s'arrête lorsque la commande est désactivée.
P	NA	NA	Base de temps	Base de temps utilisé par le timer, peut être : <ul style="list-style-type: none"><li>- 1 seconde</li><li>- 100ms</li><li>- Période du programme</li></ul>
	NA	ANA	Consigne retard marche	Consigne de retard avant l'activation de la sortie du bloc.

**Fonctionnement :**  
Voire [Timer AC en mode A](#)

### Porte logique ET à deux entrées

Identique à la porte logique ET à 4 entrées. Permet d'économiser un tout petit peu de mémoire. A utiliser pour l'optimisation mémoire.

**Icône :**



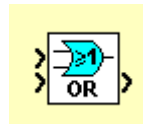
**Caractéristiques :**

NA

### Porte logique OU à deux entrées

Identique à la porte logique OU à 4 entrées. Permet d'économiser un tout petit peu de mémoire. A utiliser pour l'optimisation mémoire.

**Icône :**



**Caractéristiques :**

NA

## Les blocs SFC

Alors ici c'est une série de bloc dont vous aurez rarement besoin, mais sachez que leur utilisation fait appel à des notions de programmation séquentielle qu'il faut avoir acquises pour les utiliser correctement. Si vous connaissez le GRAFCET, ces blocs vous paraîtront plus que familiers.

Sachez aussi que les SFC sont les seuls types de bloc qui s'ils sont mal assemblés, peuvent provoquer des erreurs de compilation de votre programme, et ce sont aussi les seuls dans l'atelier M3 à avoir des entrées à gauche **et** au dessus, et des sorties à droite **et** en bas (oui oui ça se complique)

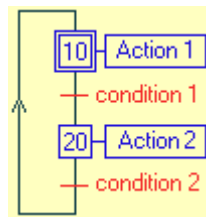
Pour ceux qui ont déjà peur avec ce que j'ai dit avant, je vais tout de même expliquer quelques bases afin que vous ayez la possibilité de comprendre ce qui va suivre.

Dans tout les cas, même si vous ne comprenez pas, aucun problème car les graphes SFC n'ont pas un grand intérêt dans le cas des régulations solaire (encore qu'on pourrait voir un intérêt pour la séquence de fonctionnement d'une installation en drain back)

Alors ce qu'il faut savoir :

Le **GRAFCET** est un acronyme qui signifie *GR*A*p*he *F*onctionnel de *C*ommande *E*tapes/*T*ransitions.

Commençons par interpréter cela et particulièrement les 2 derniers mots ***E*tapes/*T*ransitions**. Un GRAFCET (i.e. graphe SFC) est composé d'étapes, lesquelles sont séparées par des transitions. Voilà comment cela se représente.



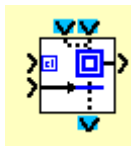
Dans cette image, les étapes sont les carrés contenant un numéro, et les transitions sont les traits rouges entre les étapes. A chaque étape est liée une action (le rectangle bleu à côté des étapes) qui est la commande (cela peut être la mise en marche d'un moteur, incrément d'un compteur, ou plein d'autres choses)

Le GRAFCET possède toujours une étape initiale qui permet de placer l'ensemble de l'installation qu'il pilote dans sa position de départ (reset des compteurs, arrêt de moteur, mise en place d'une pièce, etc). Cette étape est représentée par un double carré (ici l'étape numéro 10)

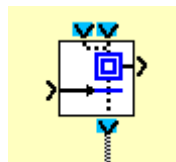
Le GRAFCET se lit et s'exécute de haut en bas, et le passage d'étape en étape est conditionné par la validation des transitions.

### ***L'étape initiale, et l'étape initiale ré-initialisable***

Cette étape est OBLIGATOIRE et doit être UNIQUE dans chaque graphe SFC. Elle représente le point de départ de votre graphe, c'est elle qui sera active au démarrage du programme.



(Étape initiale ré-initialisable)

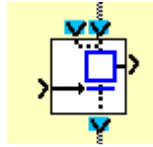


(Étape initiale)

Comme toute étape SFC, l'étape initiale possède une entrée transition, et une sortie commande. Dans le cas de l'étape initiale ré-initialisable, elle possède une entrée init qui permet de forcer le graphe SFC à revenir à l'étape initiale.

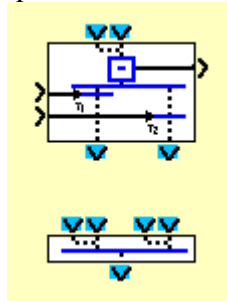
### **L'étape simple**

C'est l'étape standard du SFC, elle est activée par validation de la transition de l'étape qui la précède, et elle se termine une fois que sa transition est validée.



### **La divergence en OU et la convergence en OU**

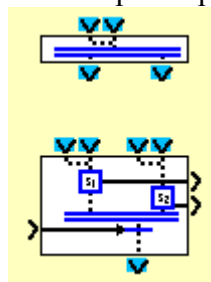
Ces blocs SFC permettent de faire diverger le graphe vers une branche ou une autre. Les deux sous branches seront composées de n'importe quelle fonction SFC (sauf étape initiale)



La divergence en OU permet d'adopter des actions différentes en fonction des transitions obtenues. Par exemple, imaginons une machine qui trie des pièces. La vérification de chaque pièce est faite en plusieurs étapes. Si on imagine le GRAFCET de la machine de vérification, on pourrait y voir une divergence en OU à chacune des étapes de la vérification. Si la pièce testée est valide, on continue sur le graphe normal, si elle n'est pas valide, on part vers un graphe qui éjecte la pièce et qui retourne ensuite à l'étape initiale.

### **La divergence en ET et la convergence en ET**

Ces blocs SFC permettent de faire diverger le graphe vers deux branches en même temps. Les deux sous branches seront composées de n'importe quelle fonction SFC (sauf étape initiale)



La divergence en ET permet de lancer l'exécution de deux actions en même temps. Si on imagine un robot qui assemble la caisse d'une voiture, on pourrait bien imaginer que le robot après avoir installé le capot, il installe les deux portières simultanément.

### **Un graphe SFC « simple »**

Alors si en SFC y'a pas des masses monstrueuses de bloc, selon l'assemblage qu'on en fera, on obtient des tas de choses différentes.

Pour l'exemple, je vous propose d'étudier le cas d'une installation en drain back.

Si on met ça en forme façon GRAFCET, voila un peut les étapes et transition qu'on a :

(Pour des raisons pratiques que vais noter les étapes **En**, et les transitions **Tn**, **n** étant un numéro)  
Une dernière petite chose, dans un graphe SFC, il est courant qu'une étape ne provoque aucune action, dans ce cas ces étapes servent uniquement à attendre la validation de leur transition.

### **Départ**

E0 (initial) : le circuit est vide, le capteur est froid, et la pompe arrêtée.

(T0): le capteur chauffé

E1 : on met en marche la pompe

T1 : détection du bon remplissage du circuit

E2 : attente de 2 minutes (temps arbitraire de remplissage avec un peu de circulation)

T2 : fin de la temporisation (il existe dans les fonctions FBD\_C une étape SFC attente qui fait cela toute seule, mais cela se fait aussi très bien avec un timer)

E3 : rien

T3 : la température du caloporteur est  $>$  à celle du bas du ballon

T3' (divergence) : la température du caloporteur est  $<$  à celle du bas du ballon

(La condition T3' est la négation (le NOT logique) de T3)

NOTE : j'ai introduit une divergence ici car un capteur en stagnation chauffe très vite. Un faible rayonnement peut suffire à le faire monter en température à vide, mais pas forcément avec de l'eau qui y circule

Donc nous avons deux branches, une où l'on peut commencer la chauffe, et l'autre où on va mettre en place une « attente » pour voir si la température ne finira pas par monter un peu.

### **Branche normale :**

E4 : rien (la pompe est en marche, on chauffe le stock)

T4 :  $T^{\circ}$  caloporteur  $<$  à  $T^{\circ}$  bas ballon (on arrête si on a plus suffisamment d'apport solaire)

E5 : Arrêt de la pompe

T5 : détection de la bonne vidange du circuit

→ Retour à T0.

### **Branche attente :**

E6 : incrément d'un compteur

T6 : E6 activée

E7 : rien

T7 : valeur max du compteur atteinte

T7' : valeur max du compteur NON atteinte

(pfïou encore une divergence)

### **Branche sur T7'**

E8 : attente de 1 minute

T8 : fin de temporisation

→ Retour à E3

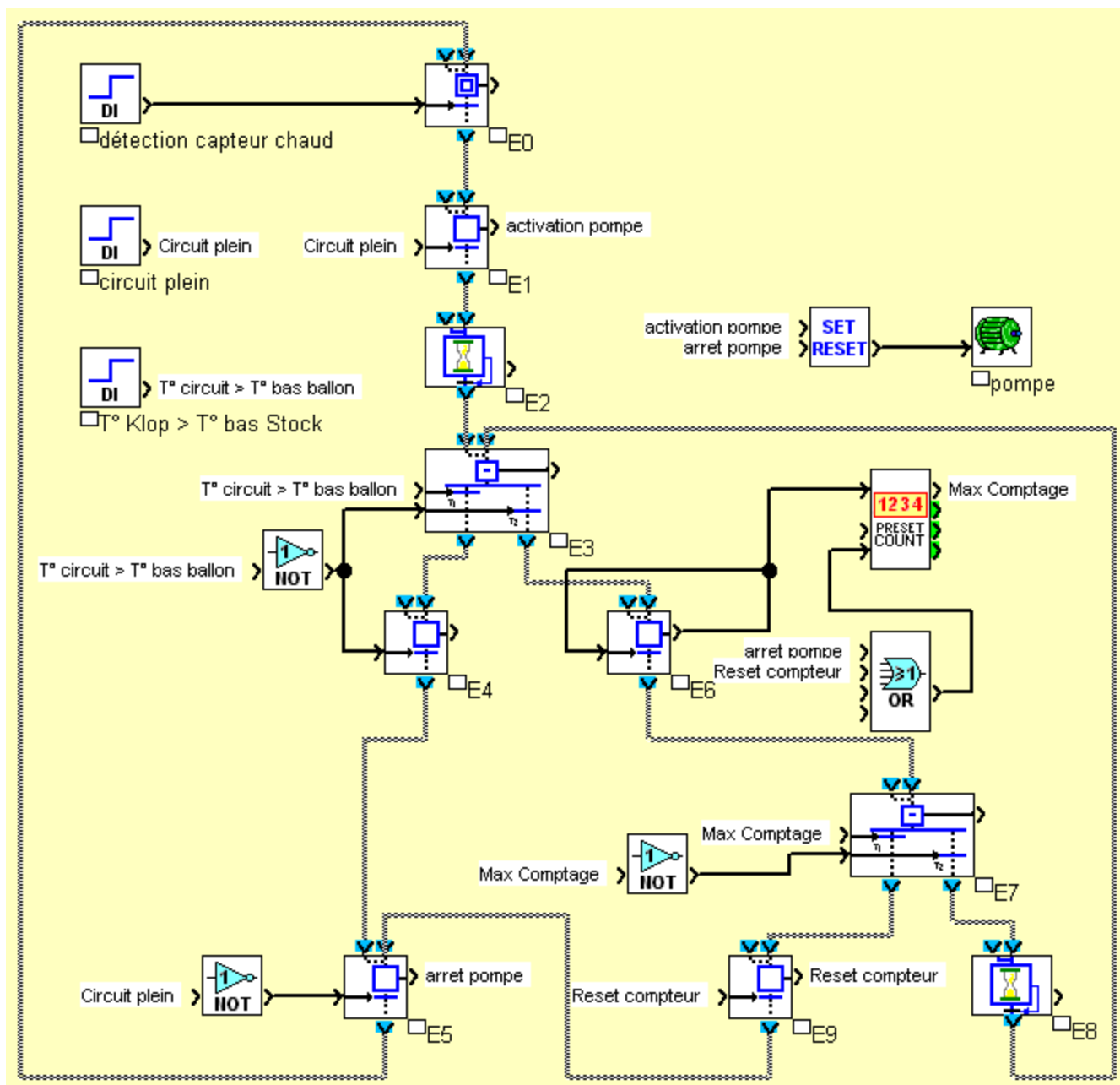
### **Branche sur T7 (compteur atteint → sortie de la boucle d'attente)**

E9 : reset du compteur

T9 : E9 activé

→ Retour à E5

Voilà un peu ce que cela donne sur papier... voyons ce que cela donne en programme dans l'atelier M3 (pour simplifier le programme, j'ai utilisé de simples entrées TOR pour gérer certaines transitions, dans un programme réel on aurait une gestion des deltas de températures avec des triggers).



Alors j'ai tout de même pris la peine de commenter un peu les blocs pour au moins repérer les différentes étapes du graphe SFC, ce qui vous permet de vous repérer entre ce que j'ai exprimé sous forme de texte, et le résultat.

Il y a toute fois une lacune dans ce graphe, c'est le cas où l'on serait en chauffe mais où le soleil serait masqué temporairement provoquant un refroidissement du caloporteur. Ceci aurait pour effet d'arrêter la pompe, et de revenir au point de départ E0, alors que dans l'idéal il faudrait faire une attente comme au démarrage (boucle de droites avec E6, E7, E8, et E9), ce qui permettrait de ne pas arrêter le circuit dès que le soleil joue à cache-cache.

### Conclusion sur le SFC :

Alors le SFC est très puissant si on doit piloter quelque chose de très séquentiel avec peu de possibilité de branches différentes. Dès lors qu'on commence à cumuler les divergences, il faut le dire c'est un vrai métier, et ce n'est pas le mieux adapté à la régulation d'une installation solaire. Imaginez un graphe où on gère : ECS, chauffage, et surchauffe... le graphe deviendrait sûrement imbuvable, en tout cas bien moins bien qu'un programme sans SFC.