

Le Millenium 3 pour les nuls !!

Phase 1 : commencer

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

Introduction :

Le millenium 3 est un petit automate programmable très performant et très accessible...pour peu qu'on sache le programmer. Pour les personnes travaillant dans des métiers liés à l'informatique, la programmation de cette bestiole est un vrai jeu d'enfant, mais pour une personne lambda, cela est nettement moins évident.

C'est ici que ce tutoriel intervient, car il a pour but de rendre accessible et compréhensible la programmation du M3 au plus grand nombre.

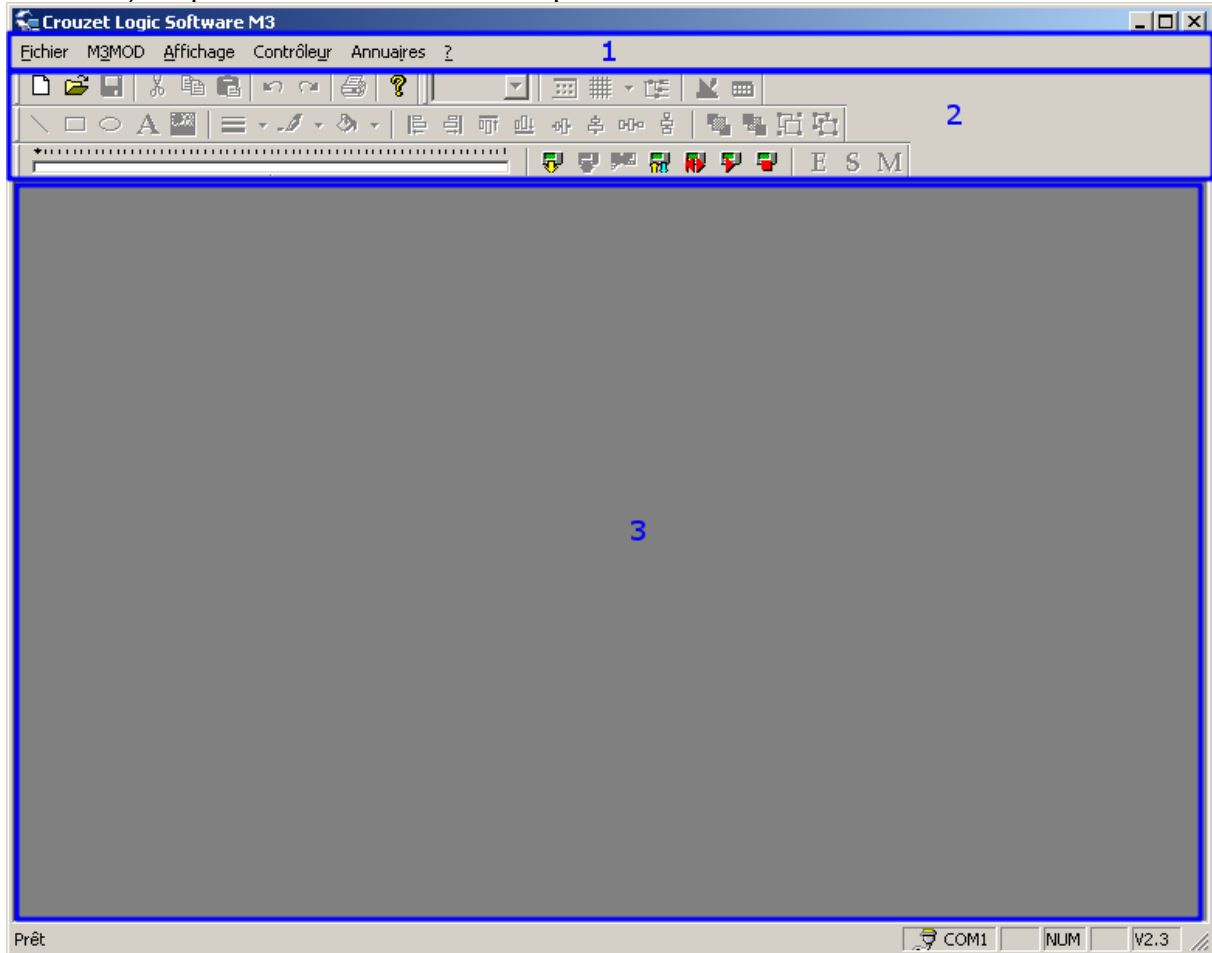
Ce tutoriel en plusieurs parties explique donc le fonctionnement de l'atelier de programmation Millenium3

Table des matières

<i>Aspect général de l'atelier de programmation.....</i>	<i>3</i>
<i>Créer un nouveau projet.....</i>	<i>6</i>
Le choix du contrôleur.....	6
Le choix des extensions.....	8
Le choix du langage.....	9
<i>Les différentes fenêtres.....</i>	<i>10</i>
La surface de câblage.....	11
La fenêtre de supervision.....	12
<i>La programmation et le câblage.....</i>	<i>13</i>
Ajouter un bloc dans la surface de câblage.....	13
La sélection d'un bloc.....	14
La sélection Multiple.....	14
Le déplacement Multiple.....	15
Tracer des liens.....	16
Les types d'entrées et de sorties des blocs.....	16
Le premier lien.....	17
L'aspect des liens filaires et leurs positionnements.....	17
Les liens textes.....	20
<i>Premier programme et Paramétrage d'un bloc fonction.....</i>	<i>23</i>
Le télérupteur.....	23
Le Thermostat d'ambiance.....	27
<i>La simulation.....</i>	<i>31</i>
Passer en simulation : les changements dans l'atelier.....	32
Les forçages fugitifs.....	33
Ralentir la simulation.....	35
Les forçages maintenus.....	38
La simulation des entrées analogiques.....	39
<i>L'utilisation avancée de la souris et des raccourcis clavier.....</i>	<i>41</i>
Dupliquer un bloc.....	41
Sélection multiple avec MAJ.....	43
Couper, Copier, Coller.....	43

Aspect général de l'atelier de programmation

Voici sous quelle forme se présente l'atelier de programmation M3 lorsqu'on le démarre. Commençons par identifier les différentes parties :



1 : la barre de menu

Cette barre est présente dans toutes les applications standards sur à peu près tout les OS (Windows, Mac OS, Linux). Elle contient souvent certains menus récurrents, comme le menu « fichier », « affichage », et « ? ».

Dans l'atelier M3, le menu a des états variables qui dépendent de ce qu'on est en train de faire dans l'atelier. Ici elle est présentée lorsqu'aucun projet n'est ouvert.

2 : les barres d'outils

Cette zone contient toutes les barres d'outils de l'application. Dans le cas de l'atelier M3, les barres d'outils peuvent être déplacées à volonté afin d'adapter leur positions à vos préférences. D'une manière générale sous Windows, les barres d'outils contiennent de nombreux raccourcis vers des commandes qui existent dans les menus.

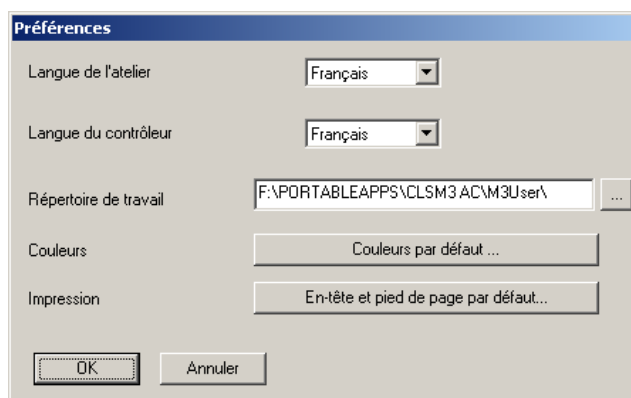
3 : le conteneur de fenêtre

Cette zone contient différents types de fenêtres en fonction de ce qu'on fait dans l'atelier. Lorsqu'aucun projet n'est ouvert, elle reste vide. Dans cette zone on peut déplacer, minimiser, ou maximiser toutes les fenêtres présentes à notre guise.

Pour l'instant je ne donne aucun détails quand au contenu, car ceci sera expliqué de façon plus détaillé en fonction des situations rencontrées lors de la programmation.

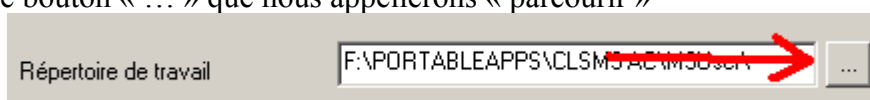
Quelques généralités sur les moyens que je vais utiliser pour m'exprimer dans le tutoriel :

Par exemple, si je dis : « Allez dans **Fichier**→**Préférences** », cela fait référence à la commande « **Préférences** » présente dans le menu « **Fichier** », faites l'essai, cliquez sur le menu « **Fichier** », et déplacer votre souris jusque sur la commande « **Préférences** » puis cliquez dessus. La page suivante s'affiche

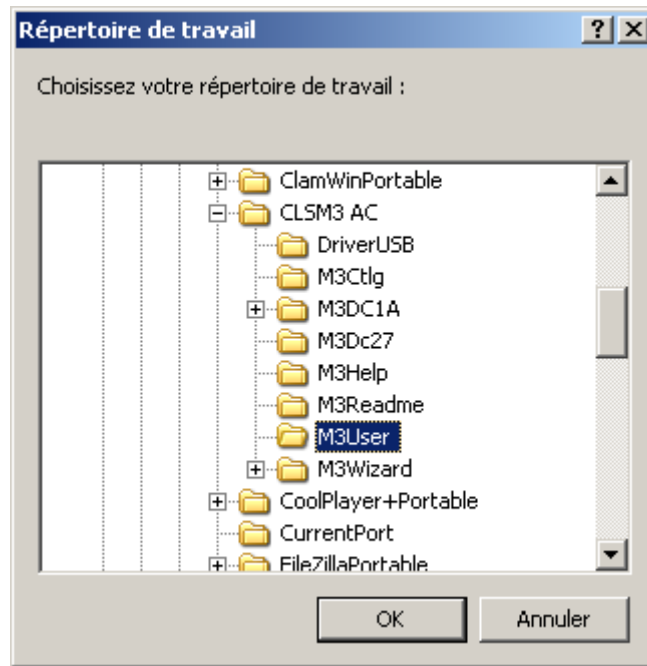


Configurer votre espace de travail

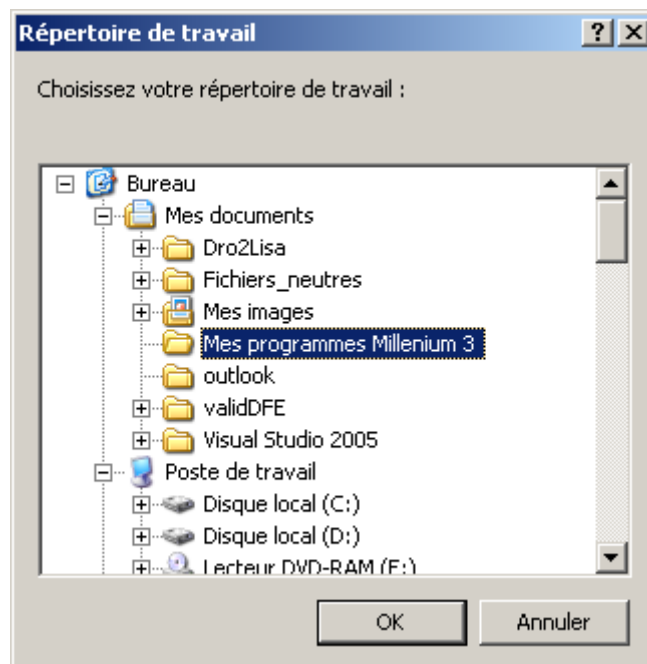
Dans les lignes précédentes, je n'ai pas choisis le menu préférences pour rien. En effet il y a une petite chose qu'il est intéressant de régler avant de commencer à travailler, c'est votre répertoire de travail. Le but de cette opération est de vous permettre de ranger tous vos programmes à un endroit où vous serez capable de les retrouver, car par défaut l'atelier de programmation va les stocker dans son répertoire d'installation qui est par défaut : « C:\Program files\Crouzet Automatismes\CLSM3 AC\M3User ». Vous l'aurez remarqué, ce dossier, n'est pas celui qu'on a envie d'aller fouiller tous les jours pour envoyer ses programmes aux copains du forum APPER et nous allons donc le changer. Cliquez sur le bouton « ... » que nous appellerons « parcourir »



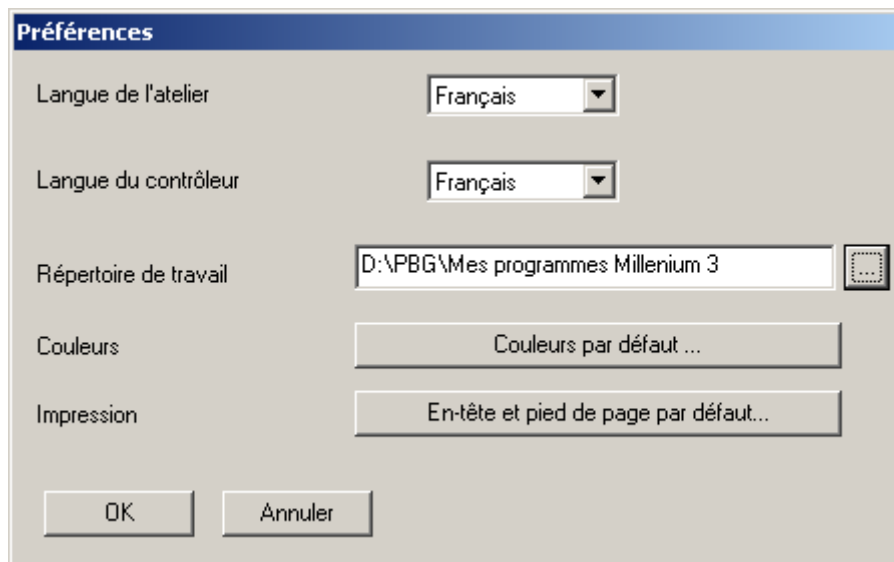
La fenêtre suivante s'affiche :



Avec cette fenêtre, vous allez pouvoir choisir dans quel dossier vous allez travailler. Pour faire simple, nous allons dans cette exemple choisir le dossier « Mes Documents », mais libre à vous de choisir un autre endroit, à partir du moment où vous êtes capables de le retrouver simplement. Commencez par aller dans votre dossier « Mes document » avec l'explorateur Windows, et créez un dossier pour vos programmes (on va essayer de rester organisé) puis à l'aide la barre latérale, remontez dans la liste et trouvez l'élément « Mes Documents » et cliquez sur le petit « + » à coté de celui-ci. Ceci affiche alors la liste des dossiers présents dans vos documents. Moi j'ai crée un dossier qui s'appelle « Mes programmes Millenium 3 » et je l'ai sélectionné.



Une fois votre dossier choisi (en surbrillance), cliquez sur le bouton « OK » pour revenir à la page des préférences.



Vous remarquerez que le « Répertoire de travail » a changé et pointe maintenant vers le dossier que vous avez choisi. Cliquer sur « OK » pour enregistrer ce changement.

Bon avec ça, votre environnement de travail est prêt, et on va pouvoir se jeter à l'eau pour de vrai (ne prenez pas votre PC avec vous...il n'aime pas l'eau).

Créer un nouveau projet

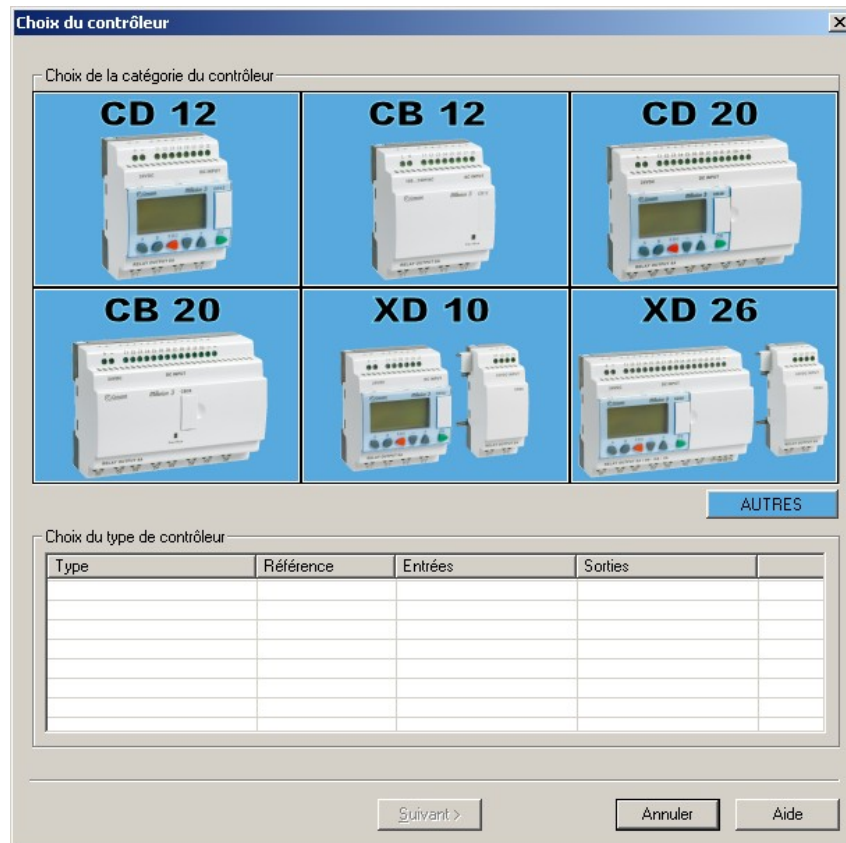
Pour créer un nouveau projet, cliquer sur le menu « **Fichier**→**Nouveau** », ou sur le bouton



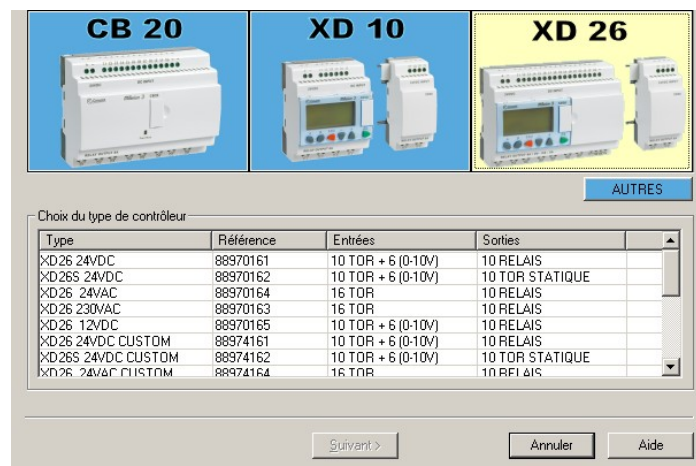
« Nouveau » de la barre d'outils, ou encore pour les aficionados du clavier, utiliser le raccourci standard « Ctrl + N ».

Le choix du contrôleur

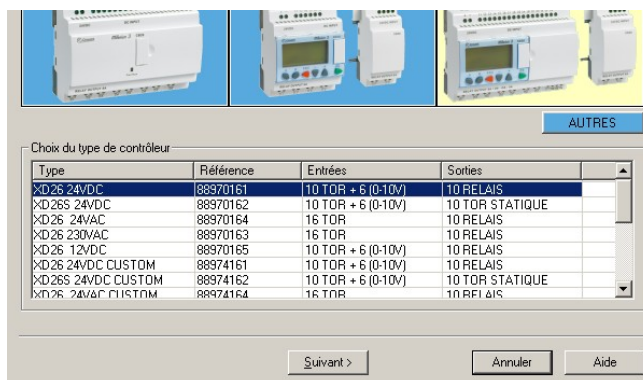
La fenêtre suivante s'affiche :



Dans cette page, vous devez choisir quel type de M3 vous utilisez. Pour faire simple, tous les M3 vendu par l'intermédiaire du groupement d'achat APPER sont des **XD 26**. Cliquez donc sur la grosse image XD26. La liste de choix située en bas se remplit comme présenté dans la capture suivante.



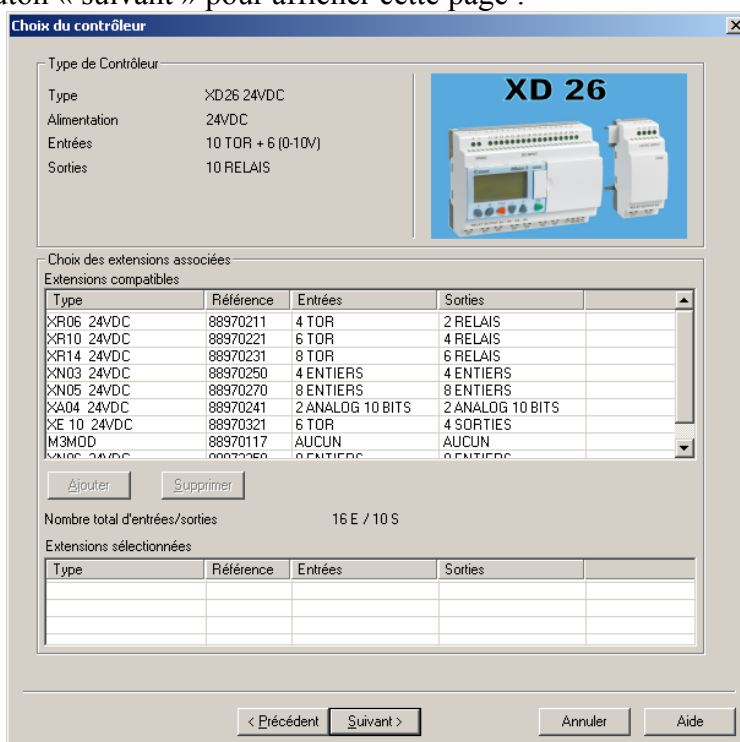
Alors dans cette liste il va falloir sélectionner le modèle que vous avez commandé et qui crie du fond de son carton « Programme-moi !! Programme-moi !! ». Si vous n'avez pas acheté un modèle « Adapté », votre M3 est le premier de la liste. Si vous avez un autre M3, il vous suffit simplement de rechercher dans la liste la référence qui correspond à votre M3 (cette référence est écrite sur le coté droit de votre M3). Une fois que vous avez sélectionné votre M3, le bouton suivant devient disponible.



Avant de continuer, je voudrais juste faire une petite parenthèse sur le bouton « Autres » présent juste en dessous de l'image des XD26. Si vous cliquez sur ce bouton, la liste contiendra les M3 un peu spéciaux comme les cartes nues ou résinées (voir le [document à propos du choix de votre M3](#))

Le choix des extensions

Cliquez sur le bouton « suivant » pour afficher cette page :



Dans cette page, vous pouvez ajouter les extensions que vous avez installées sur votre M3. Si vous n'avez pas d'extension, vous pouvez simplement cliquer sur le bouton « Suivant » en bas de la page.

Pour choisir vos extensions, le principe est le même que précédemment, on recherche par le nom, ou par la référence. ATTENTION, il n'est pas possible d'ajouter plus de deux extensions (sauf pour le M3MOD, voir le document à propos du choix de votre M3). Pour ajouter une extension, sélectionnez la dans la liste du dessus, et cliquer sur le bouton « Ajouter », ou double cliquez directement sur l'extension que vous voulez ajouter. Les extensions ajoutées sont affichées dans la liste du bas. Si vous avez ajouté une mauvaise

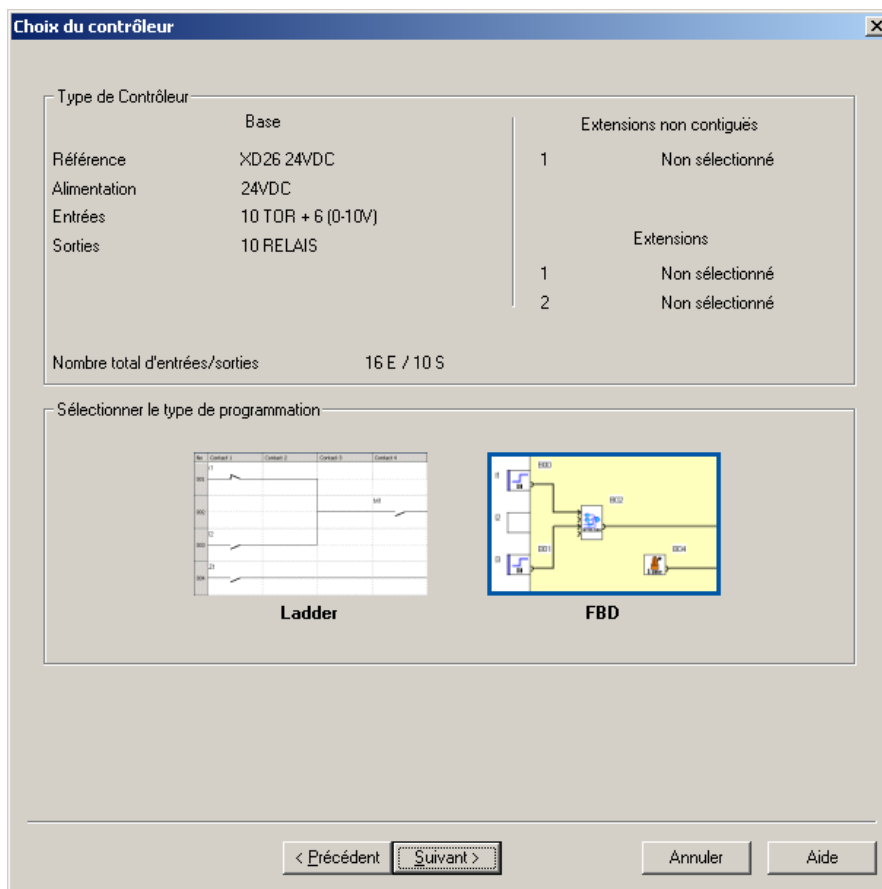
extension, sélectionnez-la simplement dans la liste du bas, et cliquez sur le bouton « Supprimer », ou double cliquez sur l'extension à supprimer.

NOTE : l'atelier M3 est sympa, car il vous donne la possibilité de changer ultérieurement votre configuration, sans pour autant casser votre programme, ceci sera expliqué plus tard.

Le choix du langage

Nous voici sur la dernière page avant de commencer à pouvoir programmer. Dans cette page il est possible de choisir entre deux types de langages :

- Le FBD (Function Bloc Diagram)
- Le Ladder



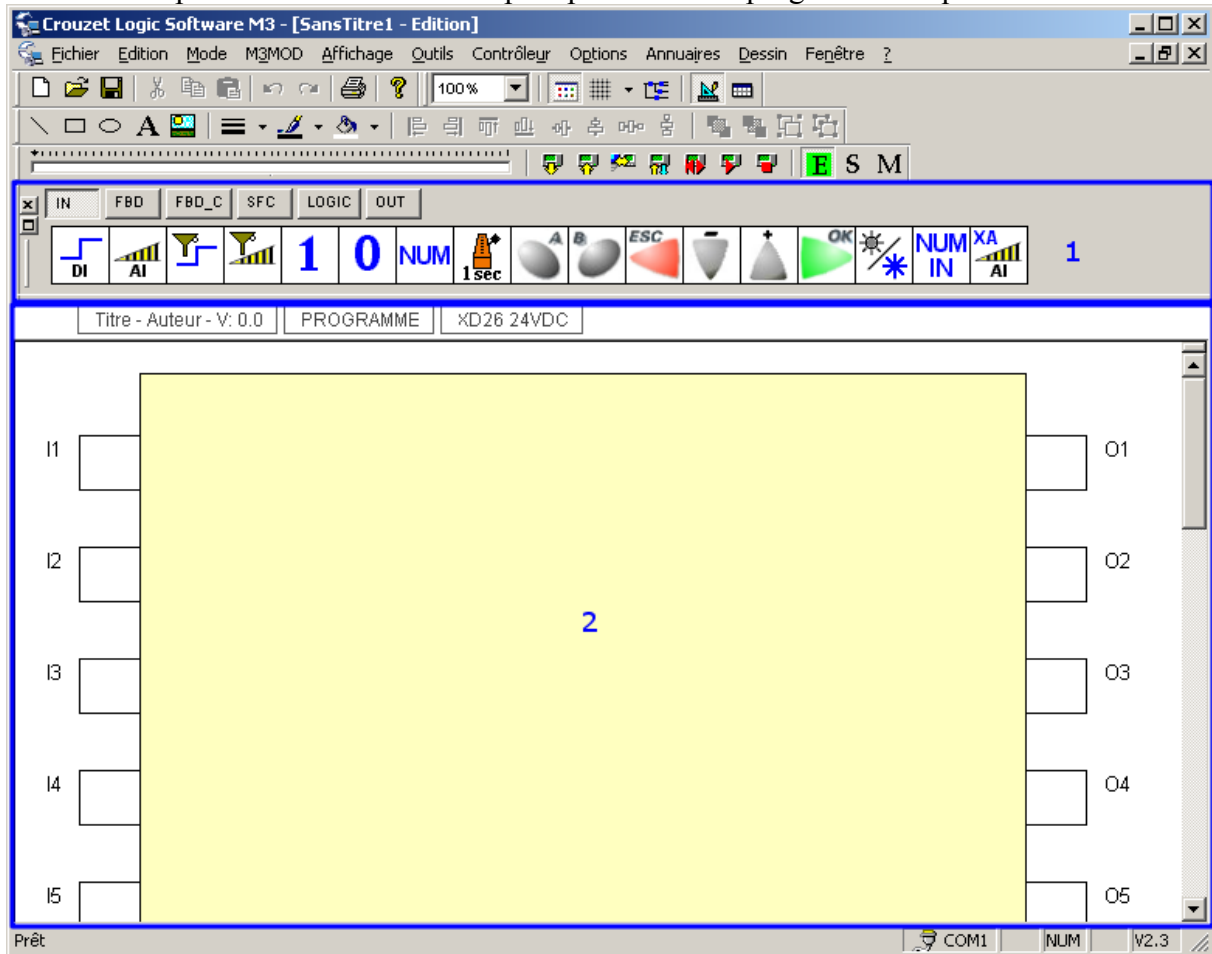
Autant le dire tout de suite, je ne présenterai pas le langage Ladder, car celui est nettement moins puissant et compréhensible que le FBD (enfin c'est mon point de vue), vous n'avez donc rien à changer sur cette page.

Pour ce qui est du contenu de cette page, la partie haute est un résumé de la configuration. A gauche vous avez le type de M3 que vous avez choisi, et à droite les éventuelles extensions ajoutées.

Cliquez sur le bouton suivant

Les différentes fenêtres

Nous voilà à présent avec notre atelier paré pour créer un programme et qui ressemble à ceci



Votre perspicacité vous aura fait remarquer que beaucoup de choses nouvelles sont apparues, commençons par les détailler.

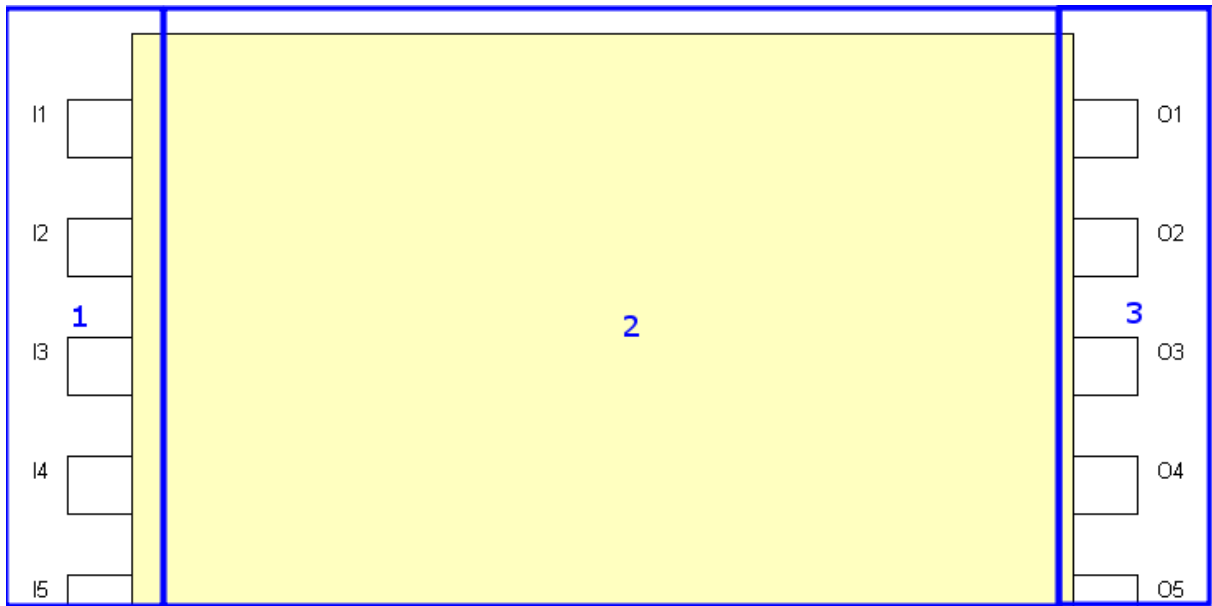
1 : Barre de fonction

C'est dans cette barre d'outils un peu spéciale que vous trouverez l'ensemble des fonctions standard du M3, elle se décompose en 6 onglets : IN, FBD, FBD_C, SFC, LOGIC, OUT. Toutes les fonctions seront détaillées dans le document « le Millenium 3 pour les nuls : phase 2 ». Cette barre de fonction reste affichée tant que l'on reste dans le mode édition de l'application, les autres modes seront abordés ultérieurement.

2 : la surface de câblage :

Voir chapitre qui suit.

La surface de câblage



C'est dans cette zone que l'on va programmer en posant des blocs fonctions, en les paramétrant, et en créant des liens.

Cette partie se découpe en 3 zones :

1 : Les entrées

Du côté gauche on retrouve des emplacements numérotés de forme carré qui représentent les entrées physiques du M3, dans chacun de ces emplacements il va être possible d'y poser un bloc fonction de type « IN ».

2 : La zone de câblage

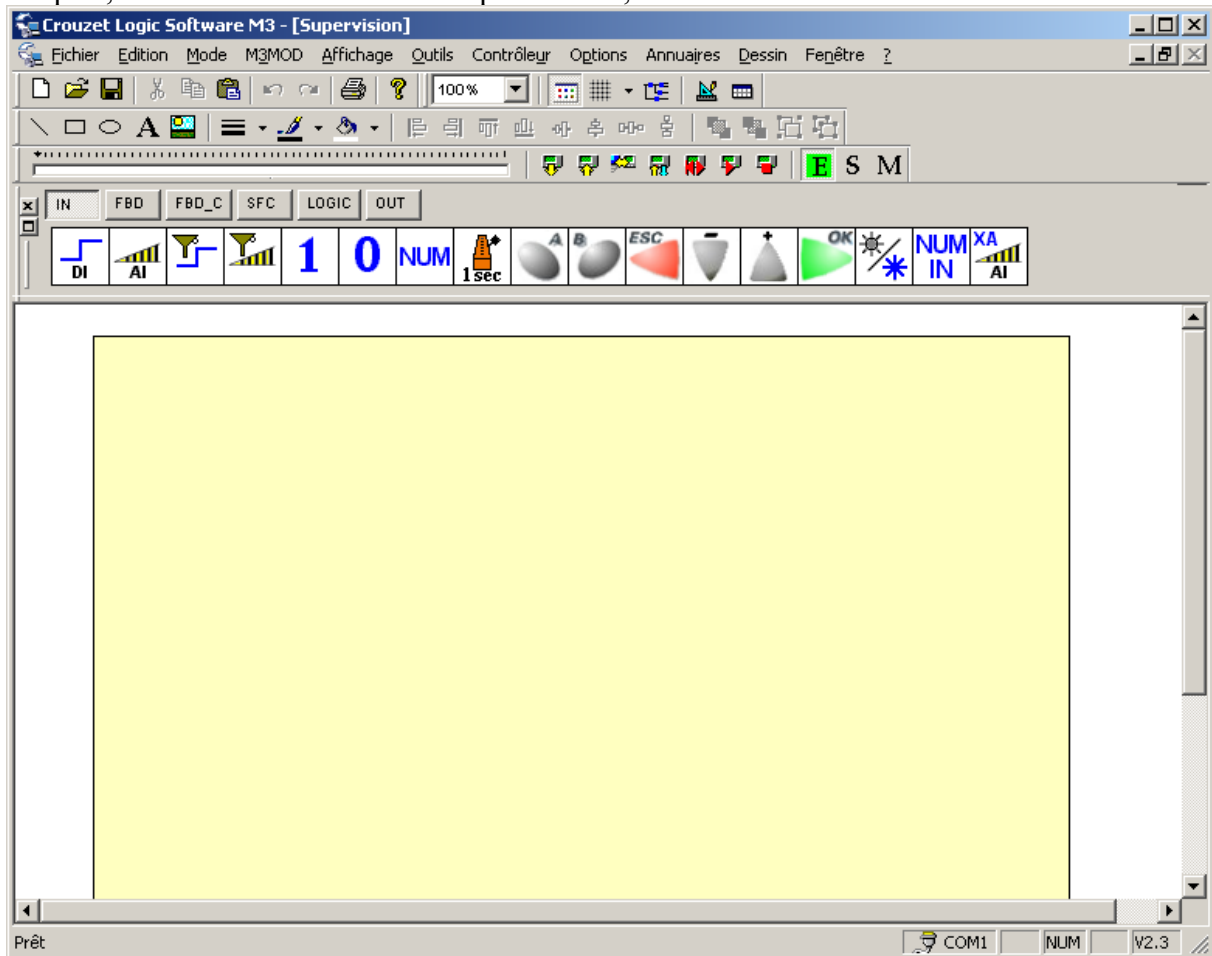
C'est dans cette partie que nous allons créer le programme en y déposant divers blocs et en les liant entre eux.

3 : Les sorties

Du côté droit, on retrouve des emplacements numérotés de forme carré qui représentent les sorties physiques du M3, dans chacun de ces emplacements il va être possible d'y poser un bloc fonction de type « OUT ».

La fenêtre de supervision

Cliquez, dans le menu « fenêtre→Supervision », la fenêtre suivante est alors affichée.



Dans cette fenêtre vous ne pouvez pas poser de bloc depuis la barre de fonction, ni tracer de lien. En revanche vous pouvez copier/coller des blocs que vous avez déjà posé dans la feuille de câblage principale. L'utilisation de cette fenêtre sera plus détaillée lorsque nous verrons la simulation d'un programme.

Revenez à la surface de câblage via le menu « Fenêtre→*NomDeProgramme* – Edition » (j'ai mis le nom en italique car le titre de cette fenêtre change en fonction du nom de votre programme par défaut il est à « *SansTitreN* »)

La programmation et le câblage

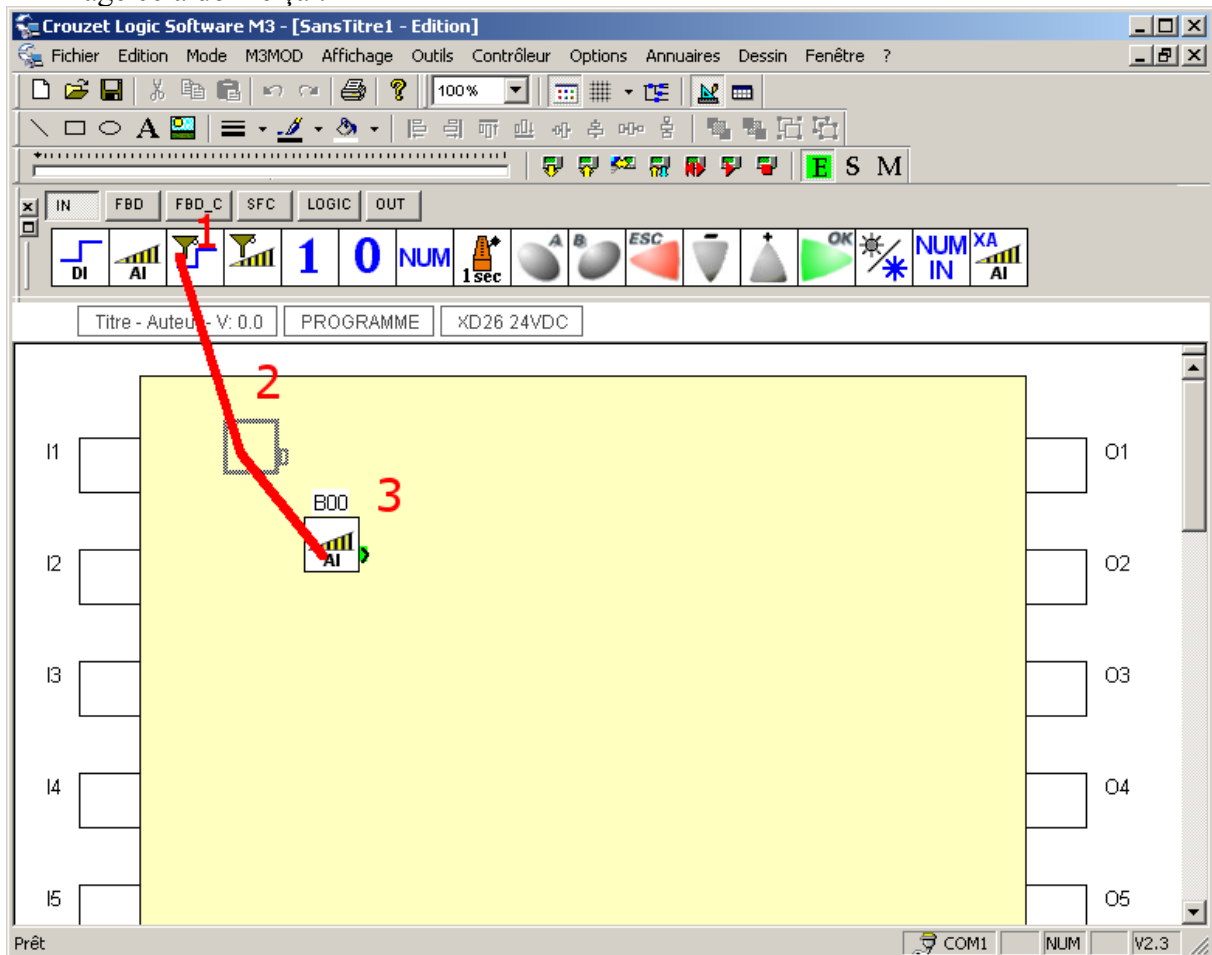
A partir de maintenant, on entame les choses intéressantes, la programmation. Pour l'utilisateur lambda du PC, ce mot peut faire très peur, et c'est pourquoi ce tutoriel existe. Avant de vraiment programmer et de comprendre ce que l'on fait, on va commencer par apprendre à manipuler cet environnement de programmation, phase essentiel pour pouvoir travailler confortablement par la suite.

Ajouter un bloc dans la surface de câblage

L'ajout d'un bloc dans la surface de câblage se fait par « Glisser/Déposer ». Alors pour les plus grands débutants, le glisser/déposer est l'action de cliquer sur un objet (avec le bouton gauche de la souris), de maintenir le clic tout en déplaçant la souris, puis de relâcher le bouton de la souris.

NOTE : il est possible que j'appelle parfois le « Glisser/Déposer » « Drag&Drop », ce qui est la même chose mais en anglais.

En image cela donne ça :



1 : on clique sur un bloc fonction et on maintient le clic gauche

2 : on déplace la souris tout en maintenant le clic gauche, normalement, un « fantôme » (rectangle gris) du bloc que vous allez déposer apparaît sur la surface de câblage sous votre curseur.

3 : Lorsque vous relâchez le bouton, le bloc sur lequel vous commencez le Glisser/Déposer est placé sur la surface de câblage.

Le Glisser/Déposer est une opération élémentaire que vous devez absolument maîtriser pour pouvoir utiliser l'atelier M3. C'est avec cette action qu'on va tout faire : placer des blocs, tracer des liens (ce n'est pas exactement un glisser/déposer mais le mouvement est le même), ou déplacer des blocs dans la surface de câblage.

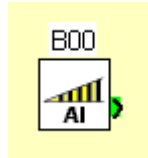
Vous remarquerez qu'au dessus du bloc de trouve un numéro **BXX**. Ceci est un numéro incrémenté automatiquement chaque fois que vous posez un nouveau bloc, j'utiliserai parfois ce numéro pour faire référence à un bloc en particulier.

La sélection d'un bloc

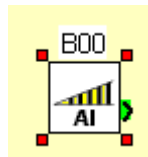
Lorsqu'ils sont posés dans la surface de câblage, il est possible de sélectionner un ou plusieurs blocs simultanément afin de les déplacer.

Commençons avec un seul et unique bloc, mais avant, observons l'apparence d'un bloc déposé dans la surface, et d'un bloc sélectionné :

Bloc déposé non sélectionné



Bloc déposé et sélectionné

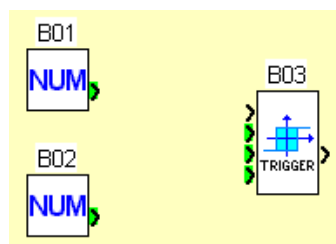


Vous aurez remarqué les petites « poignées » rouges affichées à chaque coin du bloc qui indique quel est la sélection courante.

Pour déplacer un bloc, il suffit de faire une opération de glisser/déposer à partir du bloc sélectionné (ou non sélectionné) et de relâcher la souris à l'endroit où l'on souhaite déposer le bloc.

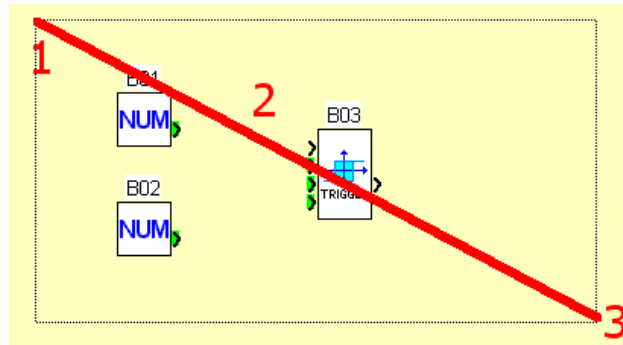
La sélection Multiple

Prenons ce groupe de blocs :



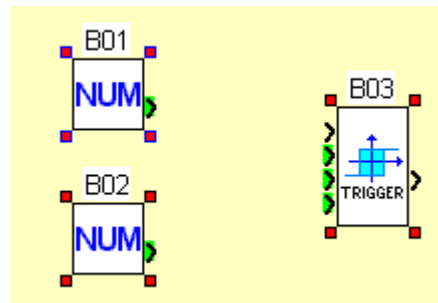
Il est composé de 2 blocs NUM (onglet IN de la barre de fonction) et d'un bloc Trigger (onglet FBD de la barre de fonction). Commencez par déposer ces trois blocs dans la surface de dessin. Leur position importe peu tant qu'ils sont assez rapprochés (pour faciliter la manipulation)

Tracez un rectangle de sélection autour du groupe de blocs en suivant la procédure suivante :



- 1 : on clique dans la feuille de câblage en dehors d'un bloc fonction.
- 2 : on déplace la souris tout en maintenant le clic gauche, normalement, un « fantôme » (rectangle gris) de votre sélection apparaît sur la surface de câblage.
- 3 : Lorsque vous relâchez le bouton, les blocs entièrement compris dans le rectangle sont sélectionnés.

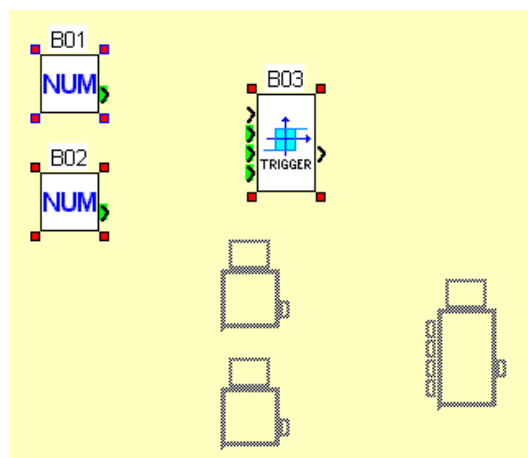
Voici le résultat :



Vous remarquerez que pour un des 3 blocs, la couleur des poignées de sélection est légèrement différente. Pour l'instant cela n'a pas d'impact, et nous verront beaucoup plus tard dans un autre document à quoi cela sert.

Le déplacement Multiple

Le grand intérêt de la sélection multiple est de permettre des déplacements de groupes de blocs. Pour ce faire, faites comme si vous déplaçiez un des blocs de votre sélection. Tous les blocs de la sélection suivent celui que vous déplacez, chacun dessinant son propre fantôme. Une fois que vous relâchez le bouton de la souris. Tous les blocs sont déplacés.



Vous aurez peut être remarqué, que parfois le fantôme du bloc apparaît en bleu plutôt qu'en gris. Cela signifie qu'il n'est pas possible de placer le bloc à l'endroit où se trouve le curseur.

Tracer des liens

Bon, poser des blocs et les déplacer, c'est bien joli, mais ça ne fait pas grand-chose. Nous allons donc voir comment se connectent les blocs fonction entre eux.

Un peu de vocabulaire :

Je vais très souvent utiliser deux abréviations qu'il est important de connaître :

TOR : Tout Ou Rien.

Le tout ou rien, c'est un interrupteur, c'est ON **ou** OFF, Marche **ou** Arrêt, 1 **ou** 0.

ANA : Analogique

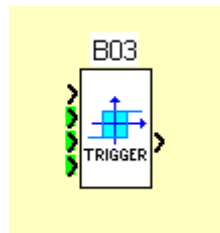
L'analogique, c'est une valeur décimale, de $-n$ à $+n$. Selon la programmation, ça peut être une température, une pression, un angle, ou même une valeur non typée. Dans le Millennium 3, les valeurs possibles en analogique varient en -32768 et $+32767$.

Les types d'entrées et de sorties des blocs

Généralité sur les entrées/sorties des blocs fonction.

Les entrées d'un bloc fonction sont TOUJOURS à gauche des blocs, et les sorties TOUJOURS à droite. ...enfin presque. Effectivement, il existe une catégorie de bloc sur laquelle on trouve des entrées en haut ET des entrées à gauche, et des sorties en bas ET à droite: les blocs **SFC**. Ceci sera expliqué bien plus tard car leur utilité est rare, et cela demande d'appréhender certains principes d'automatisme trop délicat pour le moment.

Reprenons notre bloc Trigger déjà présent sur la surface de câblage.



Si on le regarde attentivement, on observe des petites flèches noires de chaque côté du bloc, et ayant un fond de couleurs différent.

Intéressons nous dans un premier temps à la première flèche en haut à gauche du bloc :



C'est donc une petite flèche noire sans fond de couleur. Ceci est une entrée TOR. Pour l'instant on ne va pas s'intéresser à son utilité, car cela dépend de chaque bloc fonction, et cela sera décrit dans le document phase 2.

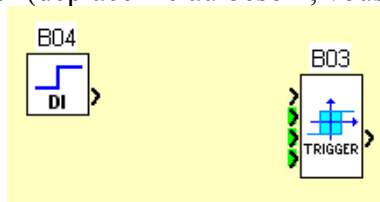
Si vous prenez la deuxième entrée du bloc (celle du milieu sur la capture suivante)




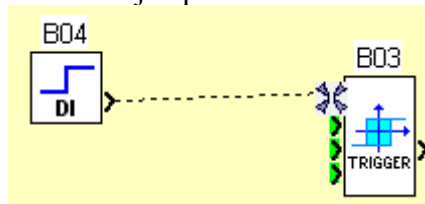
Vous observerez un fond vert. Ce fond vert indique que c'est une entrée ANA.

Le premier lien

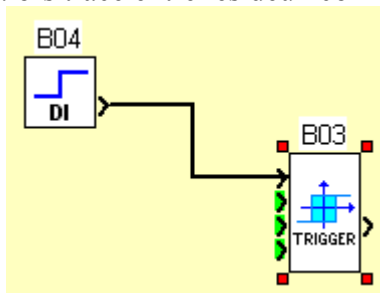
Poser dans votre surface de câblage une entrée DI (onglet IN de la barre de fonction) et placez la à gauche de votre bloc Trigger (déplacez le au besoin, vous savez comment on fait nan ;-))



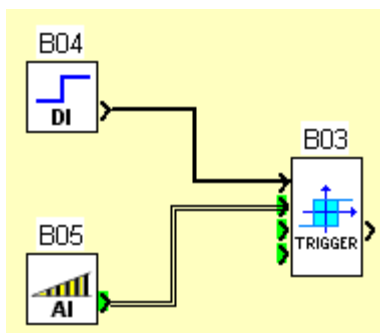
Vous remarquerez que le bloc DI, possède une sortie TOR (sa petite flèche à droite). Dans l'atelier M3, on ne peut que connecter des entrées/sorties du même type ensemble, et c'est ce que nous allons faire. Pour connectez la sortie bloc DI, commencez par déplacer votre curseur sur la sortie du bloc DI. Lorsque votre curseur survole la sortie, celui-ci change d'apparence et prend cette forme  (j'appellerais ceci le curseur de connexion). Cliquez maintenant sur le bouton gauche et déplacez votre curseur jusque sur l'entrée TOR du bloc Trigger :



Une fois que le curseur à repris son aspect « curseur de connexion », vous pouvez relâcher le bouton de la souris, un lien est alors tracé entre les deux connexions.



Tant que nous sommes sur cette lancée, nous allons poser un bloc entrée ANA et tracer un lien entre sa sortie et la première entrée ANA du bloc Trigger (la première entrée avec un fond vert comme ceci :

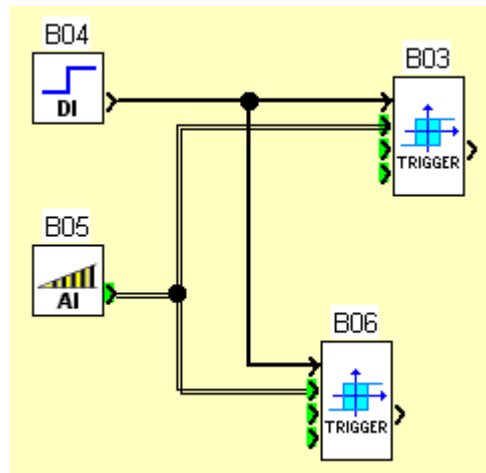


Vous aurez remarqué l'aspect différent des deux liens. L'un est un lien TOR, et l'autre un lien ANA. Leur aspect est différent afin de bien les distinguer dans le programme.

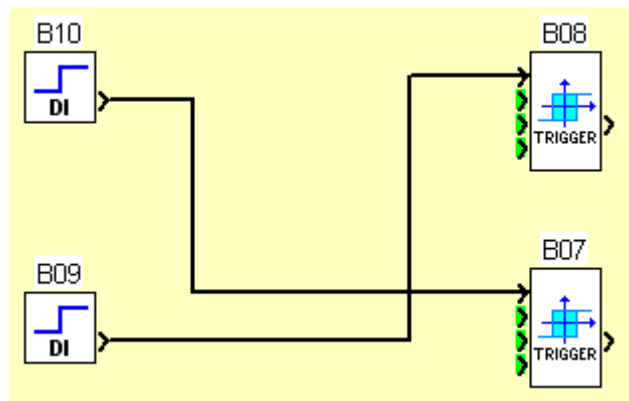
L'aspect des liens filaires et leurs positionnements

De manière générale :

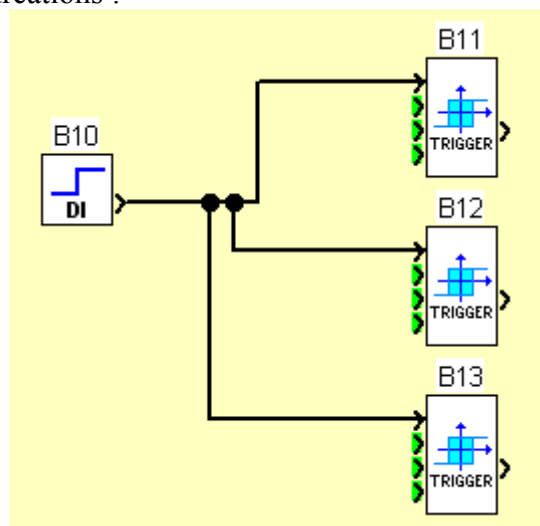
On **peut** connecter une sortie à plusieurs autres entrées de bloc comme ceci :



Vous observez ici, qu'à l'endroit de la bifurcation du lien, l'atelier dessine un rond, indiquant qu'il s'agit bien d'une bifurcation, et non pas de liens qui se croisent comme dans la capture suivante :

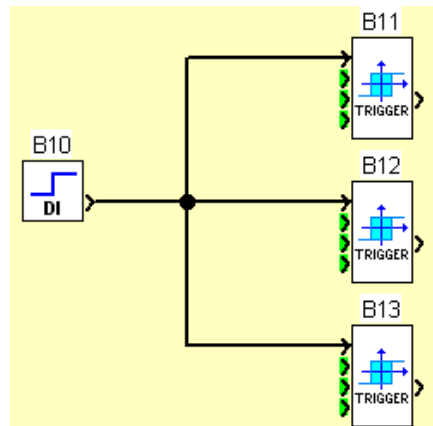


Reprenons le cas des bifurcations :



Ici j'ai connecté la sortie du bloc DI à 3 blocs triggers sans rien retoucher du placement automatique des liens réalisés par l'atelier M3.

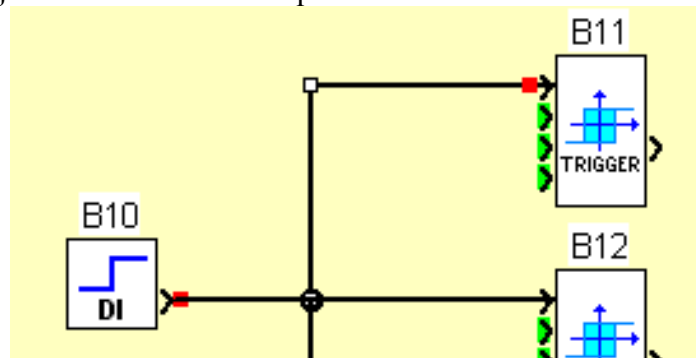
Regardons maintenant la capture suivante :



On remarquera que les liens font moins de coude, et qu'on a simplement un seul « nœud » de bifurcation. Pour mettre en forme mes blocs et mes liens de manière plus lisibles, j'ai commencé par placer le bloc DI (B10) dans l'axe du bloc Trigger (B12) de manière à ce que le lien soit droit, puis j'ai déplacé les deux autres liens pour les aligner.

Pour déplacer un lien, il faut commencer par cliquer dessus.

Dans cette capture j'ai sélectionné le lien qui relie le bloc DI B10 au bloc Trigger B11.

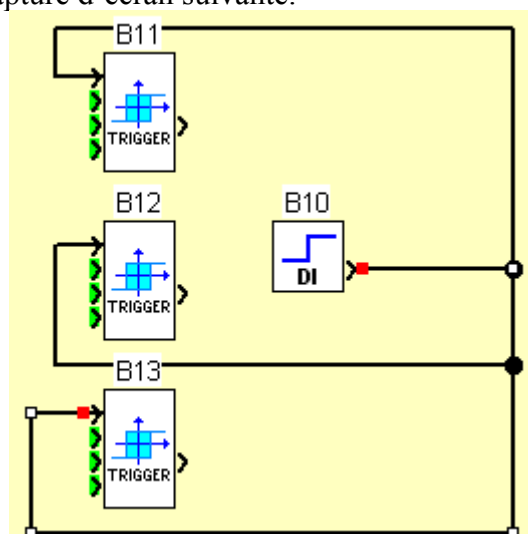


Vous remarquerez qu'à chaque extrémité du lien est apparu un carré rouge, ainsi que des carrés blancs à chaque coude de celui-ci. Ceci indique quel lien est sélectionné.

Lorsque vous survolez la partie verticale de ce lien, le curseur change et prend cet aspect :



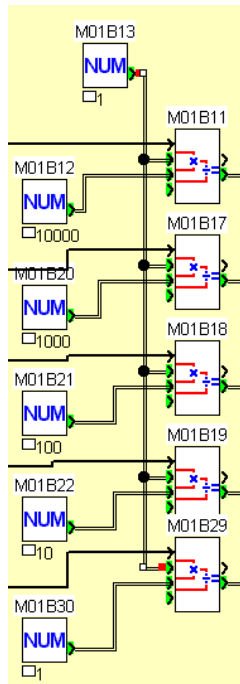
De la même manière que pour déplacer un bloc, cliquer et maintenir le bouton gauche de la souris permet de déplacer le lien. Ceci permet donc d'arranger les liens afin de rendre votre programme plus lisible. Ce déplacement peut être aussi appliqué sur des branches horizontales de liens comme dans la capture d'écran suivante.



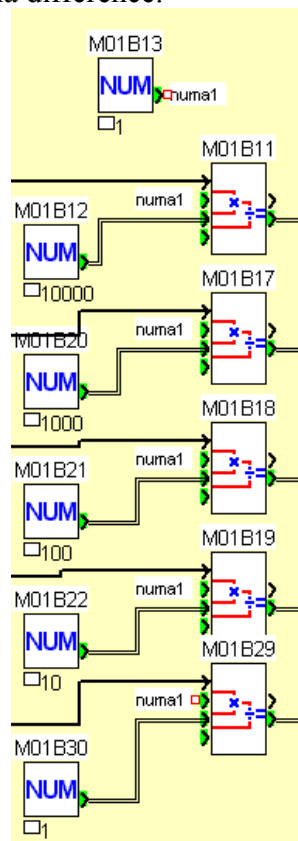
Les liens textes

Jusqu'ici les exemples présentés sont assez simples et permettent d'avoir un programme qui reste lisible même avec liens filaires.

Prenons donc un exemple complexe :

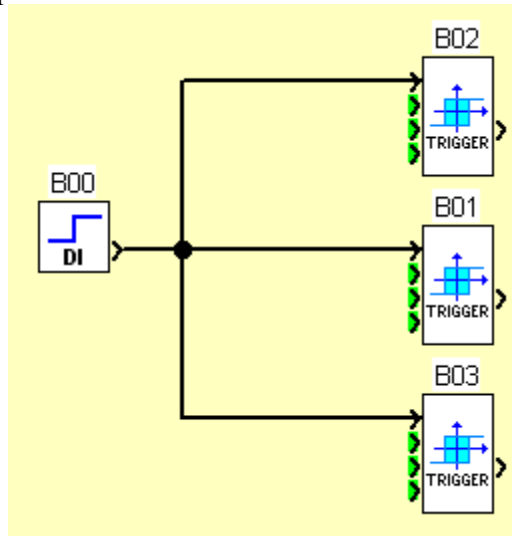


Ici, vous l'aurez vu, c'est nettement moins clair... encore que ça reste plutôt lisible. Mais mettons un lien texte et regardons la différence.

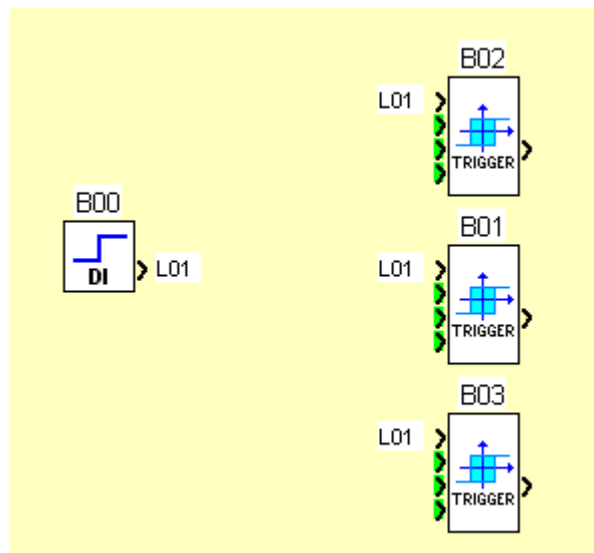


Avec un lien texte, on gagne ici en clarté car on limite le croisement des liens entre eux. Étudions comment faire un lien texte :

Reprenons notre exemple précédent :

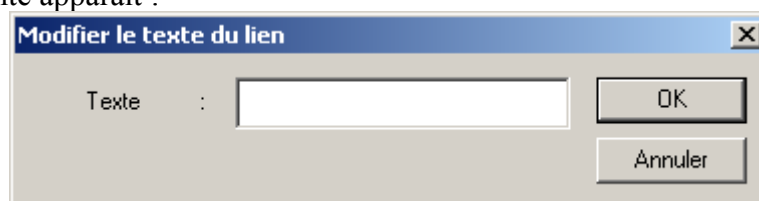


Sélectionner un lien, puis allez cliquer sur la commande de menu « Outils → Type de câblage → Texte »
Voici le résultat.



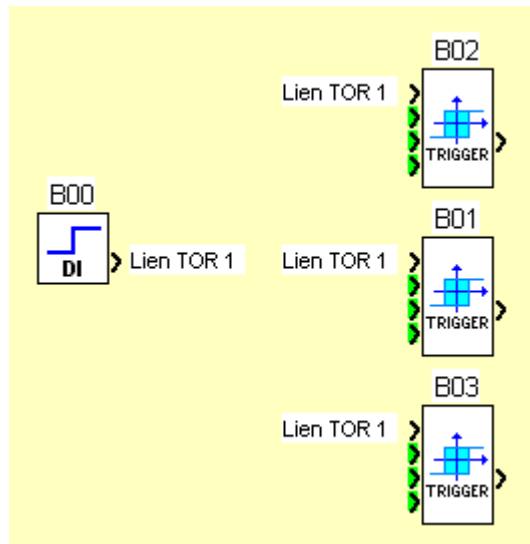
Pour l'instant le texte est un numéro automatique, mais il est possible de le changer afin de le rendre plus explicite, ceci rendant au passage le programme encore plus lisible. Pour changer le texte, sélectionner le lien texte (en cliquant sur le nom qui est affiché à côté de la sortie du bloc DI ou à côté d'une entrée TOR des blocs Trigger), puis cliquer dans le menu « Outils → Type de câblage → Modifier le Texte »

La fenêtre suivante apparaît :



Saisissez le texte que vous voulez (essayer d'abrégé autant que possible, en utilisant des acronymes explicites) puis appuyez sur OK.

Voilà ce que cela donne pour l'exemple :



Alors bien sur dans ce cas, ce n'est pas ultra intéressant, mais lorsque vous aurez un programme très compliqué, avec des blocs dispersés sur la surface de câblage et que des liens traverseront d'un bout à l'autre votre programme, vous comprendrez mieux l'intérêt d'utiliser des liens textes plutôt que des liens filaires.

Une autre petite chose utile : Créer des liens texte par le menu standard peut être parfois un peu lourd à l'utilisation, mais il est également possible de le faire depuis le menu contextuel. Le menu contextuel c'est le menu que l'on affiche en faisant un clic sur le bouton droit de la souris. Vous pouvez faire apparaître le menu contextuel n'importe où sur la surface de câblage, mais son contenu sera différent selon que vous cliquez dans le vide, sur un bloc, ou sur un lien. Si vous cliquez sur un lien avec le bouton droit de la souris, vous aurez les mêmes options que celles du menu « Outils → Type de câblage »

Premier programme et Paramétrage d'un bloc fonction

Alors pour l'instant nous n'avons pas beaucoup étudié les blocs fonctions existants dans l'atelier Millenium 3, mais maintenant que vous avez exploré les manipulations de base, intéressons nous à ces fameux blocs. Pour l'instant nous allons nous contenter d'en survoler quelques uns, afin de faire un premier programme très simple qui aura deux fonctions :

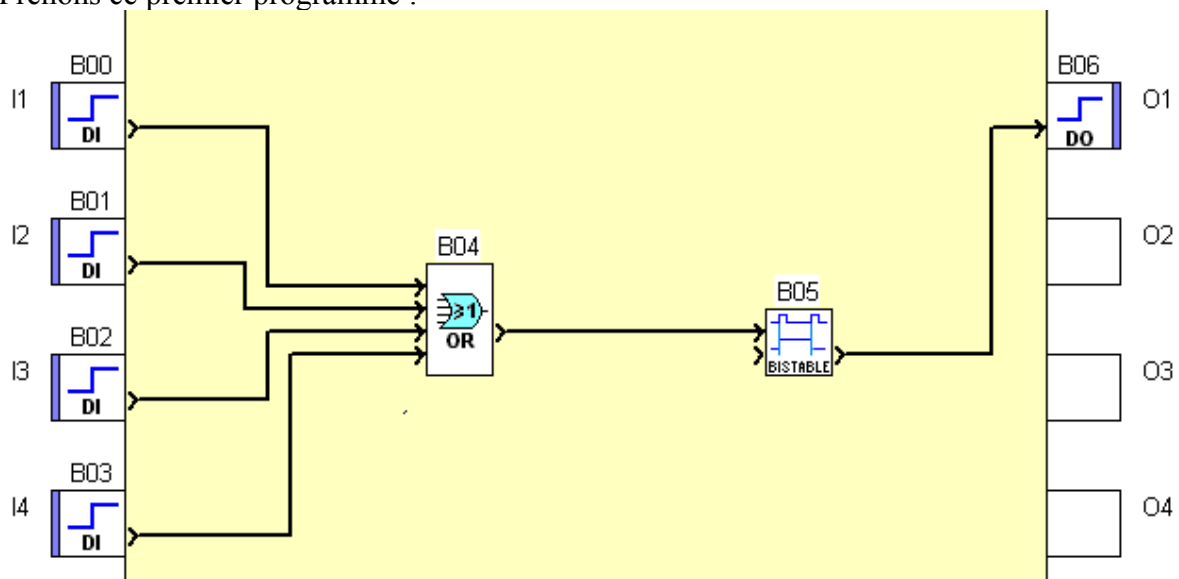
- Télérupteur pour une lampe à plusieurs boutons... soyons fou, on va en mettre 4, avec une minuterie automatique d'arrêt.
- Equivalent d'un thermostat d'ambiance mécanique

Préambule : Ici on va voir l'effet des liens qui peuvent être considérés comme des câbles électriques. Si on met une tension à un bout, on la retrouve également à l'autre bout.

J'ai choisi volontairement deux fonctions très simples à schématiser et parlantes pour tout le monde afin d'aborder un point important qui est le paramétrage des blocs fonctions. Dans le programme que nous allons réaliser, le paramétrage va être simpliste ceci afin de ne pas vous noyer dans une combinaison de paramétrage qui rendrait la compréhension finale du programme difficile.

Le télérupteur

Prenons ce premier programme :



Alors commençons par décrire les éléments présents :

- B00 à B03 : les entrées TOR
- B04 : La fonction logique OU
- B05 : la fonction BISTABLE (Télérupteur)
- B06 : la sortie TOR

Les blocs B00 à B03 sont posés sur les plots d'entrées de l'automate et permettent donc d'utiliser les entrées physiques I1 à I4 de l'automate. Ainsi, lorsqu'une des entrée physique

sera alimenté, la sortie TOR du bloc correspondant passera à VRAI (ou 1, ou ON). Sur ces entrées on y connecterai des boutons poussoirs (comme pour un télérupteur standard).

Le bloc B04 est une fonction logique OU. Alors pour l'instant je ne vais pas pousser à fond ce qu'on appelle la logique booléenne car c'est un point qui sera abordé en profondeur dans le prochain document, mais sachez que pour notre exemple, la sortie de ce bloc sera à VRAI dès qu'au moins une des entrées sera à VRAI. Donc d'après le schéma, tant que l'un des 4 boutons sera appuyé, sa sortie restera à l'état VRAI. On l'observera plus tard lors de la simulation.

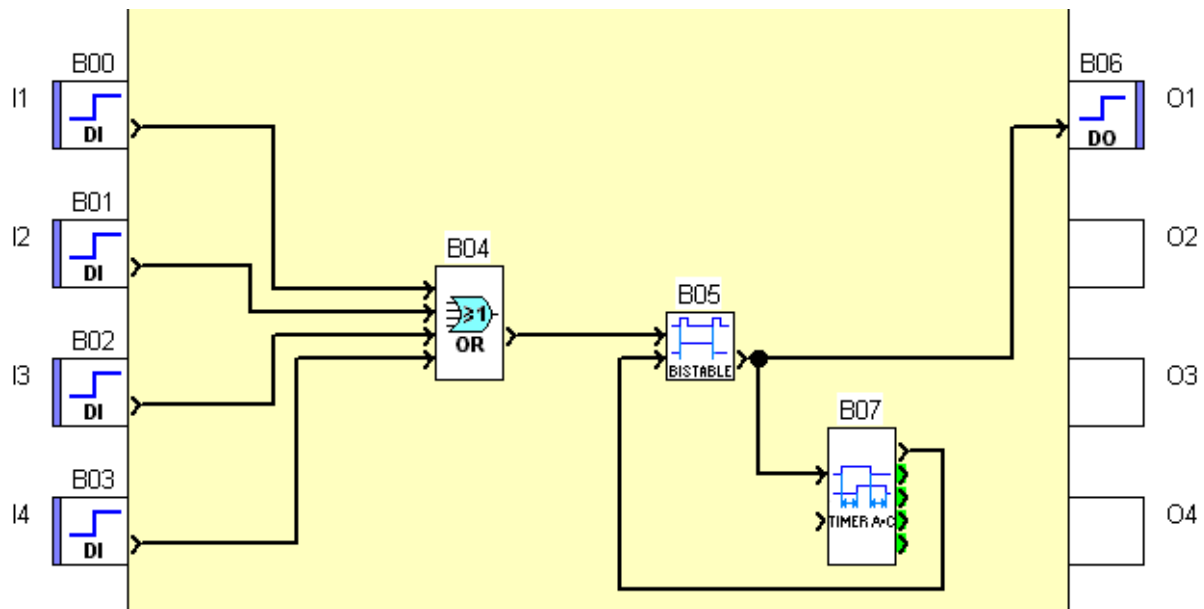
Le bloc B05 est un télérupteur : c'est-à-dire que si vous lui envoyez une impulsion ponctuelle, sa sortie change d'état. Si sa sortie était à 1, elle passe à 0, si elle était à 0, elle passe à 1. Il fonctionne comme un vrai télérupteur que l'on câble dans une armoire électrique. Autre propriété de ce bloc, c'est que sa seconde entrée TOR permet de repositionner sa sortie à FAUX quelque soit son état courant.

Le bloc B06 est une sortie TOR, sur un automate XD26 (modèle vendu par le groupement d'achat) c'est le seul type de bloc autorisé sur les plots de sortie de l'automate. Si son entrée est à VRAI, alors le relais correspondant sera fermé (pour des contacts on entend par fermé le fait que le courant puisse passer)

Alors ce premier morceau de programme ne nécessite aucun paramétrage particulier, c'est pourquoi nous allons ajouter d'autres choses.

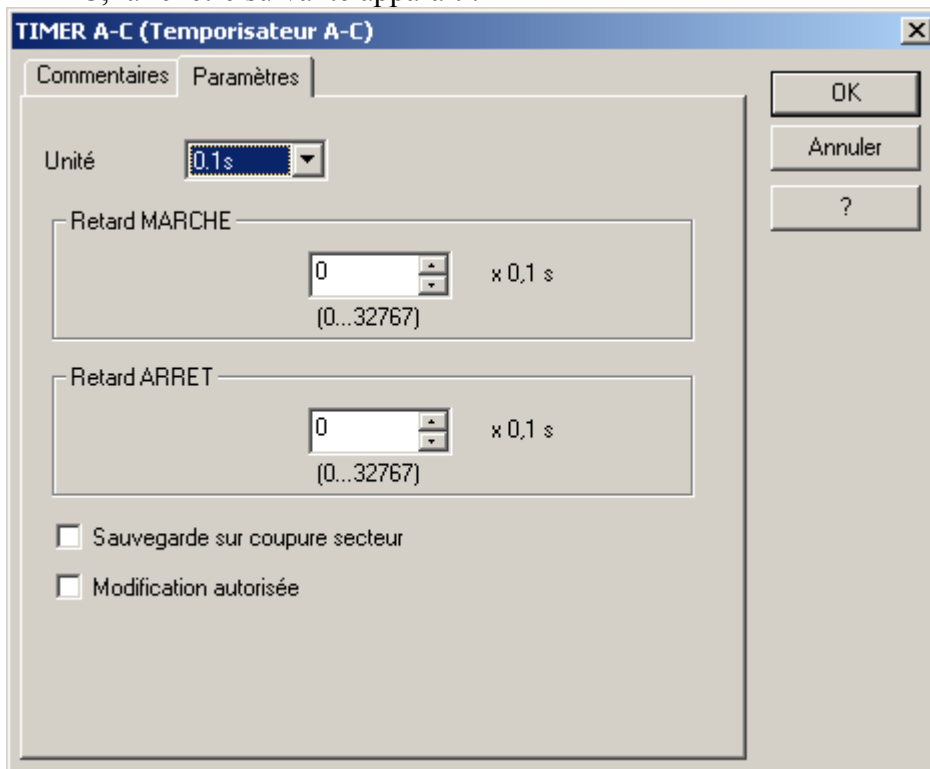
La minuterie d'arrêt :

Dans ce premier morceau de programme, un appui sur le bouton mettra la lumière en marche, et un second appui l'arrêtera. Imaginons que nous sommes dans la partie commune d'un immeuble, ou tout simplement dans votre sous sol à un endroit où la lumière n'a pas lieu de rester allumer éternellement. Nous allons donc ajouter une minuterie qui éteindra automatiquement la lumière si ceci n'a pas été fait manuellement au bout de 2 minutes.



Un nouveau bloc fonction (B07) est apparu, c'est le bloc TIMER AC ou TEMPORISATEUR. Ce bloc a pour but de retarder sa sortie par rapport au signal d'entrée suivant des paramètres, dans notre cas nous ne nous intéresserons qu'à sa première entrée et à sa première sortie. Sa première entrée TOR est l'entrée « Commande », le timer ne

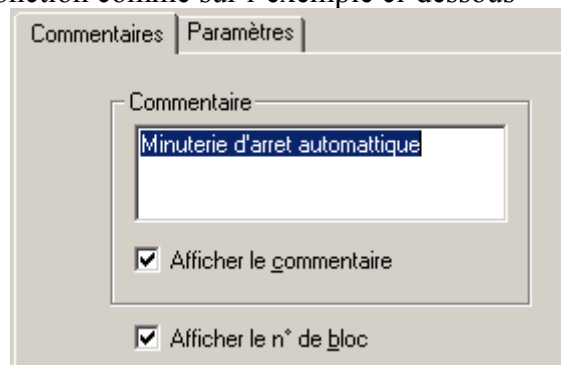
fonctionnera donc que lorsque cette entrée sera active, et sa sortie TOR (la première sortie) ne sera active qu'une fois le temps configuré en paramètre atteint. Depuis tout à l'heure je vous parle de paramètre mais on se demande bien où cela se cache. Double cliquez sur le bloc TIMER AC, la fenêtre suivante apparaît :

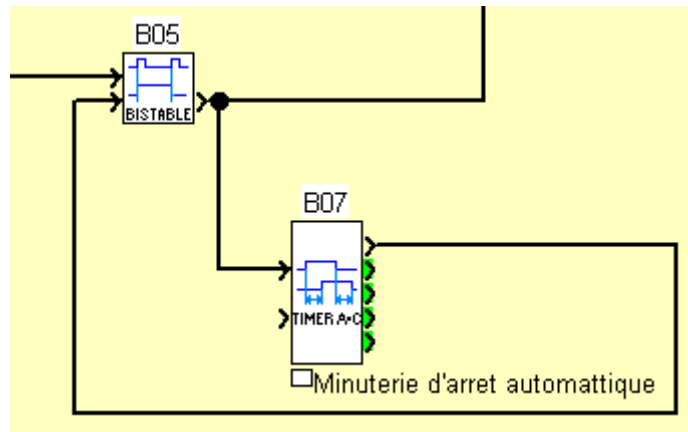


Voilà donc à quoi ressemble une fenêtre de paramétrage, et la description de certains éléments s'impose :

Les onglets :

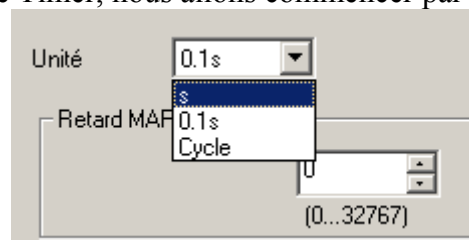
Une fenêtre de paramétrage peut avoir 1 seul onglet ou plusieurs (jusqu'à 3). L'onglet commentaire est toujours présent et fournit au minimum la possibilité d'écrire un commentaire (dans le cas où la fonction n'a pas de paramètre) que l'on pourra alors faire apparaître sous le bloc fonction comme sur l'exemple ci-dessous



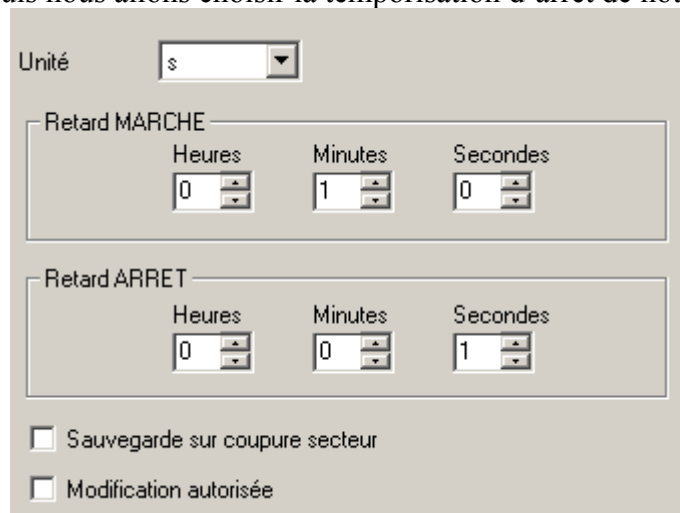


Ensuite il y a les 3 boutons à droite de cette fenêtre qui sont « OK », « Annuler », et « ? ». Bon pour OK et Annuler rien de très compliqué puisqu'ils servent à valider ou annuler les modifications que vous avez effectuées sur les paramètres de la fonction. Pour ce qui est du bouton « ? », il va vous être très utile dans vos débuts car celui-ci permet d'ouvrir l'aide de la fonction. Cette page d'aide explique les entrées et sorties de bloc fonction, ainsi que ses paramètres et son fonctionnement. Il est vrai que pour certains blocs fonction l'aide en ligne est plus que succincte, et c'est pourquoi le prochain document fera une étude poussée avec des exemples des blocs fonction présents dans l'atelier.

Pour le paramétrage de notre Timer, nous allons commencer par changer la base temps :



Pour l'unité de temps, nous allons choisir la seconde (s) comme présenté sur la capture d'écran ci-dessus, puis nous allons choisir la temporisation d'arrêt de notre système.



Dans cet exemple j'ai choisi une minute de retard marche, et 1 seconde de retard arrêt. Le 1 minute de retard marche correspond au temps pendant lequel notre lampe restera allumée (si personne ne l'éteint avant), le 1 seconde correspond au temps où le signal de sortie du timer sera actif afin de désactiver le télérupteur.

Des paramètres qu'on voit souvent :

Sauvegarde sur coupure secteur : lorsque cette case est cochée, elle permet au bloc fonction de sauvegarder sa valeur courante lorsqu'une coupure secteur est détectée, et de la restituer lorsque le courant revient.

En effet, le M3 est capable de détecter une perte d'alimentation, et dans ce cas il utilise sa réserve d'énergie présente dans les condensateurs afin d'effectuer des opérations de sauvegarde de valeurs.

Modification autorisée : Le millenium 3 offre la possibilité de modifier les paramètres des blocs fonction via le menu de la face avant du M3 même si celui-ci est en fonctionnement. Je ne m'étendrai pas sur cette fonction, car dans de nombreux cas il est préférable d'utiliser les fonctions de programmation de l'affichage pour réaliser ce type d'opération (il existe un document dédié à la programmation de la face avant sur le site APPER)

Valider vos paramètres avec le bouton « OK ».

Avant de passer à la suite, quelques remarques sur ce qu'on vient d'ajouter. Le timer AC (ou temporisateur) que nous avons ajouté fait office de minuterie mais comment ça marche.

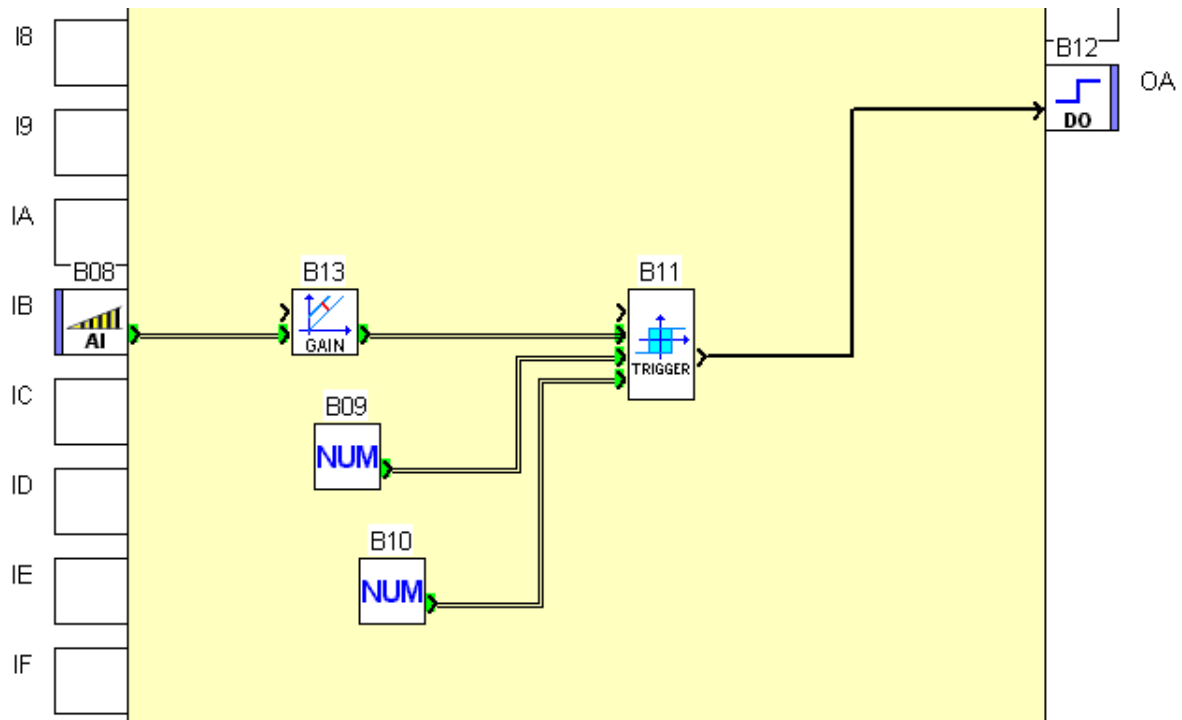
Quand on active le télérupteur, celui-ci active en même temps la lumière (sortie de l'automate), et l'entrée d'activation du timer. Ce dernier étant activé, il commence à compter jusqu'à atteindre le temps « Retard Marche » qui a été paramétré. Une fois ce temps atteint, il active sa sortie qui est connecté à l'entrée RESET du télérupteur et provoque donc l'arrêt de ce dernier, ce qui en conséquence arrête par la même occasion la lumière et le timer lui-même.

J'admets qu'expliquer sous forme de texte, ce n'est pas forcément très limpide, mais vous pourrez observer cela plus en détail un peu plus tard lorsqu'on passera le programme en simulation.

Mais avant cela, ajoutons autre chose dans notre programme.

Le Thermostat d'ambiance

Un peu de nouveauté, ça ne fait pas de mal, et nous allons donc utiliser des entrées analogiques. Je sais votre but (pour beaucoup d'entre vous) est de réguler votre installation solaire et donc c'est un sujet qui va être particulièrement instructif (enfin pour un début). Voici le bout de programme que je vous propose de câbler.



D'abord une petite liste des fonctions présentes :

- B08 : l'entrée analogique. Celle si est posée sur le plot d'entrée IB car c'est la première des 6 entrée analogiques du M3 (les plots I1 à IA ne supportent que des blocs DI)
- B13 : La fonction « Gain », dans notre exemple, elle va nous servir à convertir le signal de l'entrée analogique en degrés
- B11 : la fonction TRIGGER. Cette fonction n'est pas forcément simple à comprendre, je vais détailler son fonctionnement juste après.
- B09 – B10 : la fonction constante numérique
- B12 : la sortie TOR.

Le bloc TRIGGER :

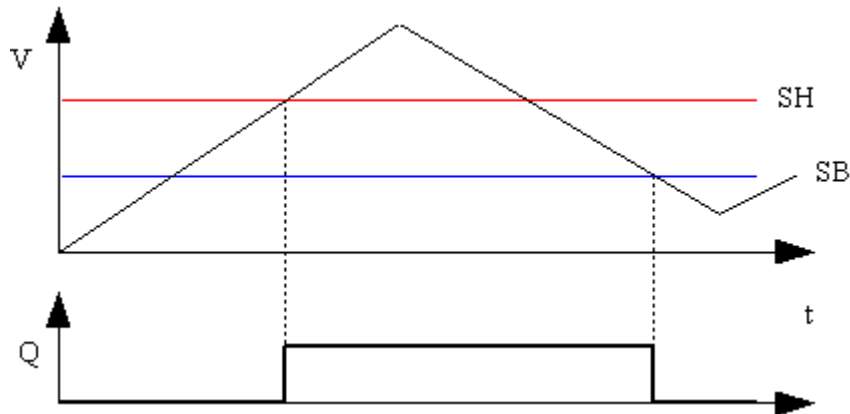
Le trigger va être un élément indispensable de toute régulation, et c'est pourquoi j'ai décidé de l'approfondir un peu dans ce tutoriel.

Imaginons un CESI simple, et disons que l'on souhaite démarrer le circulateur dès que la température dans les panneaux dépasse 50°C (dans la réalité on le fera pas rapport à la température du ballon). Si on utilise un comparateur simple (marche du circulateur si $T^\circ \text{ capteur} > T^\circ \text{ consigne}$ et arrêt dès que $T^\circ \text{ capteur} < T^\circ \text{ consigne}$), dès que la température dans le capteur aura atteint notre consigne, le circulateur va démarrer, et provoquer le refroidissement du capteur, ce qui aura pour effet de faire repasser la température du capteur sous la consigne et donc arrêter le circulateur. L'eau ne circule plus, la température remonte donc dans le capteur, ce qui va une nouvelle fois re-déclencher la circulation, qui va refroidir le capteur... et ainsi de suite.

Vous l'aurez compris, ce n'est pas la bonne façon de procéder car le circulateur risque de ne pas beaucoup aimer d'être arrêté et redémarré 50 fois par jour, et va donc vieillir prématurément.

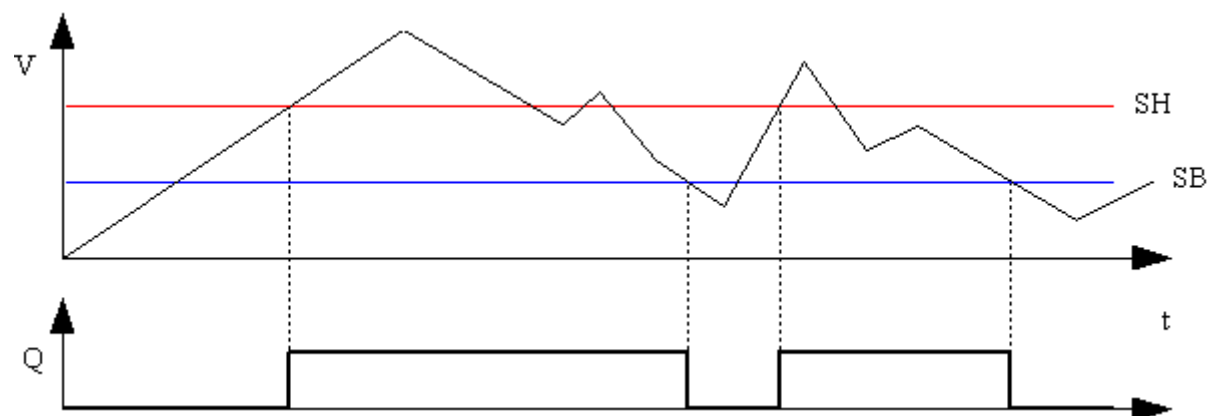
Mais ~~zéro~~ le trigger est arrivé, et va nous permettre de résoudre ce problème. En fait le trigger est un double comparateur. En gros on va pouvoir lui donner des consignes de marche et d'arrêt différentes.

Prenons le chronogramme suivant : en rouge nous avons le seuil haut noté SH (seuil de arrêt à marche), et en bleu le seuil bas noté SB (seuil de marche à arrêt). La courbe noire fine sur le graphique du haut représente la valeur d'entrée (valeur à comparer qui pourrait être notre température capteur), et sur le graphique du bas, en trait noir épais nous avons la sortie de notre TRIGGER (Q).



On remarque donc que la sortie du trigger (activation du circulateur) ne s'active que lorsque la température du capteur a dépassé le seuil haut, et ne s'arrête qu'une fois que cette même température est repassée sous le seuil bas.

Sur un exemple un peu plus complet on obtient ce résultat :



Ainsi, malgré les nombreuses variations de la température du capteur, le nombre de démarrage/arrêt du circulateur est limité.

Revenons maintenant à notre histoire de thermostat mécanique. Si vous avez déjà fait un peu attention à son fonctionnement, vous aurez remarqué qu'il y a un certain delta entre son passage de arrêt à marche, et de marche à arrêt... comme le trigger que nous venons de voir, à la différence près que dans notre M3, nous allons pouvoir régler ce delta et avoir des seuils variables en fonction d'une autre température. (Voir le document de T.Streiff : [Régulation CESI](#))

Le bloc Gain :

Ce bloc fonction est très simple, puisqu'il applique l'opération :

$$\rightarrow \text{Sortie} = ((A / B) \times \text{Entrée}) + C.$$

L'intérêt principal de ce bloc c'est qu'il regroupe en une seule fonction tout ce qu'il nous faut pour convertir le signal d'une sonde en degré. Il est également possible de définir le maximum et minimum que peut atteindre la sortie. Sa page de paramétrage est la suivante :

Une règle simple pour paramétrer ce bloc :

A = plage de température de la sonde

B = plage de valeur du convertisseur numérique du M3

C= offset de la plage de température de la sonde.

Prenons un exemple concret avec des sondes présentes dans le bon de commande du groupement d'achat d'automates programmables.

- Sonde extérieure "-10 à +40 °C
- Sonde ambiance "-10 à +40 °C
- Sonde immergée "-10 à +150 °C

Commençons par la sonde extérieure (et la sonde intérieur):

- La plage de température est de 50°
- La plage de valeur du convertisseur numérique du M3 est toujours la même (0-1023)
- Offset de la plage de température de la sonde = -10

Donc nous allons régler nos coefficients comme cela :

A = 50, B = 1023, C = -10.

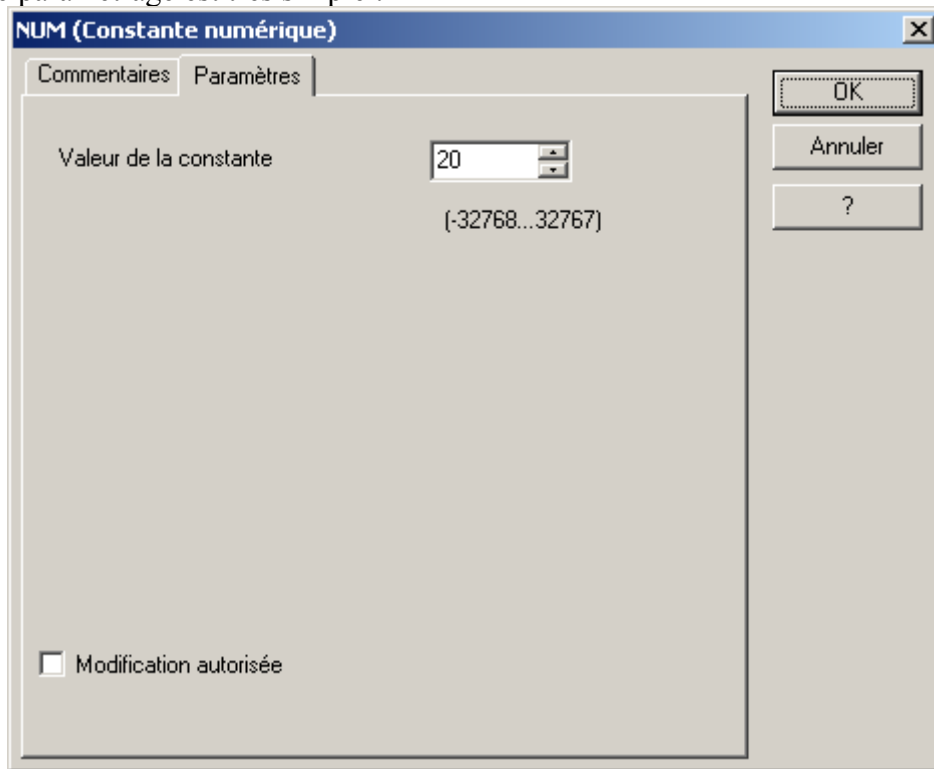
En appliquant le même procédé, nous aurons pour la sonde immergée :

A = 160, B = 1023, C = -10.

Le bloc constante numérique :

Alors ce bloc la est très simple, ce n'est qu'une constante numérique. On défini la valeur qu'il aura via son paramétrage, et il ne changera jamais (enfin pour ne pas vous mentir, on pourra le

changer grâce aux fonctions de programmation de la face avant, mais je ne vais pas vous embrouiller avec ça pour le moment)
Sa page de paramétrage est très simple :



Il suffit donc de choisir la valeur que l'on souhaite donner à la constante.
Configurez les deux blocs constante comme décrit ci-dessous :

- B09 = 20
- B10 = 21

Je vais être directe sur ce point, ce paramétrage va poser un petit problème mais le mettre en évidence va permettre de vous faire comprendre que lorsqu'on programme une régulation avec des entrées analogiques, il faut prendre en considération les légères fluctuations électriques qui peuvent aboutir à des phénomènes d'oscillation d'état dans votre programme.

Bon nous sommes maintenant fin prêt pour faire un premier test de notre programme.

La simulation

Pour l'instant nous allons nous intéresser qu'aux fonctions de base de la simulation, à savoir l'action sur les éléments d'entrées sorties, les forçages, et la simulation des entrées analogiques.

Une information important sur le simulateur de l'atelier Millenium 3. Le simulateur de l'atelier Millenium 3 est ce qu'on appel un simulateur fidèle. Ceci veut dire que tout est exécuté exactement de la même manière que si votre programme était chargé dans votre M3. Le coté intéressant de ceci est que vous allez pouvoir presque complètement mettre au point votre programme tranquillement assis devant votre PC. Pourquoi presque? Parce que comme je l'ai dit juste avant, il peut arriver que l'on n'ait pas pris en considération certains facteurs physiques du monde réel, qui auront un impacte sur votre programme.

Passer en simulation : les changements dans l'atelier

Dans la barre d'outils, vous avez des boutons bien utiles pour changer de mode :
Ici on va s'intéresser aux boutons E, S, M.



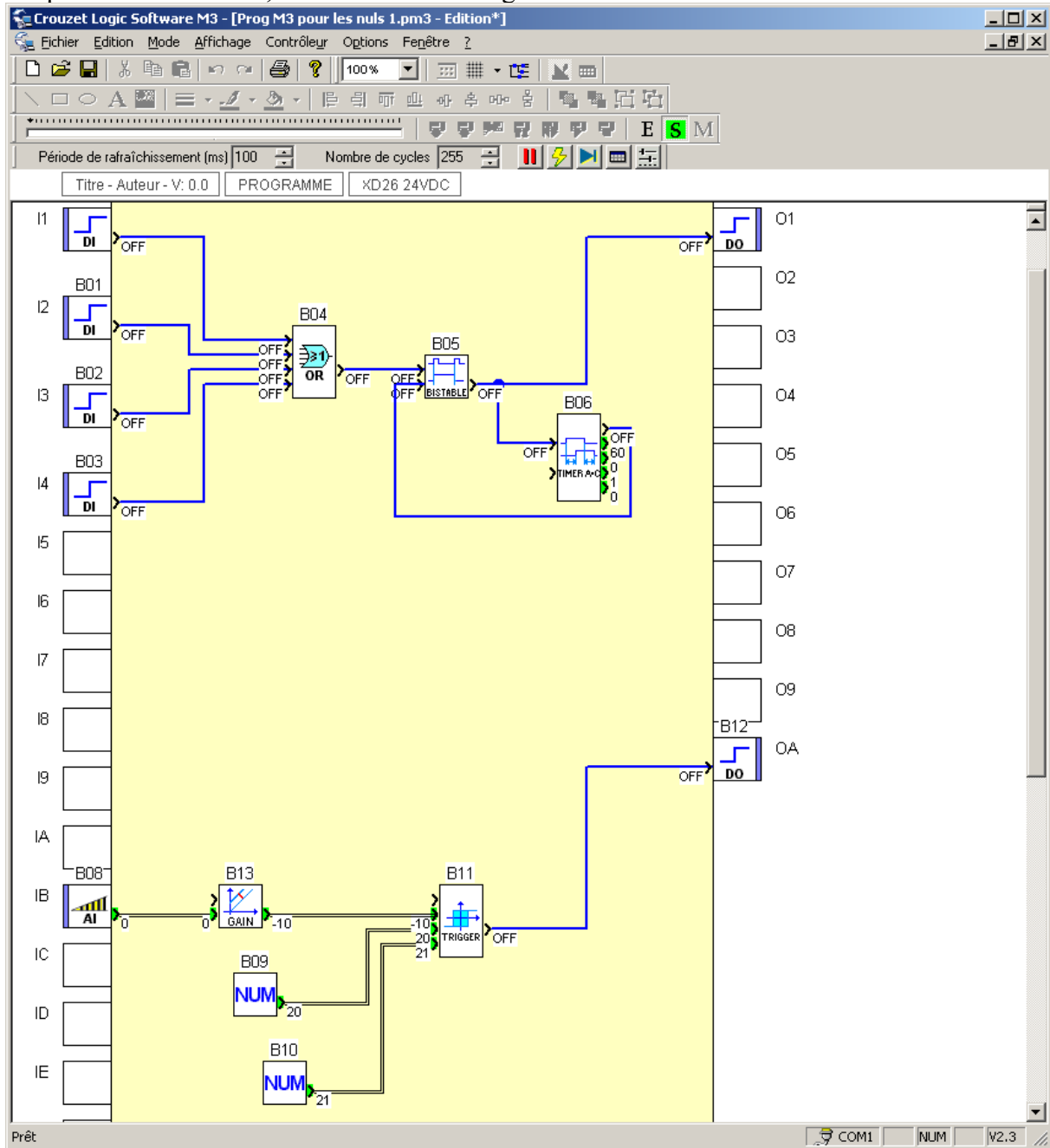
E : bouton pour passer en mode édition

S : bouton pour passer en mode simulation

M : bouton pour passer en mode monitoring.

Le bouton surligné de vert correspond au mode courant.

Cliquez sur le bouton S, et observez les changements dans l'atelier :



Voici une petite liste des changements :

- la barre de fonction a disparue

- Les liens TOR ont pris de la couleur.
- Des valeurs sont affichées à coté des entrées et sorties de bloc
- Vous ne POUVEZ PLUS déplacer les blocs fonction ou tracer des liens
- Vous pouvez toujours modifier des paramètres (sauf pour quelques blocs fonctions spécifiques, mais qui n'ont pas encore été présentés pour le moment)
- Une nouvelle barre d'outils est apparue : la barre d'outils de simulation (image ci-dessous)



Une description un peu plus poussée de cette barre d'outils sera faite au fur et à mesure,

- Vous pouvez actionner certains éléments de votre programme.

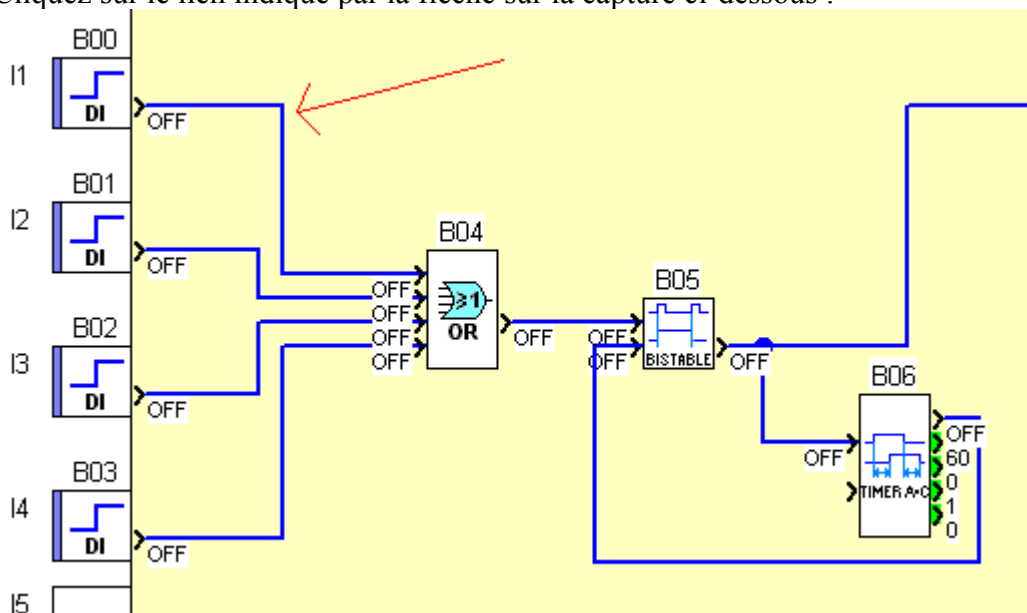
Pour l'instant intéressons nous à ce qu'on peut faire sur notre programme en mode simulation. Une petite note au passage par rapport au fait les liens TOR ont pris de la couleur (pour rappel ce sont les traits noir plein en mode édition). Lorsque ces liens sont bleus, cela signifie que l'état est à OFF, cet état est également matérialisé par un texte OFF affiché à chaque extrémité de ce lien. Lorsque ces liens sont rouge, cela signifie que l'état est à ON, cet état est également matérialisé par un texte ON affiché à chaque extrémité comme pour l'état OFF.

Les forçages fugitifs

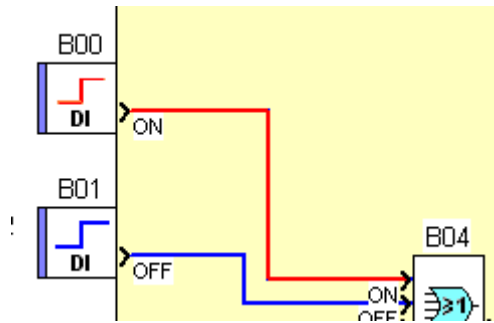
Pour étudier ce type d'action nous allons pour le moment nous concentrer uniquement sur la partie du programme qui concerne notre télérupteur temporisé.

Alors il faut bien distinguer les différents types de forçage que l'on peut effectuer sur le programme, et pour cela nous allons commencer par les forçages fugitifs. Ce sont des forçages qui ne sont pas maintenues dans le temps

Cliquez sur le lien indiqué par la flèche sur la capture ci-dessous :

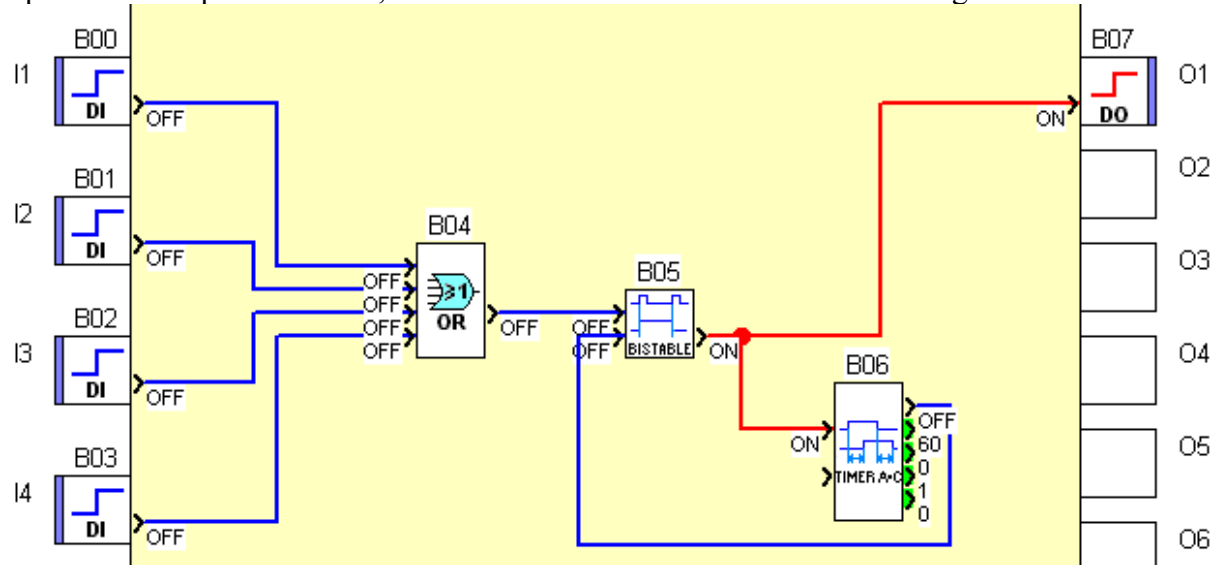


Si vous avez bien visé avec votre souris, vous avez du le voir changer de couleur rapidement avant de reprendre son état initial :



Vous aurez noté que ce changement d'état est très fugitif.

Après avoir cliqué sur ce lien, vous avez du observer d'autres choses changer d'état :



Eh oui, vous avez « appuyé ET relâché » un bouton (action fugitive), ce qui a déclenché le télérupteur, et vous voyez donc que la sortie O1 (bloc B07) est active.

Alors pour les plus pointilleux d'entre vous, un petit truc aurait du vous chagriner. En effet, on n'a pas vu changer d'état la sortie du bloc logique **OU** B04 qui est sensé activer le télérupteur... mais alors comment celui-ci s'est activé !!! Non, je vous jure que ce n'est ni par l'opération du saint esprit, nous l'œuvre du malin (encore que vous pouvez considérer un ordinateur comme étant une machine diabolique qui ne fait jamais rien comme vous le voulez)

L'explication :

Il faut savoir que par défaut le simulateur exécute 10 cycles de l'automate en 100 millisecondes, ce qui veut dire qu'un changement effectif sur seulement un seul cycle a très peu de chance d'être visible, et si il l'est, il sera extrêmement furtif.

Un cycle...mais c'est quoi cette bête !!! Pourquoi il utilise des mots barbares le Pasquall, il cherche à nous embrouiller !!!!

Alors oui la j'ai sortis comme ça un mot vulgaire, qui sans explications risque de vous perdre et je vais donc m'empresse de vous expliquer ce que c'est.

Il faut considérer que le M3 va exécuter votre programme en partant des entrées, en suivant les liens, pour finalement arriver aux sorties. Une exécution complète de gauche à droite correspond à 1 cycle. De manière générale, le M3 va mettre en 10 et 20 millisecondes pour exécuter un cycle selon la complexité de votre programme.

Quand on est en simulation on a la chance de pouvoir volontairement régler la vitesse d'exécution du programme afin de pouvoir observer ces fameux états furtifs qui sont mangés par la vitesse de calcul de nos PC. Voyons ca en détails dans ce qui suit.

Ralentir la simulation

Je vous avez dit qu'on allait reparler de la barre d'outils de simulation !!!



Alors nous allons nous intéresser aux deux premiers éléments en partant de la gauche : La **période de rafraîchissement**, et le **nombre de cycles**. Ce sont ces deux paramètres qui vont définir la vitesse de simulation.

La **période de rafraîchissement (que je noterai PF)** est réglable de 100 ms à 2000 ms. Elle va définir la période à laquelle on va simuler le **nombre de cycles (que je noterai NbC)** définit dans le champ suivant.

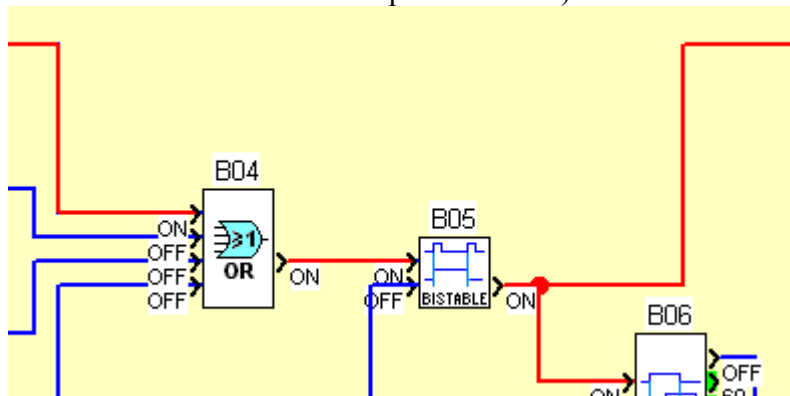
Le **nombre de cycles** va définir combien de cycles sont exécutés à chaque **période de rafraîchissement**. Le **NbC** est réglable de 1 à 255.

Alors oui la on a un beau serpent qui se mord la queue mais prenons un exemple concret pour mieux comprendre.

Par défaut ces deux valeurs sont réglées à **PF = 100**, et **NbC = 10**. Cela veut dire qu'on exécute un cycle toutes les 10ms ($100\text{ms} / 10 = 10 \text{ ms/cycle}$), soit environ ce que fait le M3 en vrai sur un programme simple. On est à peu près en temps réel (je dis bien à peu près, le simulateur est bien moins précis sur le temps que le M3)

Allez ! On va jouer un peu, et on va changer le **NbC** à 1 (utilisez les petites flèches pour régler la valeur). Nous exécutons donc 1 cycle toutes les 100ms ce qui veut dire qu'on est ralenti environ 10 fois.

Voyons ce que ça donne en cliquant sur le même lien que tout à l'heure. Normalement vous devez voir de façon fugitive cet état (note, l'état de la sortie du télérupteur peut être OFF s'il était encore actif au moment où vous avez cliqué sur le lien) :



Vous voyez, aucune source occulte n'était responsable du changement d'état du télérupteur !! Bon OK, ça passe vite et on a tout juste le temps de voir ce qui se passe, nous allons donc encore ralentir la simulation pour que tout le monde (même un suisse ;-)) ait le temps de voir ce qu'il se passe.

Cette fois on va modifier le **PF** et le régler à 2000 (valeur maximum). Vous pouvez rester appuyé sur la petite flèche qui pointe vers le haut pour que la valeur augmente toute seule.

Nous sommes donc à 1 cycle exécuté toutes les 2 secondes, ce qui veut dire qu'on est ralenti 200 fois !! Avec un tel ralenti, il devient très facile d'observer ce qu'il se passe dans un programme qui a un bug.

Je vous laisse refaire la même opération que précédemment et observer le résultat qui est le même mais en plus lent.

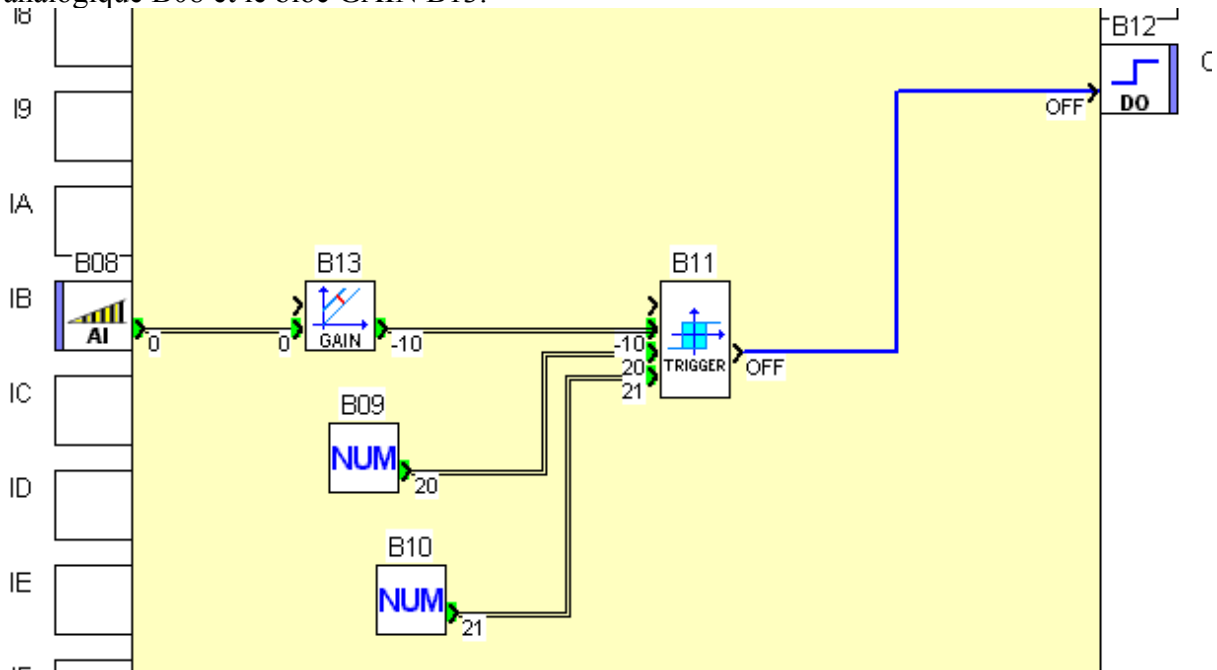
Petit point sur les actions fugitives :

Ce que vous venez de faire à plusieurs reprises sur un des liens de votre programme peut être fait sur N'IMPORTE quel lien de votre programme. C'est-à-dire que vous pouvez directement intervenir sur le lien qui connecte le bloc logique OU au bloc télérupteur, ou encore sur la sortie du timer qui arrête le télérupteur. Bien entendu l'action sur un lien ne provoque pas de changement en amont de la sortie du bloc ou le lien est connecté (si vous cliquez entre le OU et le télérupteur, cela ne va pas changer l'état de vos entrées).

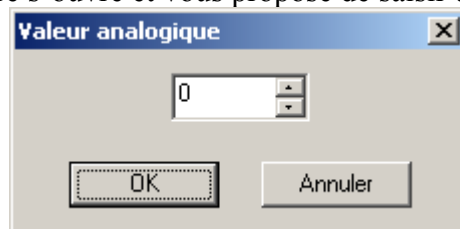
L'intérêt de ceci est de pouvoir tester des morceaux de programme en partant des sorties, pour voir à partir de où ça ne fonctionne plus lorsqu'on a un problème.

UN BUG DANS L'ATELIER M3: Si vous quittez la simulation, puis que vous la relancez, bien que la **PF** affiche la dernière valeur utilisée, celle-ci est mal renvoyée au simulateur, il faut juste la changer une fois puis la remettre à la valeur souhaitée pour qu'elle soit correctement prise en compte.

Alors on a vu les forçages fugitifs sur les liens TOR, attaquons maintenant les forçages sur les liens analogiques. Prenons donc notre petit « thermostat » et cliquons sur le lien entre l'entrée analogique B08 et le bloc GAIN B13.



Cette fois ci une petite fenêtre s'ouvre et vous propose de saisir une valeur

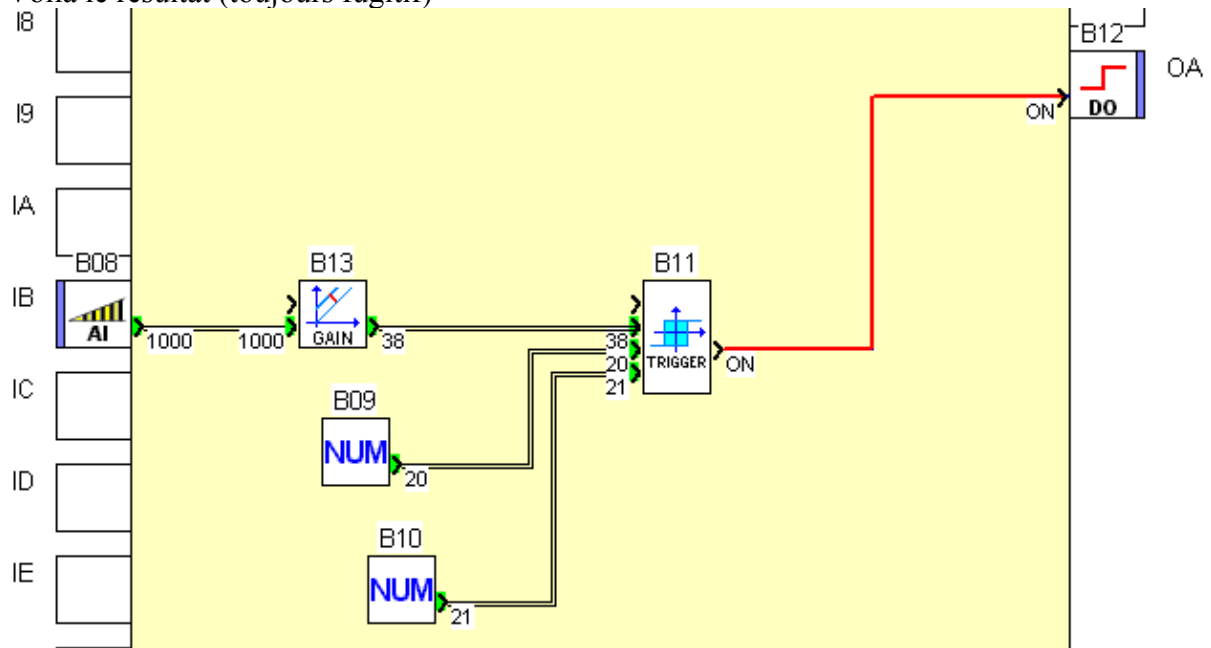


Petite anecdote de développeur que je ne peux pas me retenir de raconter, car aussi stupide que cela puisse paraître, cette petite fenêtre m'a valu une bonne tranche de rire alors que je travaillai encore au développement de l'atelier de programmation du M3. Donc pour la petite histoire, le nom de cette « classe » (pour les non développeurs, c'est le nom que porte

cette fenêtre dans le code) porte le joli nom de **CIhmDlgForcageAnal**. Oui oui, anal pour analogique bien sur... on s'amuse beaucoup nous les développeurs.

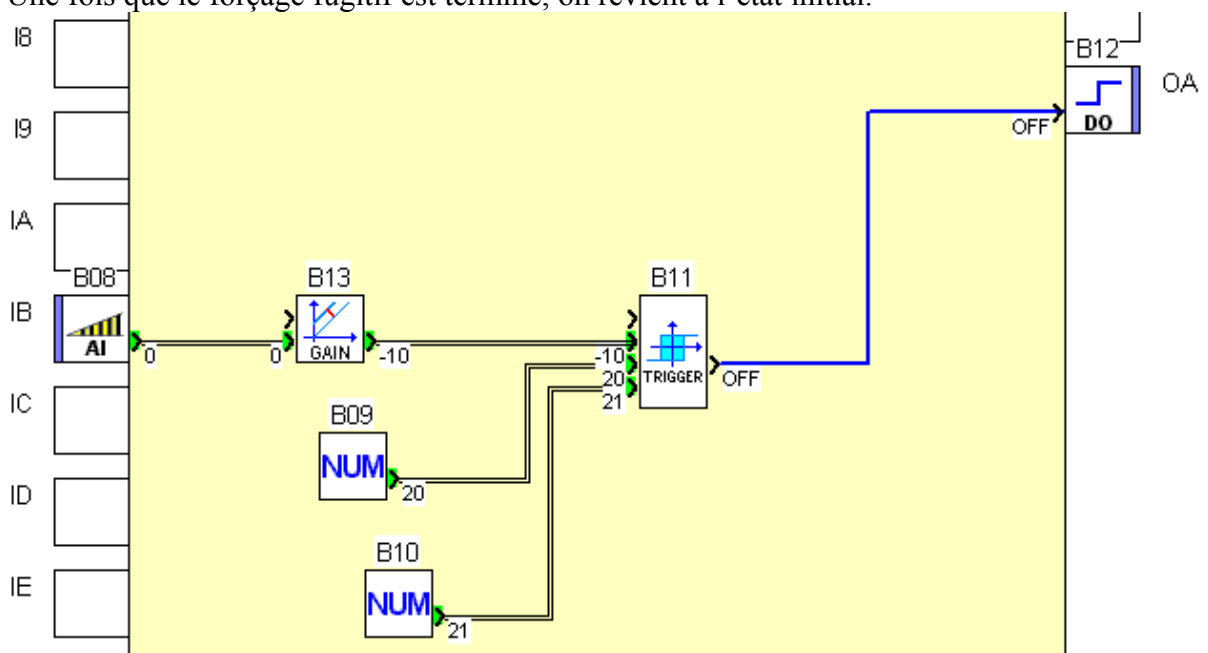
Donc dans cette petite fenêtre vous allez pouvoir saisir la valeur que vous voulez donner à votre lien analogique. Comme pour le lien TOR, cette valeur ne sera appliqué que de façon fugitive et on va donc profiter d'être toujours en mode ralenti pour bien observer ce qu'il va se passer. Saisissez la valeur 1000 et cliquez sur le bouton OK.

Voilà le résultat (toujours fugitif)



Alors là qu'est ce que nous avons de beau. L'entrée analogique envoie grâce au forçage la valeur 1000. Par l'intermédiaire du bloc GAIN qui nous convertis ceci en degré soit 38° (nous avons une sonde -10 → +40°). Ainsi nous sommes passés à une température supérieure à notre consigne de « Arrêt à marche » (B10 qui vaut 21) Ceci à donc provoqué le changement d'état de la sortie du Trigger qui est passé à ON.

Une fois que le forçage fugitif est terminé, on revient à l'état initial.



Allez, vous pouvez essayer de faire la même opération sur le lien qu'il y a entre le bloc GAIN B13 et le bloc Trigger B11. Jusqu'ici rien de très nouveau.

Par contre essayez donc de cliquer sur un des deux liens en sortie de B09 ou de B10... comment ca !! Ca ne marche pas... !!

Alors sachez que c'est tout à fait normal et cela me permet d'introduire un concept qu'il est important de comprendre si on souhaite utiliser la programmation de la face avant dans le futur, c'est le concept du **paramètre sortie** (non il ne manque pas mot).

Alors il faut bien faire la différence entre une sortie, et un paramètre sortie !!

Une sortie de bloc est calculée en fonction de sa (ses) entrée(s), alors qu'un paramètre sortie est un paramètre qu'on retrouve en sortie du bloc et qui est FIXE à l'exécution (seul vous, par l'atelier ou par la programmation de la face avant peuvent modifier sa valeur)!!

En fait, on en a déjà vu avant...souvenez vous des paramètres du timer, le retard marche et le retard arrêt... et bien ces valeurs aussi son des paramètres sortie !!

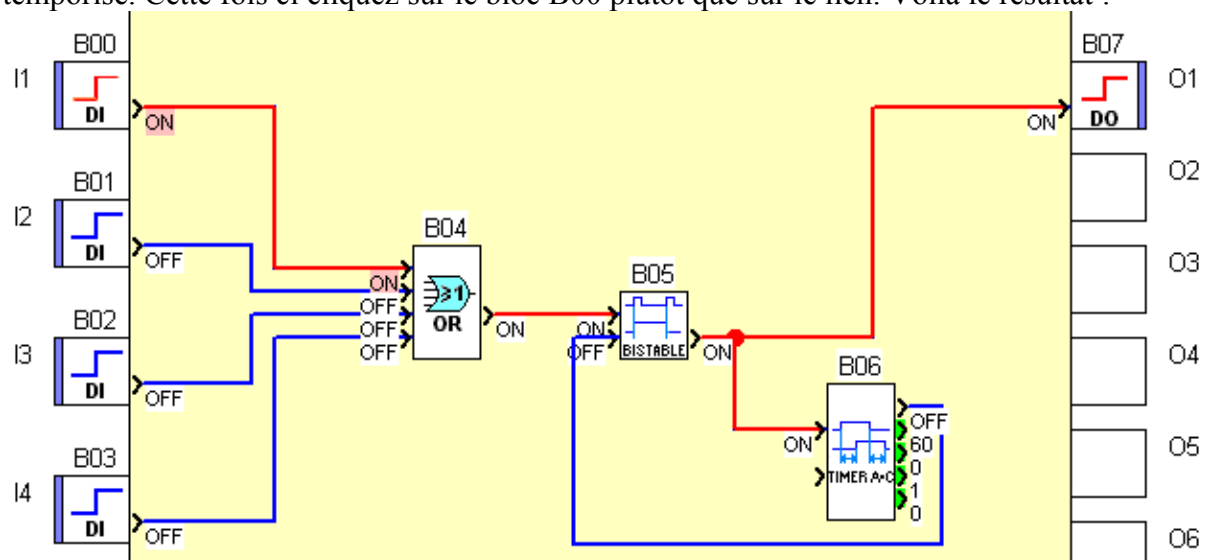
Si vous n'avez pas tout compris de suite, ce n'est pas grave pour le moment et vous avez le temps pour appréhender ce concept.

Pour finir cette partie de la vitesse de simulation, dans notre cas nous avons ralenti les choses pour bien observer comment ca marche, mais il est également possible d'accélérer la simulation jusqu'à 25 fois. Cela peut parfois s'avérer utile, libre a vous de tester ceci avec le timer que du télérupteur temporisé (pour ne pas à avoir une minute que celui-ci s'active par exemple)

Allez, on va voir maintenant la suite des actions que l'on peut faire en simulation.

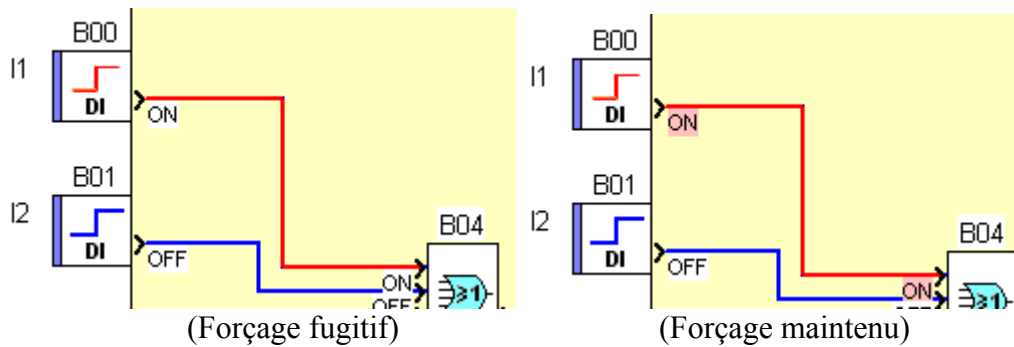
Les forçages maintenus

Alors pour cette nouvelle action en simulation, nous allons revenir sur notre télérupteur temporisé. Cette fois ci cliquez sur le bloc B00 plutôt que sur le lien. Voici le résultat :



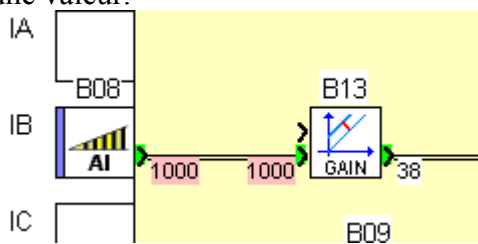
Alors en apparence c'est la même chose que tout à l'heure... mais en dehors du fait que ca ne disparaît pas avec le temps (l'état est maintenu) si vous êtes observateurs vous avez du remarquer autre chose....

Allez ! Je vous aide, comparez ces deux images :



Eh oui la ruse est là... le couleur de fond du texte ON n'est pas la même !! Eh oui ce petit fond (de) rosé ca change tout (hips). Bref, en dehors de toute plaisanterie ceci indique que la valeur est forcée et ne changera pas d'état sans action de votre part. (Petite note d'humour... malgré cette association de forçage et rosé... il ne faut pas forcer sur le Rosé... re hips)
 Si vous recliquez sur l'entrée B00, on repasse à OFF mais toujours à l'état forcé !! Pour relâcher un forçage, il faut faire un clique droit sur le lien correspondant et choisir l'action « Relâcher » ou « Relâcher tout ». L'action relâcher va juste désactiver le forçage sur le lien ou vous avez cliqué, alors que relâcher tout va relâcher TOUT les forçages courant dans votre programme.

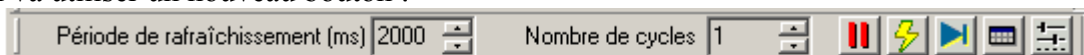
Les actions de forçages maintenu sur les liens analogique est aussi possible et voici un exemple. Pour le mettre en œuvre le principe est le même cliquez sur le bloc B08 et saisissez une valeur.



Pour en finir avec tout ca, faites un clique droit quelque part, et sélectionnez relâchez tout (il est important de le faire pour la partie suivante de ce tutoriel)

La simulation des entrées analogiques

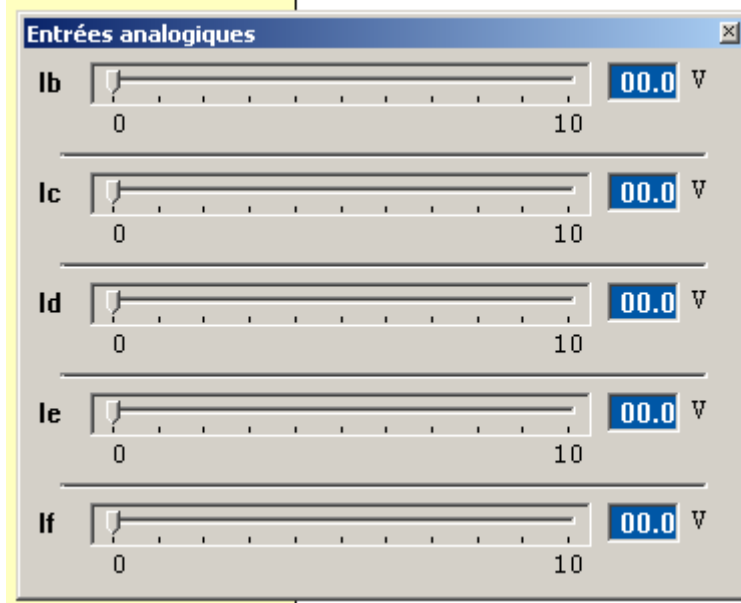
Alors les forçages fugitifs et maintenu, c'est bien marrant, mais pour simuler rapidement des variations de température on doit quand même pouvoir faire mieux !! Eh oui !! Pour ca il y a **la simulation des entrées analogiques**. Allez ! On reprend notre barre d'outils de simulation, et on va utiliser un nouveau bouton :



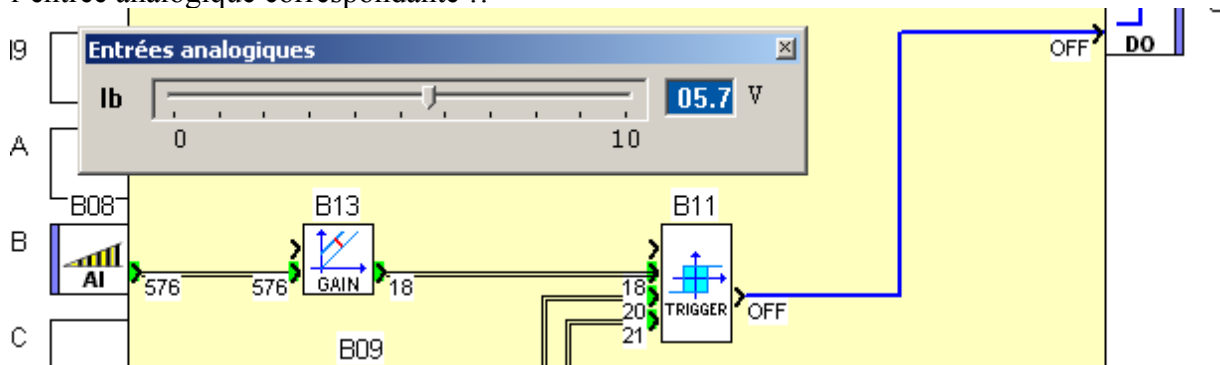
Ce petit bouton anodin permet de faire apparaître une petite fenêtre bien pratique : la fenêtre de simulation des entrées analogiques :



Alors dans notre exemple, elle n'a qu'une seule ligne car nous avons utilisé qu'une seule des 6 entrées analogiques de notre M3 (elle détecte automatiquement quelles entrées sont utilisées et crée les lignes correspondantes). Si nous avons utilisé les 6 entrées analogiques, elle ressemblerait à cela :



Comment ça marche ? C'est très simple, déplacez les curseurs afin de faire varier la valeur de l'entrée analogique correspondante !!



Avec ça, il devient très simple de régler les valeurs simulées.

Petite astuce clavier : Cliquez sur le curseur et ensuite utilisez les flèches droites/gauche ou haut/bas du pavé de directionnel du clavier pour changer finement la valeur. Vous pouvez rester appuyé sur une touche afin que la valeur augmente ou diminue en continu.

Alors je vous ai parlé bien plus tôt d'un problème que nous allons avoir avec le paramétrage de nos blocs constantes, souvenez vous :

« Je vais être directe sur ce point, ce paramétrage va poser un petit problème mais le mettre en évidence va permettre de vous faire comprendre que lorsqu'on programme une régulation avec des entrées analogiques, il faut prendre en considération les légères fluctuations électriques qui peuvent aboutir à des phénomènes d'oscillation d'état dans votre programme. »

En utilisant l'astuce clavier que je viens de vous citer, approcher la valeur d'entrée de sorte que vous êtes à la limite entre 20° et 21° en sortie du bloc gain (valeur de l'entrée analogique = 634). Toujours avec le clavier, faite osciller la valeur de l'entrée analogique entre 634 et 635. On observe que la sortie du Trigger oscille entre les états ON et OFF !!

Alors en quoi cela est un problème. Bah imaginez que vous avez une toute petite oscillation de votre entrée, et pensez aux conséquences... votre relai piloté en sortie va lui aussi osciller et ça il ne va pas beaucoup aimer (et vos oreilles non plus au passage). Tout cela sert à vous expliquer une chose : quand on utilise un trigger, le delta entre les valeurs de MARCHE A ARRET et d'ARRET A MARCHE doit toujours être supérieur ou égale à 2. En effet, le Trigger change d'état dès que la valeur d'entrée atteint la consigne correspondante à un changement d'état, et en conséquence, avoir un delta de 1 revient à utiliser un bloc de comparaison simple. On perd donc tout l'intérêt du trigger.

Modifiez la valeur du paramètre du bloc B10 pour le passer à la valeur 22. Et refaites le même test d'oscillation de valeur... cette fois-ci plus de problème !

Voilà tout pour la simulation des entrées analogiques.

Conclusion de cette première phase du tutoriel :

Dans cette première phase du tutoriel « Le M3 pour les nuls », vous avez donc appris à manipuler l'atelier logiciel M3 dans ses principales fonctions qui sont l'édition de programme, et la simulation. Vous avez maintenant toutes les bases pour partir à la conquête du monde M3 en attendant que je rédige les prochaines phases de ce tutoriel.

Posez différents blocs, câblez les ensemble, passez en simulation, utilisez l'aide en ligne (le bouton « ? » dans les fenêtres de paramétrage), et faites preuve de curiosité car si rien ne vous a semblé insurmontable dans cette première phase, vous êtes parfaitement capable de vous lancer dans la création de votre programme M3.

Le chapitre qui suit va vous permettre de programmer avec plus de fluidité, vous pouvez très bien vous en passer, mais croyez-moi il n'est pas inutile.

L'utilisation avancée de la souris et des raccourcis clavier

Alors, je sais que ce chapitre n'est pas essentiel à l'utilisation de l'atelier M3, néanmoins j'estime que de savoir utiliser quelques combinaisons de touches permet d'aller plus vite pour beaucoup de choses et cela vous facilitera la vie dans votre programmation, et vous fera sans doute gagner du temps..

Dupliquer un bloc

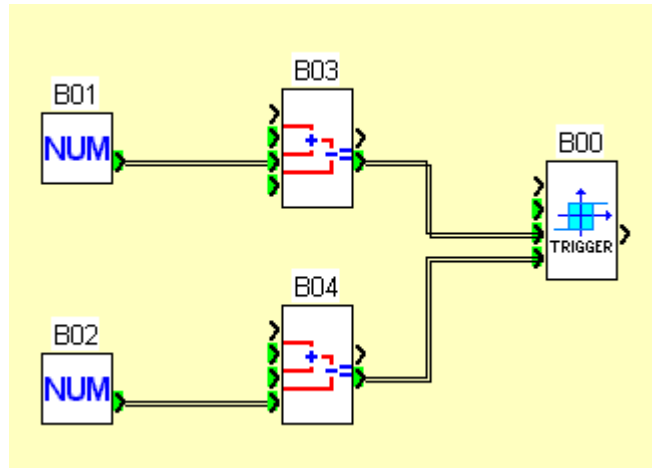
Premier truc super utile, celui-là, mémorisez-le, car de tout ce que je vais vous donner, c'est à mon sens le plus utile. Ceci permet de créer une copie d'un bloc existant.

Maintenez enfoncée la touche contrôle (Ctrl) de votre clavier, et faites la même opération que pour déplacer un bloc (glisser/déposer). Au moment où vous aurez relâché le bouton de la souris, au lieu d'avoir déplacé le bloc, l'atelier en aura créé un nouveau.

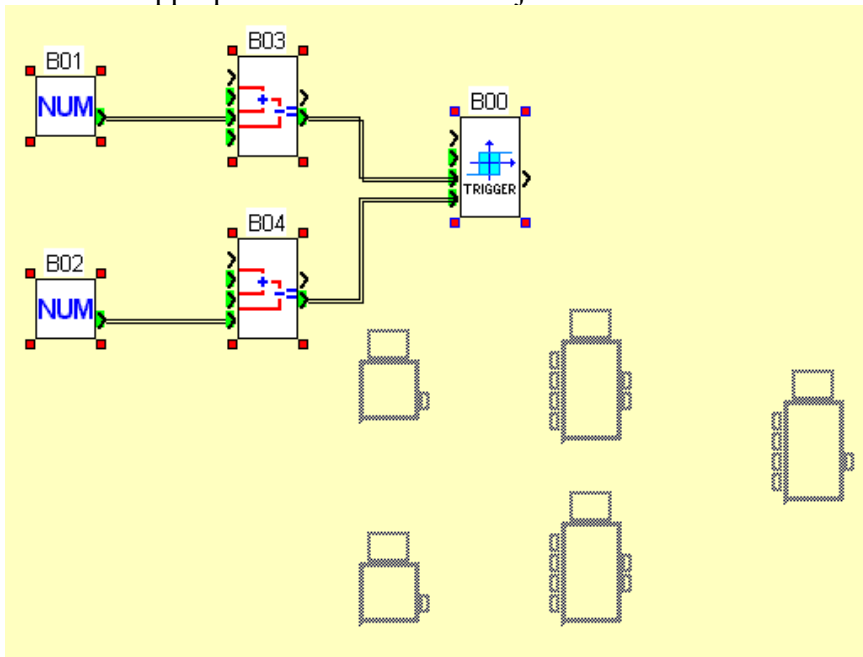
Quel est le principal intérêt de ceci? Imaginons que vous avez besoin d'ajouter un bloc que vous avez déjà en plusieurs exemplaires dans votre programme, eh bien utiliser cette méthode sera plus rapide que d'aller chercher le bloc dont vous avez besoin dans la barre de fonction.

Encore plus fort:

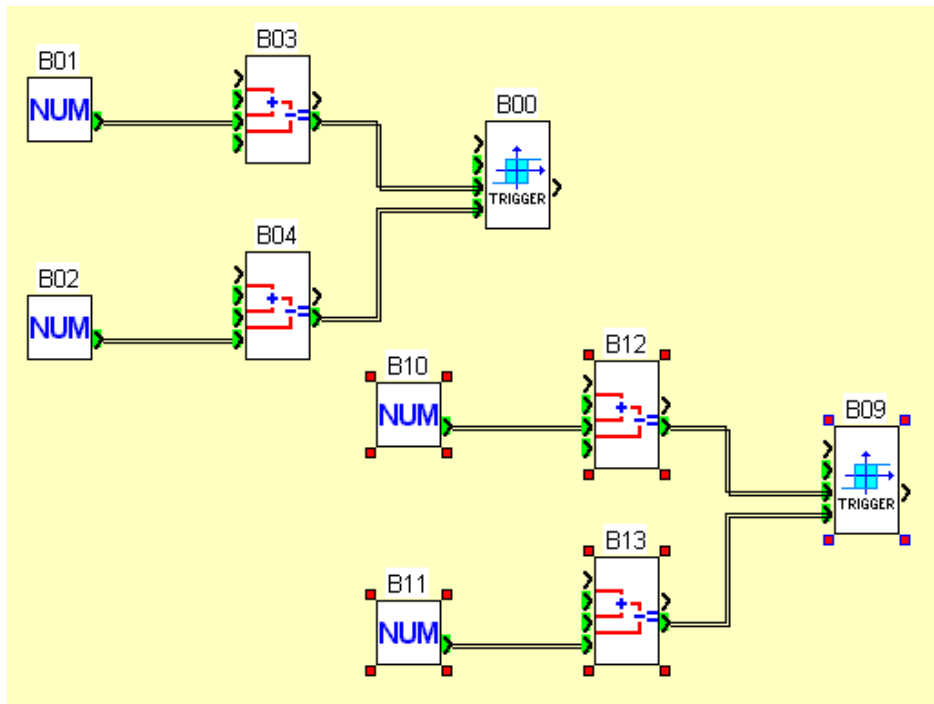
Prenons ce groupe de bloc qui est assez courant dans un programme de régulation solaire:



Sélectionnez les tous et appliquez la méthode décrite juste avant:



Une fois le bouton de la souris relâché, vous obtenez une copie complète de votre groupe de bloc AVEC les liens.



Voilà, grâce à cette méthode très simple, vous avez économisé 8 actions : 5 actions pour aller chercher les blocs et 4 pour tracer les liens, et vous en avez fait qu'une seule.

NOTE : Cette technique pour copier des objets (fichiers) est standard sous Windows ou Linux (ou Mac OS je ne sais pas)

Sélection multiple avec MAJ

Cette fois ci, c'est nettement plus simple que l'astuce précédente, Maintenez le bouton « MAJ » (que j'appelle souvent « Shift »), et cliquez l'un après l'autre sur les blocs que vous souhaitez sélectionner. Chaque nouveau bloc sur lequel vous cliquez s'ajoute à la sélection courante. Si vous avez cliqué par erreur sur un bloc, cliquez à nouveau sur celui ci tout en maintenant la touche MAJ pour l'enlever de la sélection.

Couper, Copier, Coller

Tous ceux qui utilisent assez régulièrement un ordinateur connaissent les raccourcis claviers, pour ceux qui ne les connaissent pas ce sont des combinaisons de touches standard dans qui beaucoup de logiciels permettent de faire certaines opérations présentes dans les menus. Ici on va s'intéresser au couper/copier/coller.

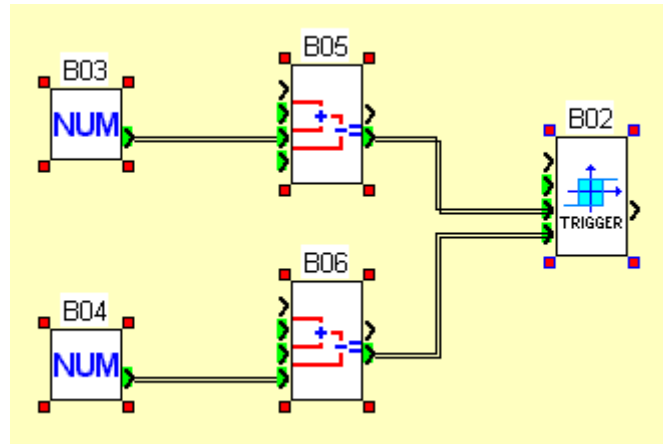
Les raccourcis claviers sont les suivants :

Couper = Ctrl + X

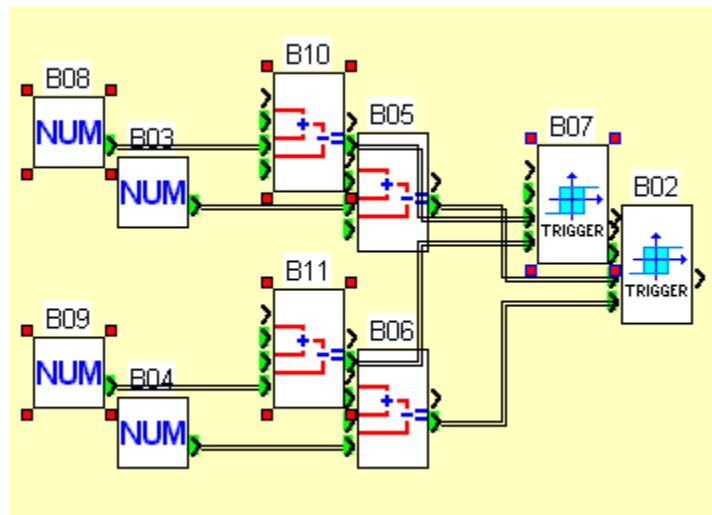
Copier = Ctrl + C

Coller = Ctrl + V

Reprenons notre exemple de la duplication de bloc :



Une fois les blocs sélectionné, pressez la combinaison de touche Ctrl + C, puis la combinaison de touche Ctrl + V.
Voilà le résultat



Alors oui cela ressemble à un joyeux cafouillage, mais les blocs qui viennent d'être posé sont sélectionné il ne reste donc plus qu'à les déplacer.

Quel est l'intérêt par rapport à la duplication de bloc ?

L'intérêt de cette manœuvre c'est qu'elle peut être réalisée depuis un autre atelier M3 ayant un programme ouvert, ou depuis un programme en import dans l'atelier M3 courant. Ça permet donc d'importer des bouts de programme existant assez simplement (les programmes des membres APPER)

NOTE : j'ai parlé dans ce cas des raccourcis clavier, mais cela peut également être réalisé à l'aide des commandes de menu « Edition → Couper », « Edition → Copier », et « Edition → Coller », commandes également disponibles dans le menu contextuel (celui qu'on fait apparaître avec le clic droit de la souris).

Merci d'avoir parcourut ce tutoriel et merci à l'équipe de correction grammaticale et orthographique qui a ouvré pour que ce document soit le plus propre possible.

A bientôt pour les prochaines phases du tutoriel « le M3 pour les nuls »