

Le Millenium 3 pour les nuls !!

Phase 3 : Organiser son programme.

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

Introduction :

Pour ceux qui ont bien suivi les deux premiers tutoriaux, vous savez maintenant manipuler correctement l'atelier de programmation, et vous avez une idée de ce que fait chaque bloc fonction disponible dans l'atelier. Avec ces éléments vous avez pu vous lancer dans la création de vos propres programmes. Quand on commence à programmer, il faut toujours penser qu'un jour il faudra retoucher le programme.

Cette nouvelle phase a donc pour objectif de vous présenter les outils mis à disposition dans l'atelier de programmation du M3 afin d'organiser votre programme ainsi que les bonnes pratiques à avoir pour que dans 6 mois, 1 an ou plus, quand vous modifierez votre installation solaire, vous retombiez rapidement sur vos pieds en redécouvrant votre programme.

Table des matières

•	<u>Les outils d'organisation du programme</u>	<u>3</u>
➤	<u>Outils de dessin</u>	<u>3</u>
	<u>La barre d'outils de dessin</u>	<u>3</u>
	<u>Généralités sur les objets de dessin</u>	<u>3</u>
	<u>Le trait</u>	<u>4</u>
	<u>Le rectangle plein</u>	<u>4</u>
	<u>L'ellipse pleine</u>	<u>4</u>
	<u>Le texte</u>	<u>5</u>
	<u>L'image</u>	<u>5</u>
	<u>La largeur des bordures</u>	<u>6</u>
	<u>La couleur des bordures</u>	<u>7</u>
	<u>La couleur de remplissage</u>	<u>7</u>
➤	<u>Grouper les blocs</u>	<u>8</u>
➤	<u>Un bref mot sur les MACROs</u>	<u>9</u>
•	<u>Les bonnes pratiques</u>	<u>11</u>
➤	<u>Utiliser les outils de dessin pour représenter une fonction</u>	<u>11</u>
	<u>Placer un rectangle pour « grouper » visuellement les blocs</u>	<u>11</u>
	<u>Utiliser des traits pour séparer des parties de programmes qui n'ont rien à voir</u>	<u>14</u>
➤	<u>Mettre des commentaires</u>	<u>14</u>
	<u>Sur les blocs fonction eux-mêmes</u>	<u>15</u>
	<u>A l'aide des objets texte</u>	<u>15</u>
➤	<u>Utiliser des images</u>	<u>16</u>
	<u>Sur les entrée ou sorties TOR</u>	<u>16</u>
	<u>Dans la feuille de câblage</u>	<u>17</u>
➤	<u>Mettre des liens texte</u>	<u>18</u>
•	<u>Un exemple sur un programme des membres APPER :</u>	<u>19</u>
➤	<u>La première étape pour y voir plus clair : les liens textes</u>	<u>19</u>
•	<u>Conclusion</u>	<u>27</u>

Les outils d'organisation du programme

Outils de dessin

La barre d'outils de dessin

L'ensemble des outils de dessin sont accessibles depuis le menu dessin, et du menu outils, mais également depuis la barre d'outils de dessin représenté ci-dessous.



On trouve dans l'ordre de gauche à droite les fonctions suivantes :

- Le trait
- Le rectangle plein
- L'ellipse
- Le texte
- L'image
- La largeur des bordures
- La couleur des bordures
- La couleur de fond.
- Aligner les objets à gauche
- Aligner les objets à droite
- Aligner les objets en haut
- Aligner les objets en bas
- Centrer les objets horizontalement
- Centrer les objets verticalement
- Répartir les objets horizontalement
- Répartir les objets verticalement
- Amener les objets au premier plan
- Amener les objets à l'arrière plan
- Grouper les objets
- Dissocier le groupe

D'un point de vue tout à fait personnel, les fonctions de positionnement (aligner, centrer, répartir) n'ont pas un grand intérêt dans l'atelier M3 car, dans presque 100% des cas, on utilisera plus facilement la grille qui permettant d'aligner facilement les blocs fonction au moment où ils sont posés ; c'est pourquoi je ne m'étendrai pas sur ces fonctions (d'autant plus que leurs noms sont suffisamment explicites). Si la grille est désactivée, il faut passer par le menu « Affichage → Grille » pour afficher/masquer la grille, ou changer sa taille,

Généralités sur les objets de dessin

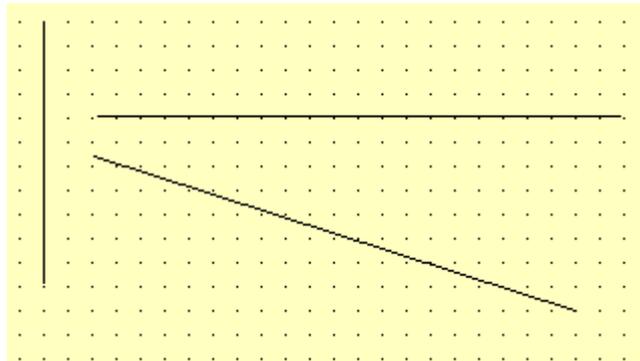
Tous les objets de dessin peuvent être redimensionnés à volonté après avoir été placés dans la surface de dessin. Pour cela, ils disposent de « poignées » permettant de les redimensionner en largeur, hauteur ou les deux à la fois.

Tous les objets de dessin se placent à la manière d'un glissé/déposé, c'est-à-dire en pressant le bouton gauche pour placer le début de l'objet, et en relâchant le bouton pour définir la fin de l'objet.

Le trait

C'est un simple trait de base, comme celui que l'on peut dessiner sous Paint.

Pour placer un trait, cliquez sur le bouton « trait » dans la barre d'outils de dessin puis dessinez-le dans la surface de câblage.

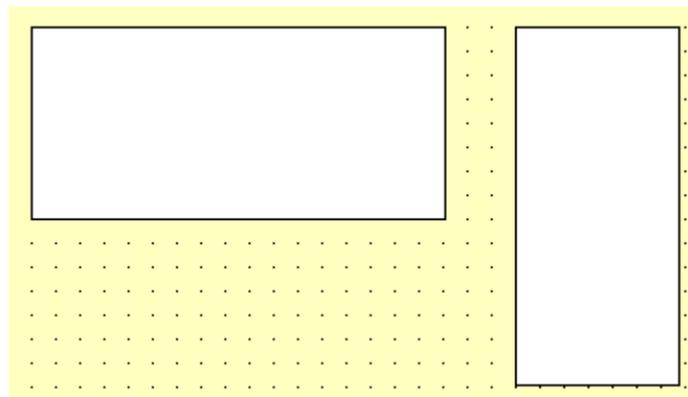


Exemples de traits

Le rectangle plein

Comme son nom l'indique, c'est un rectangle plein avec une bordure.

Pour placer un rectangle, cliquez sur le bouton « rectangle » dans la barre d'outils de dessin puis dessinez-le dans la surface de câblage.

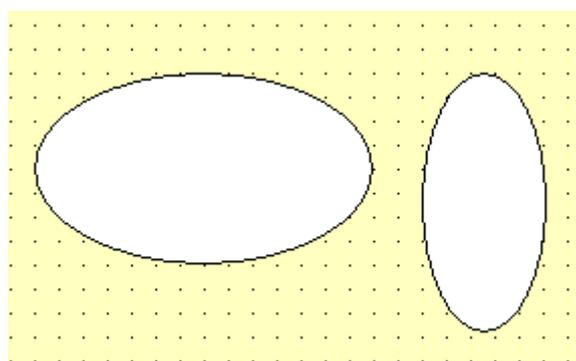


Exemples de rectangles

L'ellipse pleine

Comme pour le rectangle, c'est un objet plein en forme d'ellipse avec une bordure.

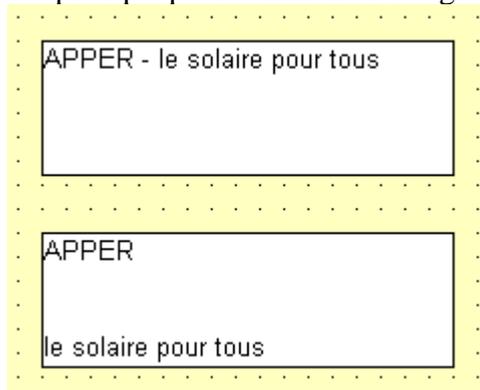
Pour placer un rectangle, cliquez sur le bouton « rectangle » dans la barre d'outils de dessin puis dessinez-le dans la surface de câblage.



Exemples d'ellipses

Le texte

Le texte fonctionne comme un rectangle sauf qu'en plus il offre la possibilité d'écrire du texte à l'intérieur. C'est objet est très pratique pour commenter des groupes de blocs.



Exemples de textes

Pour éditer le texte présent dans un objet texte, il suffit de double cliquer dessus.

Alors ceux qui ont déjà utilisés l'objet texte se demandent comment j'ai fait pour mettre du texte sur plusieurs lignes. Eh oui ! l'objet texte ne fait pas automatiquement de retour à la ligne, ce qui peut être très gênant si on aime la prose (comme moi), ou si tout simplement on a beaucoup de choses à expliquer sur une fonction ou un paramétrage un peu complexe. Pour faire un retour à la ligne lorsqu'on est occupé à éditer le texte, il suffit d'utiliser la combinaison de touche Ctrl+Entrée.

L'image

Dans le précédent tutoriel, je vous avais fait une note sur l'utilisation des images, et c'est ici qu'elle prend tout son sens.

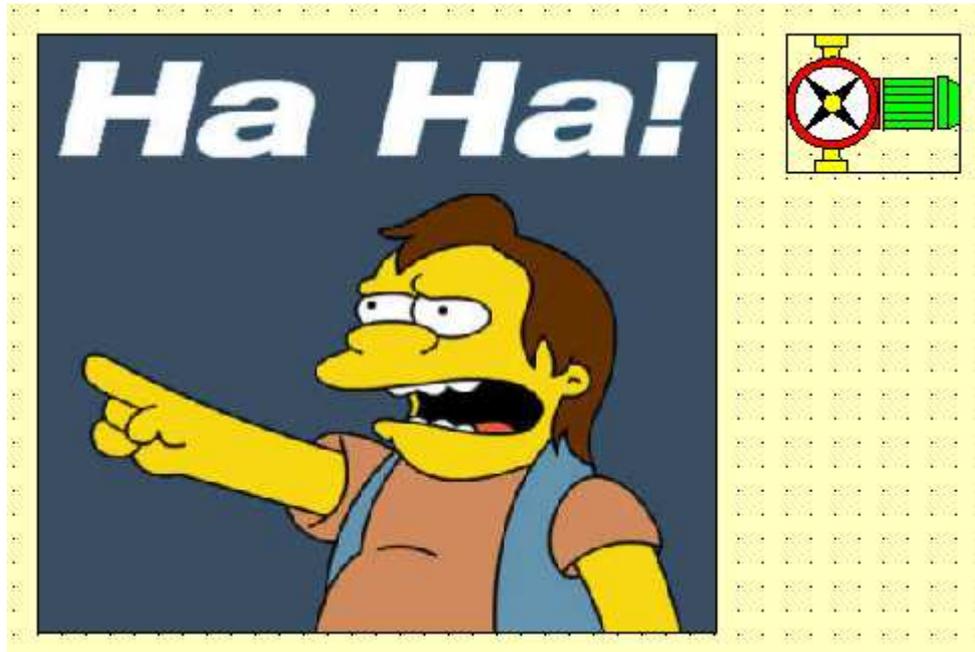
Petit rappel :

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.

Avec l'objet image, vous avez la possibilité de caser de jolies photos, en grand format, mais vous devrez toujours faire attention au volume du fichier.

Pour placer une image, cliquez sur le bouton image. Vous devez alors sélectionner votre fichier. Une fois votre fichier sélectionné, vous pouvez placer votre image dans la surface de dessin. Notez que contrairement aux autres objets, il prend automatiquement la taille de votre image.

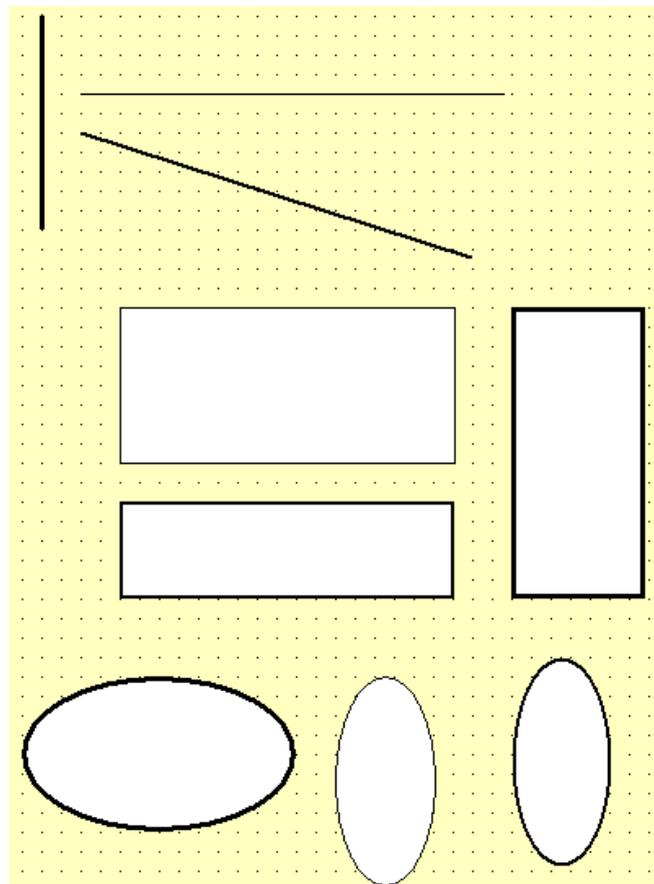


Exemple d'images

La largeur des bordures

Sur tous ces objets, il est possible de modifier la largeur des bordures.

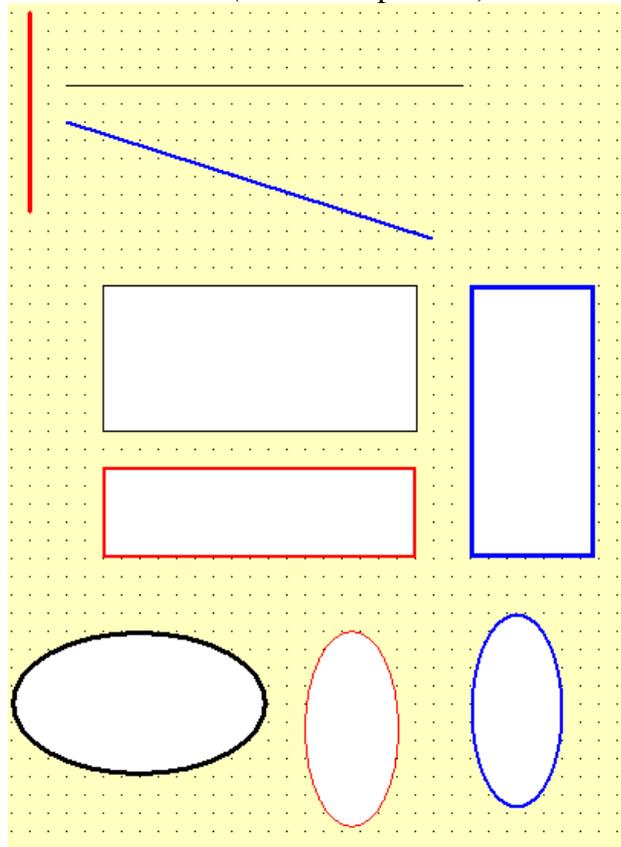
Pour cela il faut simplement sélectionner l'objet dont on souhaite modifier les bordures, puis de cliquer sur la flèche à droite de l'icône bordure dans la barre d'outils de dessin et de sélectionner la largeur que l'on souhaite. Ceci peut très bien être effectué sur une sélection multiple d'objets.



Exemples avec différentes tailles de bordures

La couleur des bordures

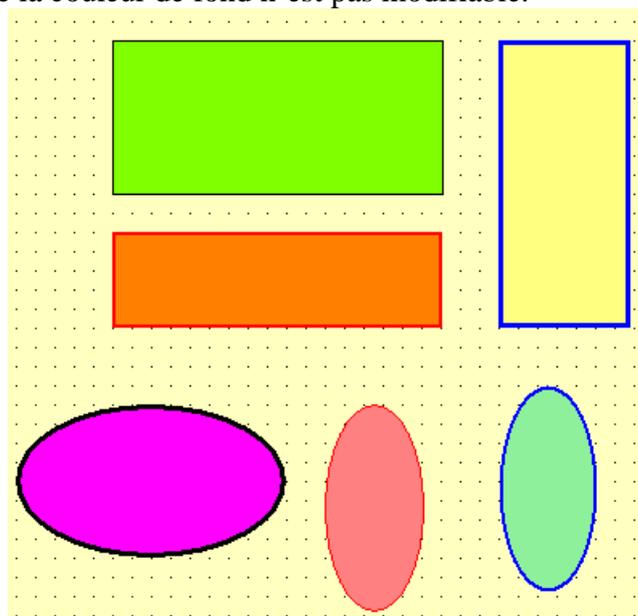
De la même manière que l'on peut modifier la largeur des bordures, il est possible de changer la couleur de ces bordures. Le principe est le même que pour la largeur mais cette fois on utilisera le bouton couleur des bordures (l'icône du pinceau)



Exemples avec différentes couleurs de bordures

La couleur de remplissage

Toujours dans la même série, il est possible de changer la couleur de fond de ces objets. Pour le trait et l'image la couleur de fond n'est pas modifiable.

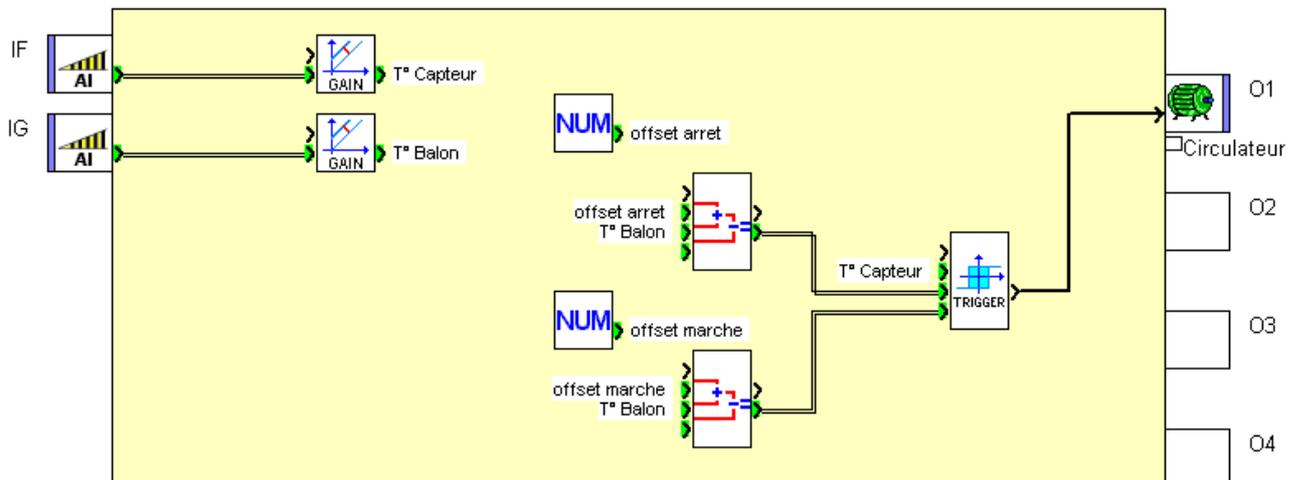


Exemples avec différentes couleurs de fond

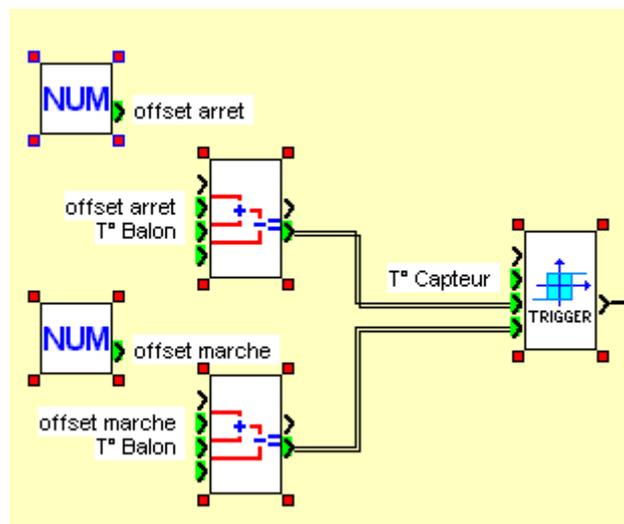
Grouper les blocs

Voici une fonction qui s'avère très pratique une fois que l'on a défini des fonctions bien précises, et que l'on souhaite les organiser sans avoir à se compliquer la vie avec des sélections multiples de nombreux blocs.

Prenons par exemples le programme suivant, qui comprend l'ensemble des blocs du système de trigger que nous avons vu dans la phase 1 du tutoriel.



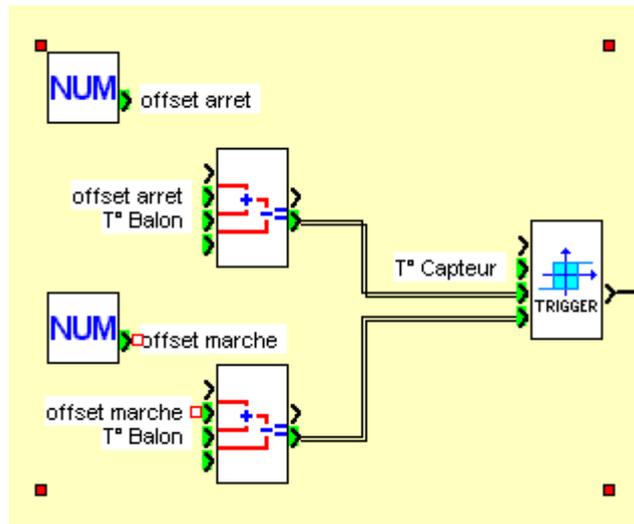
L'ensemble qui nous intéresse ici est celui comprenant le trigger, les deux blocs ADD/SUB, et les deux blocs constantes numériques. Si on souhaite les déplacer, nous devons, à chaque fois, sélectionner l'ensemble des blocs pour ensuite les déplacer en groupe.



Ici les blocs sont tous indépendants

Alors c'est sur, cette méthode marche bien, mais il est fort probable (certain) que dans une régulation solaire complète, on retrouve plusieurs fois cet ensemble, et il y a donc tout intérêt à les grouper. Lorsque tous les blocs sont sélectionnés, utilisez le bouton grouper de la barre d'outils (avant dernier en partant de la fin).

Le changement d'aspect n'est pas flagrant, mais faites plus attention aux petites poignées rouges qui sont normalement positionnées aux coins de chaque bloc :



Eh oui, tous ces blocs sont considérés comme un unique ensemble, et il suffit donc de cliquer sur un seul d'entre eux pour tous les sélectionner simultanément. De la même manière, maintenant qu'ils sont groupés, si vous déplacez un des blocs de ce groupe, les autres vont suivre automatiquement. Ceci s'avère très pratique lorsqu'on commence à organiser son programme car, en commençant par créer des groupes, il ne reste alors qu'à les agencer. Ceci est nettement plus simple car on n'a pas à faire de sélections multiples à chaque fois.

Un bref mot sur les MACROS

Je vais passer très vite sur le sujet des MACROS car elles sont incontestablement le meilleur moyen qui existe pour organiser son programme et pour en simplifier l'aspect général. Mais nous y reviendrons dans la dernière phase du tutoriel car les MACROS sont une forme avancée de la programmation qui ne font pas partie de l'objectif de cette phase du tutoriel.

Sélectionnez le groupe créé précédemment, faites un clic droit sur l'un des blocs et choisissez la commande « Créer une MACRO »

La fenêtre suivante s'affiche :

Configuration de la MACRO

Identifiant de MACRO (1 à 5 caractères)

Nom de la MACRO

Symbole du bloc

Image Standard

Image Personnalisée

OK Annuler ?

Nom des entrées

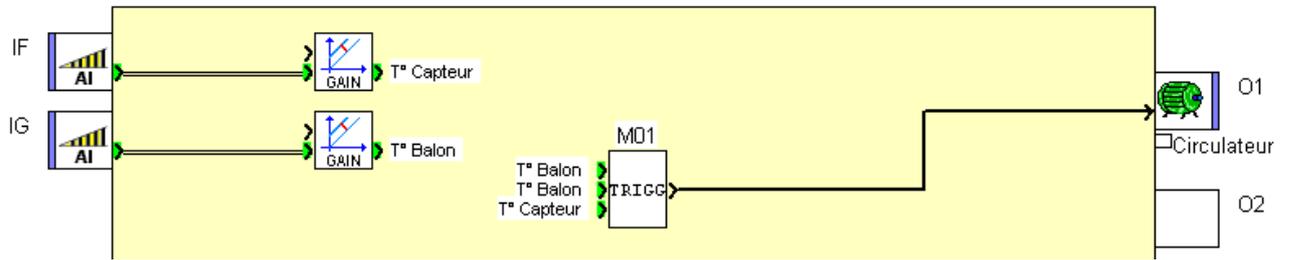
N°	Bloc et nom	Label
1	B04 Entrée 2	Entrée 2
2	B05 Entrée 2	Entrée 2
3	B06 Valeur à comp	Valeur à comparer

Nom des sorties

N°	Bloc et nom	Label
1	B06 Sortie	Sortie

Dans cette fenêtre, contentez-vous simplement de saisir quelque chose dans le champ « Identifiant de MACRO » puis validez avec OK.

Voici le résultat de la création d'une MACRO :



Alors oui ! le programme est nettement plus simple visuellement, mais vos blocs n'ont pas été supprimés pour autant (en fait ils font maintenant partie de la MACRO). J'ai présenté ici les MACRO pour leur intérêt visuel au niveau de l'organisation du programme. Mais, comme je l'ai dit précédemment, je ne m'attarderai pas plus longtemps sur ce sujet pour le moment (notamment sur le fait que la macro aie deux fois la même entrée, alors que ce n'est pas indispensable, et encore plein d'autres choses...)

Les bonnes pratiques

En tant que développeur logiciel, ce que je vais tenter de vous expliquer ici est quelque chose qui m'a été enseigné depuis bien longtemps. Essayez de regrouper de façon logique un ensemble d'instruction (blocs fonction) sous forme de fonction (groupes de blocs) afin de pouvoir facilement faire évoluer ou modifier dans son coin chaque fonction sans que cela n'impacte le reste du programme.

Cette partie est forcément un peu subjective, puisque je vais vous présenter ma façon de grouper les fonctions, ce qui forcément correspond à ma logique personnelle et mes habitudes de programmation du M3.

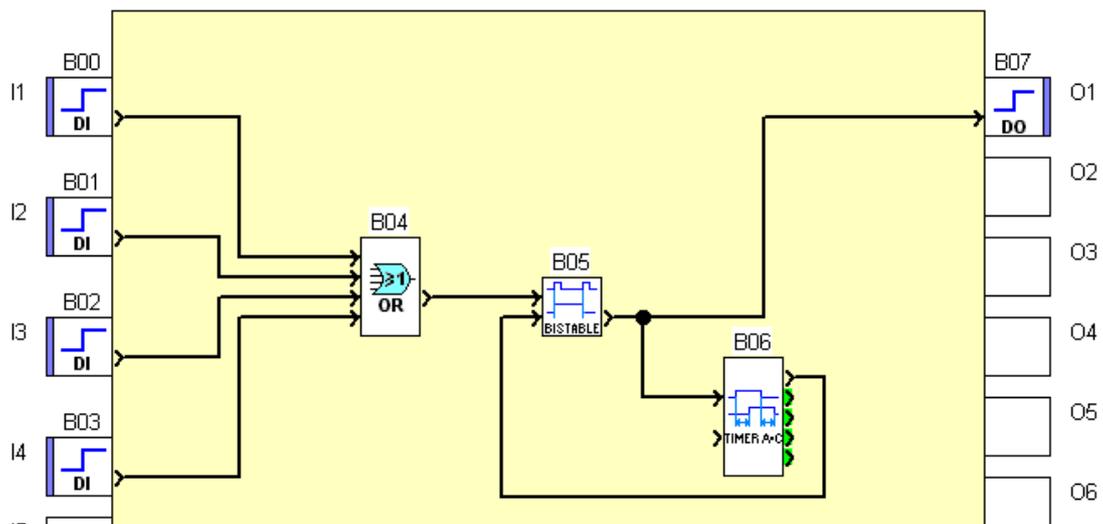
Utiliser les outils de dessin pour représenter une fonction

Placer un rectangle pour « grouper » visuellement les blocs

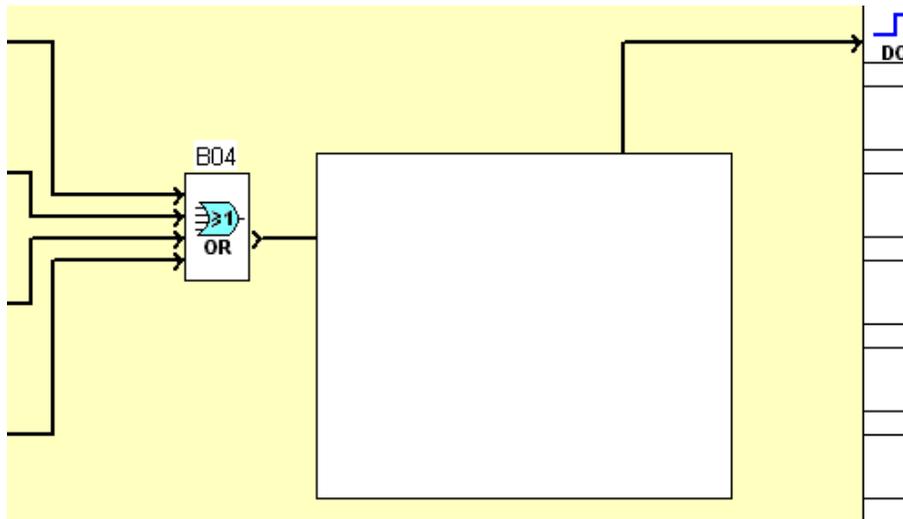
Mon truc « classique » pour visuellement isoler un groupe de bloc et les marquer comme étant une « fonction » c'est d'utiliser un rectangle placé en arrière plan sous ce groupe de bloc.

Pour démo, j'ai repris l'exemple du tutoriel phase 1 avec le télérupteur temporisé, et la régulation simple.

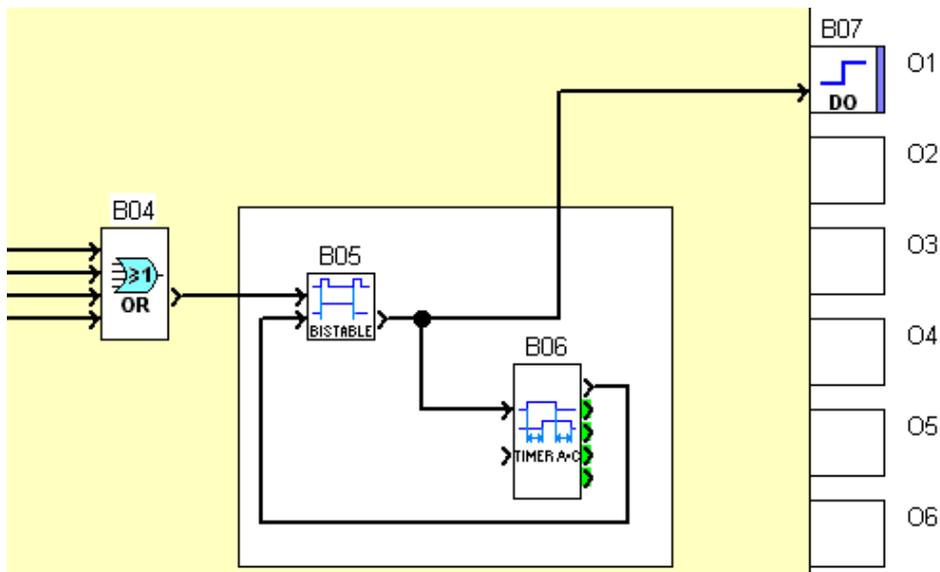
Commençons par la partie télérupteur temporisé :



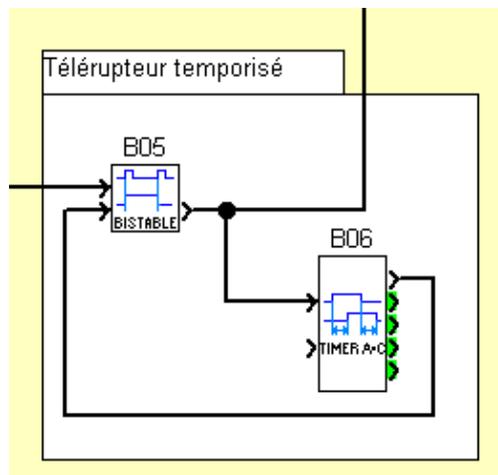
Ici on peut identifier la fonction « télérupteur temporisé » qui est donc le groupe des blocs B05 et B06. Tout d'abord on groupe les deux blocs ensemble, puis on pose par-dessus un rectangle simple....



Aïe, le rectangle se pose devant...pas de panique ! il suffit de le placer en arrière plan pour retrouver nos blocs fonction. Pour cela, faite un clique droit sur le rectangle et utilisez la commande du menu contextuel « Ordre → Mettre à l'arrière plan », voila ce qu'on obtient :

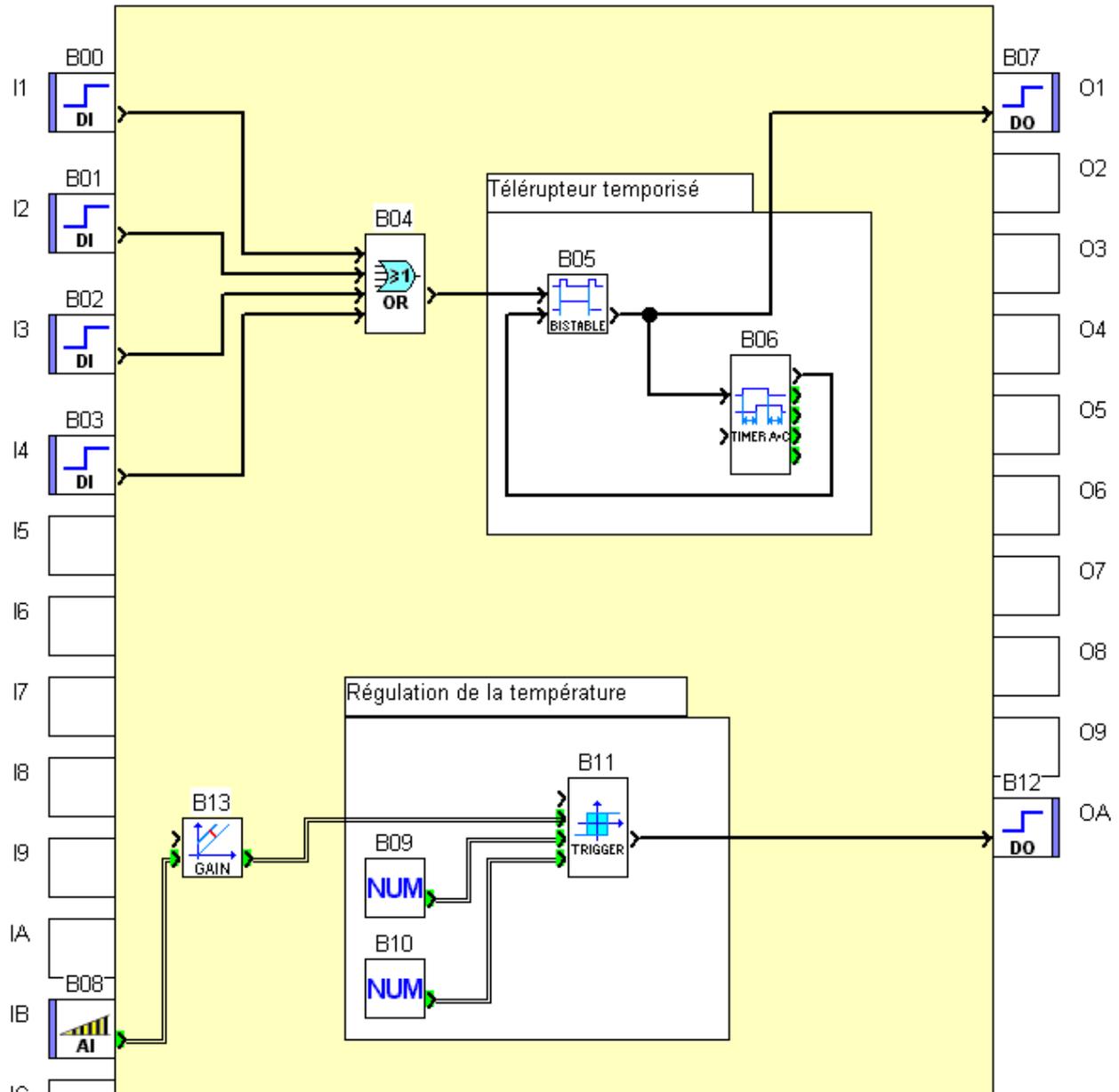


Reste maintenant à ajouter à ceci un objet texte afin de décrire ce que fait cette « fonction ».



Dans ce cas, rien de très compliqué, il suffit donc de mettre un « nom de fonction » pour que ce soit suffisamment explicite.

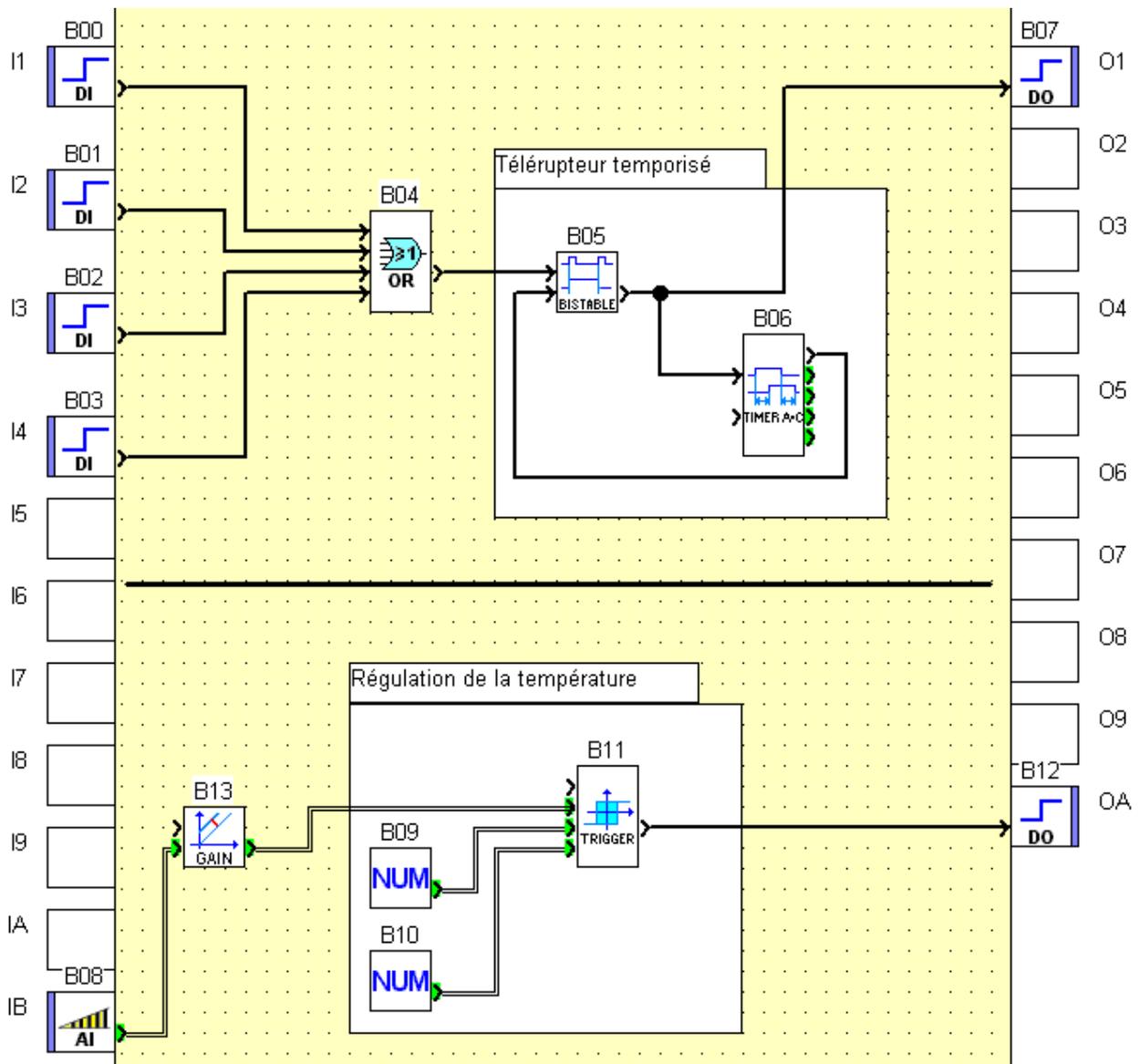
En appliquant le même principe à la partie régulation de la température d'ambiance, on obtient ceci pour l'ensemble du programme :



On a fait très peu de choses mais, avec ça, on sait maintenant ce que font les différentes parties du programme du premier coup d'œil. Et même si vous retournez à ce programme dans 2 ans, vous pourrez facilement et rapidement le reprendre en main.

Utiliser des traits pour séparer des parties de programmes qui n'ont rien à voir

Toujours en prenant l'exemple de la première phase du tutoriel, on va simplement ajouter un trait horizontal, afin de bien prononcer la séparation entre les parties du programme.

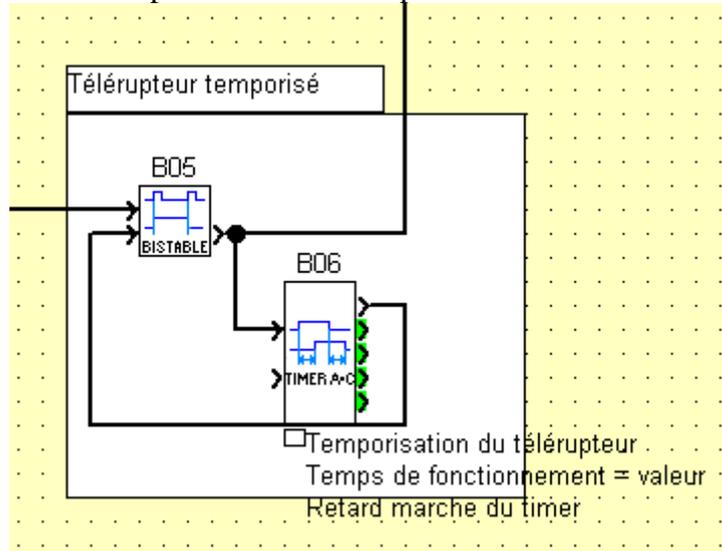


Mettre des commentaires

Mettre des commentaires dans son programme est essentiel, cela permet de donner des indications à soi même ou aux autres (les copains du forum) à comprendre ce que l'on a fait pour une reprise ultérieure du programme.

Sur les blocs fonction eux-mêmes

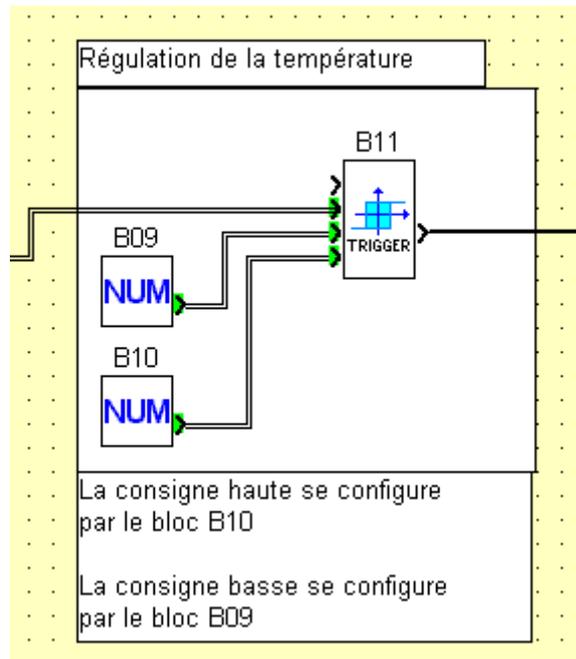
Pour mettre un commentaire sur un bloc, il faut ouvrir sa page de paramétrage et afficher le premier onglet (celui nommé « commentaires »). Voici par exemple un commentaire un peu plus complet ajouté sur la temporisation du télérupteur :



Alors oui ! le problème des commentaires sur les blocs, c'est qu'on ne peut pas les placer où l'on veut, et qu'en plus on ne peut rien placer sur l'espace occupé par le commentaire lorsqu'il est affiché et ce, même si il est masqué. C'est pourquoi personnellement je préfère largement utiliser l'objet texte.

A l'aide des objets texte

En plaçant un objet texte sous la partie régulation de la température, il est possible d'indiquer comment paramétrer la fonction :



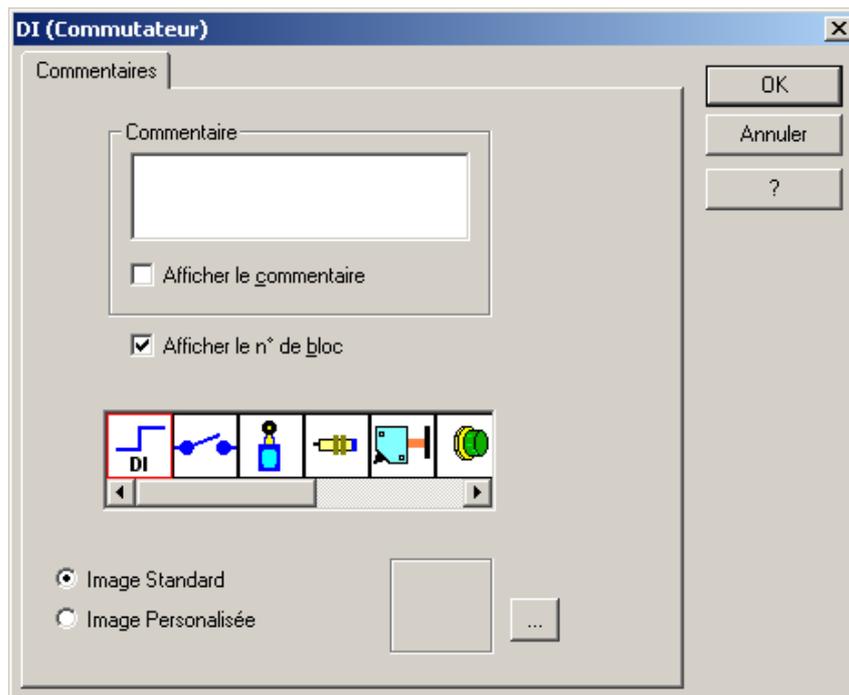
Avec un commentaire comme celui-ci, n'importe quelle personne sera capable de réutiliser ce morceau de programme.

Utiliser des images

Sur les entrée ou sorties TOR

Dans votre installation, vos entrées tout ou rien seront certainement câblées à des interrupteurs ou à des capteurs facilement identifiables. Il peut donc y avoir un intérêt à utiliser une image standard ou personnalisée sur les blocs entrée TOR pour améliorer la correspondance programme/installation.

Lorsqu'on ouvre la fenêtre de paramétrage du bloc DI (entrée TOR) on voit qu'il existe déjà une liste d'images standards disponibles qui représente les « objets » les plus standards que l'on peut utiliser en entrée TOR : contact, capteur à galet, inductif, ou à réflexion directe, bouton poussoir, commutateur ou encore relai.



Rien qu'en utilisant ces images, cela aide visuellement à la compréhension du câblage d'entrée et donc de son action sur le fonctionnement du programme. Par exemple, on n'utilisera pas de la même manière le signal d'un bouton poussoir et l'état d'un commutateur.

Après, vous avez aussi la possibilité de mettre vos propres images (qui dans l'idéal auront une dimension de 36 pixels par 36 pixels) en sélectionnant « Image personnalisée » et en cliquant sur le bouton « ... » pour sélectionner le fichier que vous souhaitez utiliser.

Voici un exemple d'image personnalisée dans une entrée TOR :

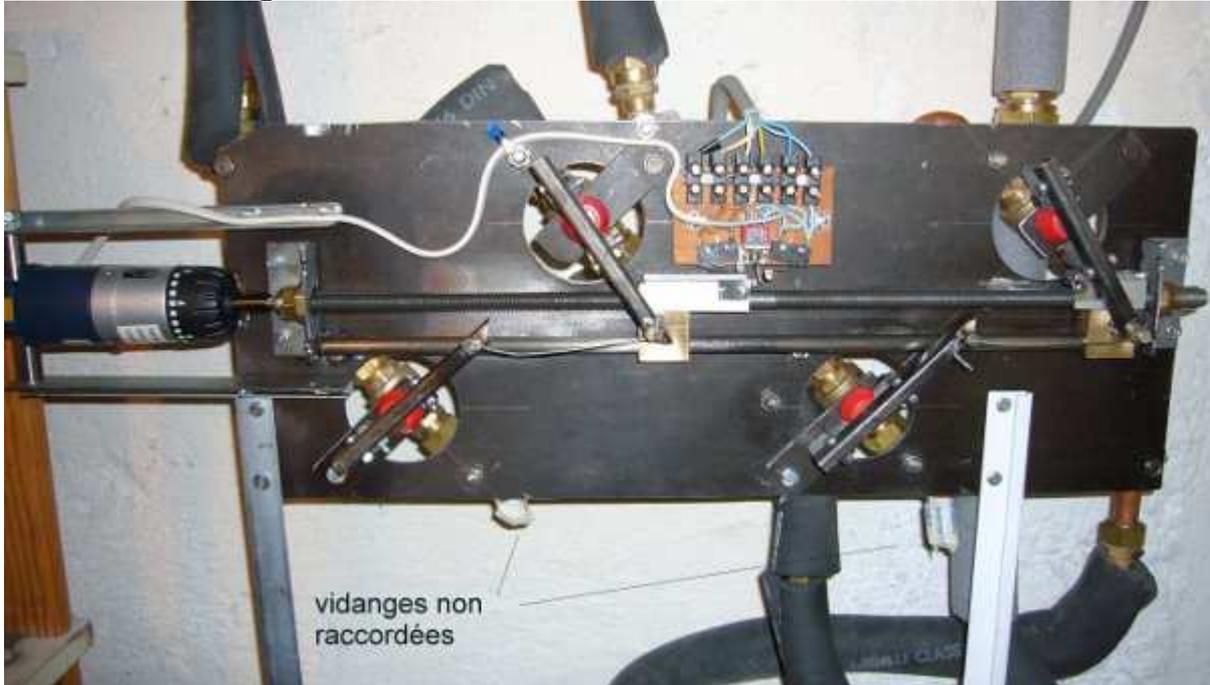


Il est également possible d'utiliser ce principe pour les sorties TOR ainsi que pour les MACROS.

Dans la feuille de câblage

On l'a vu un peu avant, il existe un objet permettant d'insérer des images dans le programme. Ceci n'est pas forcément essentiel mais peut s'avérer très utile si on pilote une partie « home made » qui possède plusieurs capteurs et actionneurs.

Voici pour moi un très bon exemple, cette réalisation de Fred04 ([Veyret Frederic](#)) Dans cette installation nous avons un moteur de commande qui peut être piloté dans les deux sens, et deux capteurs de fin de course. C'est « simple », mais parfait pour introduire l'intérêt de l'utilisation des images dans l'atelier M3.



Dans les faits, ce montage n'a pas besoin d'un M3, mais il est parfaitement possible d'utiliser le M3 pour piloter le système.

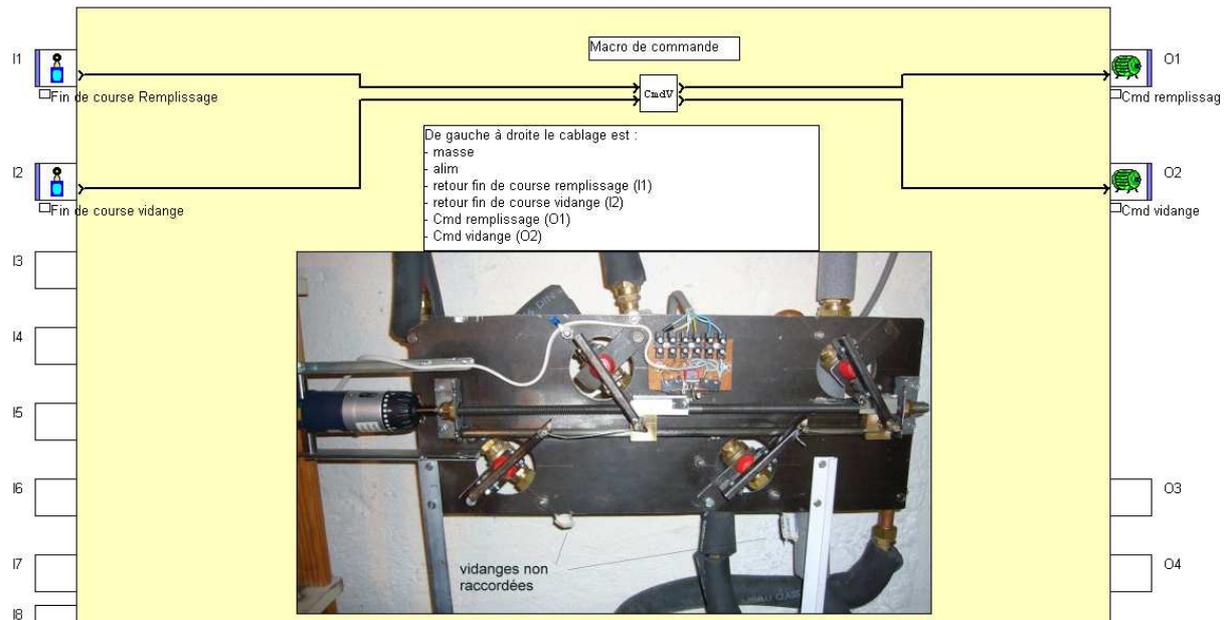
Alors je n'ai pas réalisé le programme de pilotage (ce n'est pas le but de cette phase du tutoriel), mais je montre simplement une façon « utile » d'intégrer cette image au programme :

Dans le programme j'ai placé, deux entrées correspondant aux capteurs, et deux sorties correspondant au pilotage dans les deux sens.

La macro au milieu est le morceau de programme qui réalise le pilotage du système.

L'image associée au bloc texte juste au dessus permet de repérer physiquement les connexions du système.

NOTE : le placement des connexions que j'ai décrit est complètement arbitraire et est présent uniquement pour illustrer la façon dont on peut associer image et texte dans un programme M3 afin de faciliter la compréhension d'une partie de programme.



Dans ce cas, c'est vrai que l'image est facultative, mais dans le cas d'une machine avec plus d'entrée sorties, avoir une sorte de schéma de câblage ou fonctionnel sous les yeux peut apporter un confort pour trouver la bonne logique de programmation.

Mettre des liens texte

J'en ai parlé dans la première phase du tutoriel, mettre des liens texte est vraiment un moyen de multiplier par 1000 la lisibilité d'un programme. Je ne vais pas réexpliquer ici comment on fait mais, par contre, je vais illustrer l'intérêt dans le chapitre suivant.

Un exemple sur un programme des membres APPER :

Pour présenter une bonne façon de travailler, j'ai volontairement pris un programme bien pensé mais moyennement bien présenté, c'est-à-dire un programme avec très peu de commentaires, pas de regroupage logique des blocs, que des liens filaires ou presque... bref, un programme qui ne peut être compris rapidement que par la personne qui l'a créé. Notons tout de même que l'exemple que j'ai pris est loin d'être le plus mauvais niveau présentation qu'il m'ait été donné de voir puisqu'il comporte le minimum nécessaire pour la compréhension, c'est-à-dire à quoi correspondent les entrées, et à quoi correspondent les sorties, avec en prime quelques liens textes.

La première étape pour y voir plus clair : les liens textes

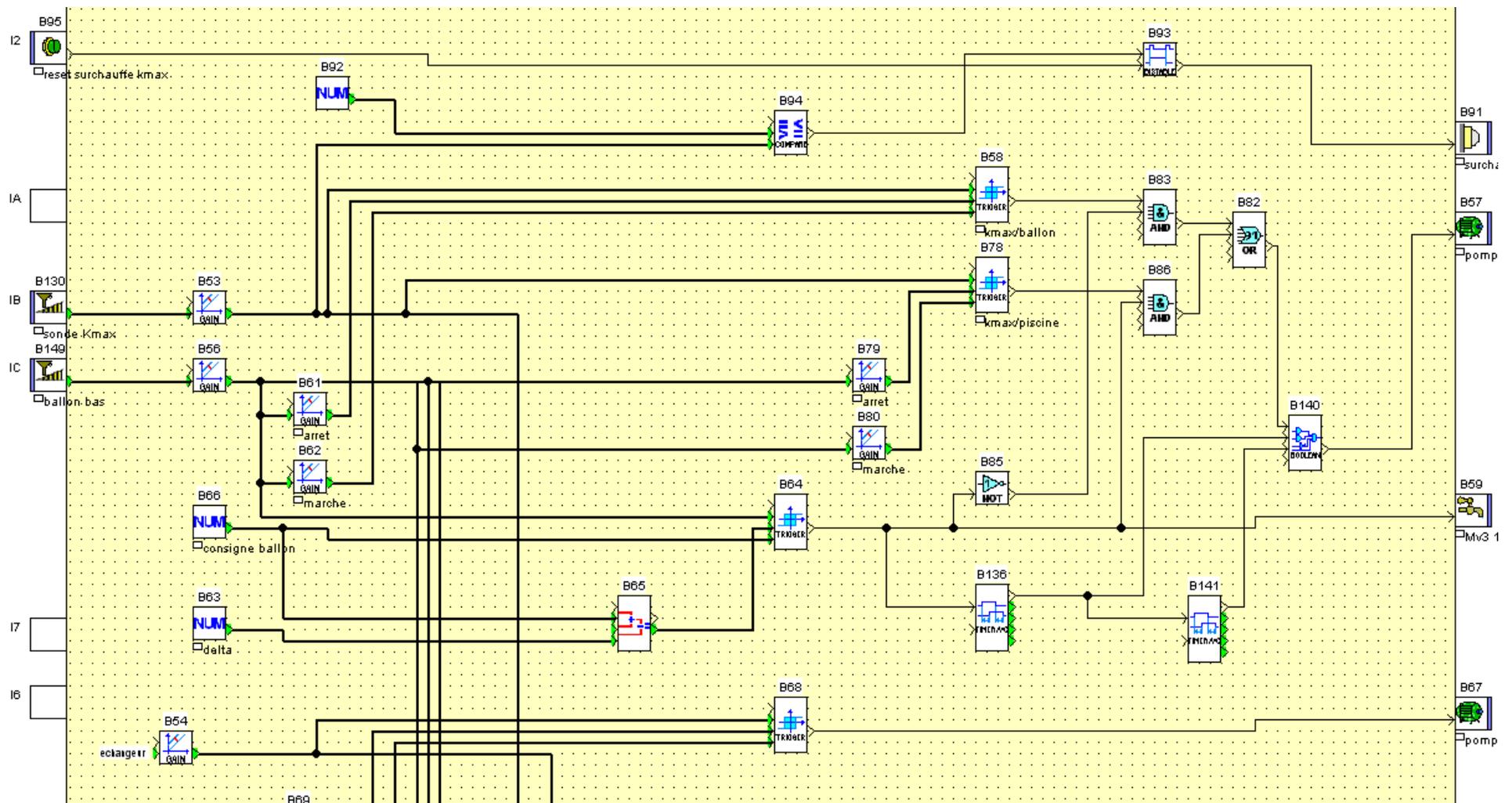
Les liens textes ne se placent pas au hasard. Le premier cas où on mettra toujours un lien texte c'est le cas où la sortie d'un bloc est connectée sur plusieurs entrées d'autres blocs, formant ainsi un réseau tentaculaire. Le deuxième cas est celui des liens qui traversent la feuille de programmation de part en part.

Une fonctionnalité de l'atelier, dont je n'ai pas parlé mais qui s'avère très utile lorsqu'on attaque l'organisation des programmes, est l'affichage des dépendances.

Dans la barre d'outils principale, on trouve un petit bouton discret mais qui est une fonction magique :

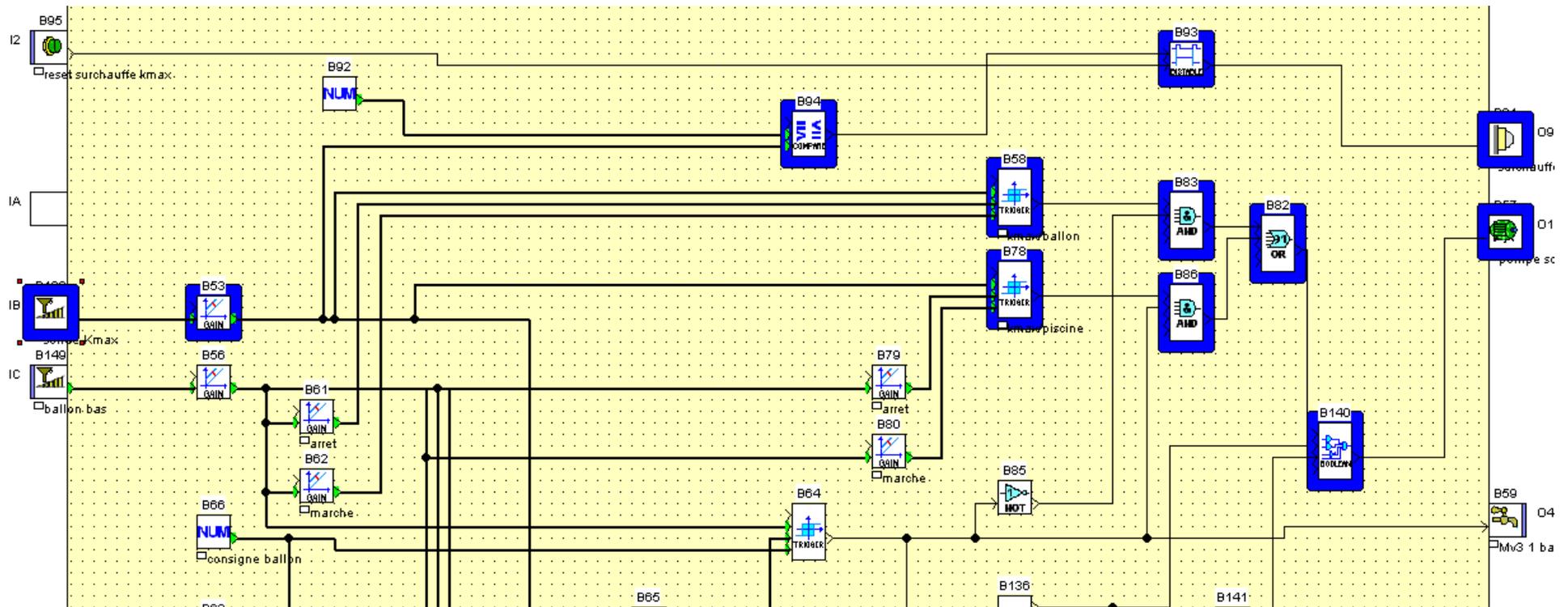


Prenons ce programme :



On le remarque facilement, certain liens sont tentaculaires mais à l'aide de l'affichage des dépendances, on va pouvoir identifier très vite à quoi tout ceci est connecté :

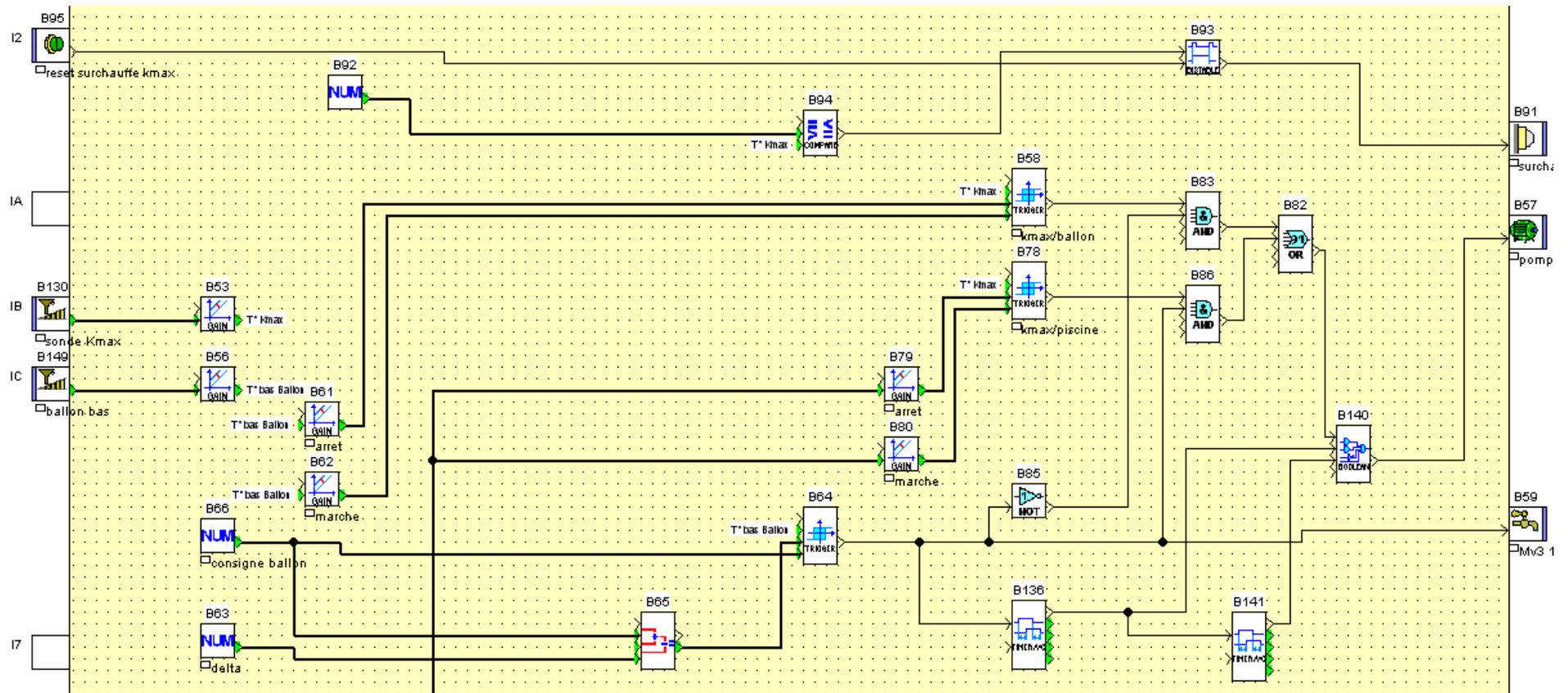
Il suffit de sélectionner la première entrée analogique (sonde Kmax sur l'image) en ayant préalablement activé l'affichage des dépendances (cliquez sur le petit bouton que je vous ai montré avant) :



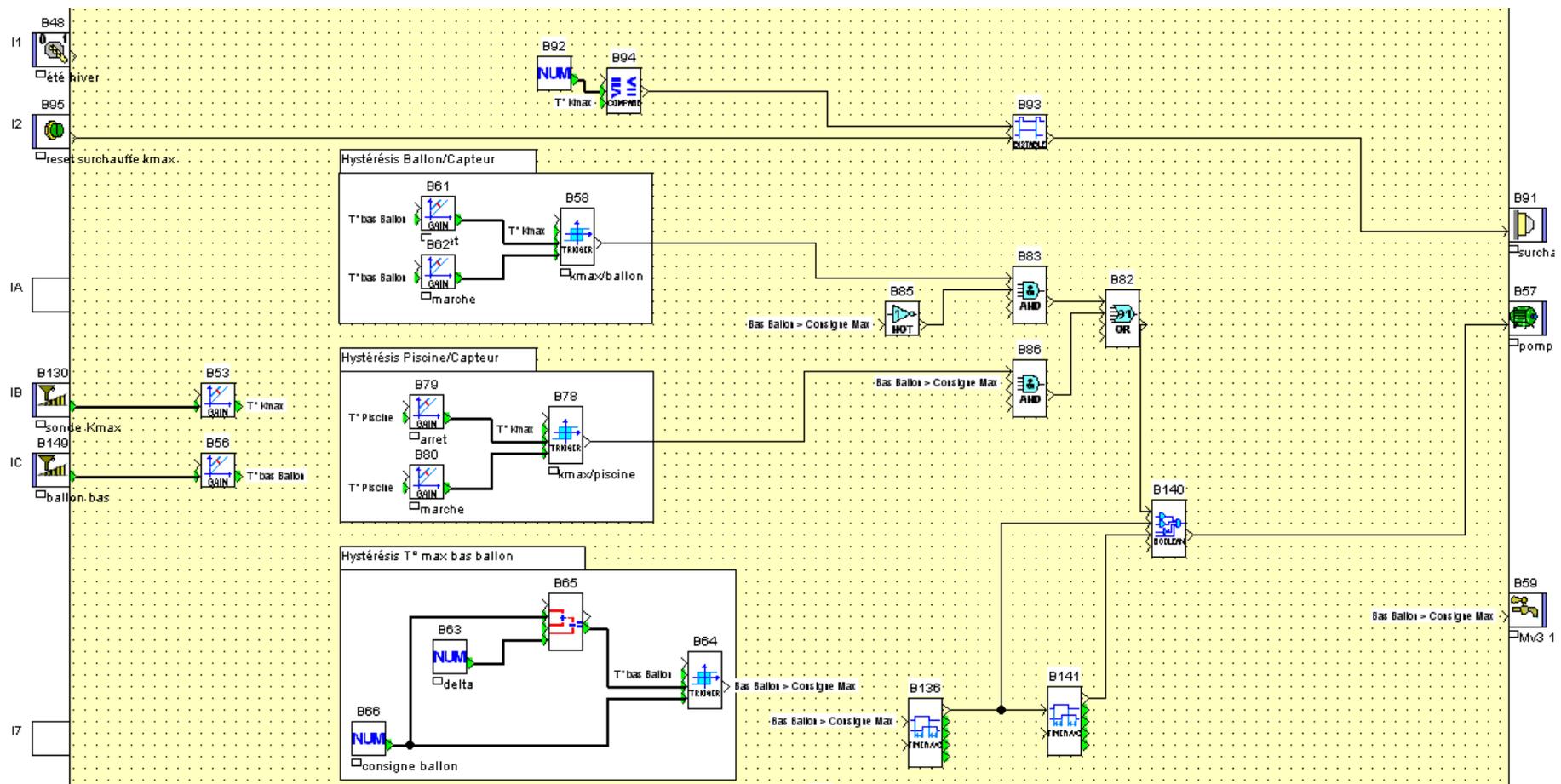
Si ce n'est pas une révélation ca !!!, on voit du premier coup d'œil toute la chaîne de blocs qui dépend de cette entrée.

ATTENTION : l'affichage des dépendances surligne tout les blocs qui sont connectés directement ou indirectement à celui qui est sélectionné, il est donc normal que la chaîne qui apparait ici aille jusqu'aux blocs de sortie. Note : Il est également possible de cliquer sur un lien en particulier plutôt que sur un bloc pour afficher les dépendances (on peut donc visualiser ce qui dépend d'une sortie d'un bloc en particulier)

Alors tout de suite, avec mon habitude, je repère que la sortie du bloc GAIN B53 crée plein de « nœud ». Je vais donc m'empresse de changer de lien filaire en lien texte, et je vais appliquer la même chose à la sortie du bloc B56 juste en dessous.



Voici le résultat du simple passage de deux liens en mode texte (avec un nom explicite). Cela offre déjà un gain de lisibilité conséquent et on se rend mieux compte des regroupements que l'on peut faire. Cette fois-ci, je fais un « saut dans le temps » et après quelques regroupements/liens textes/placements de rectangle pour regrouper des fonctions, on arrive à cela pour cette partie du programme :

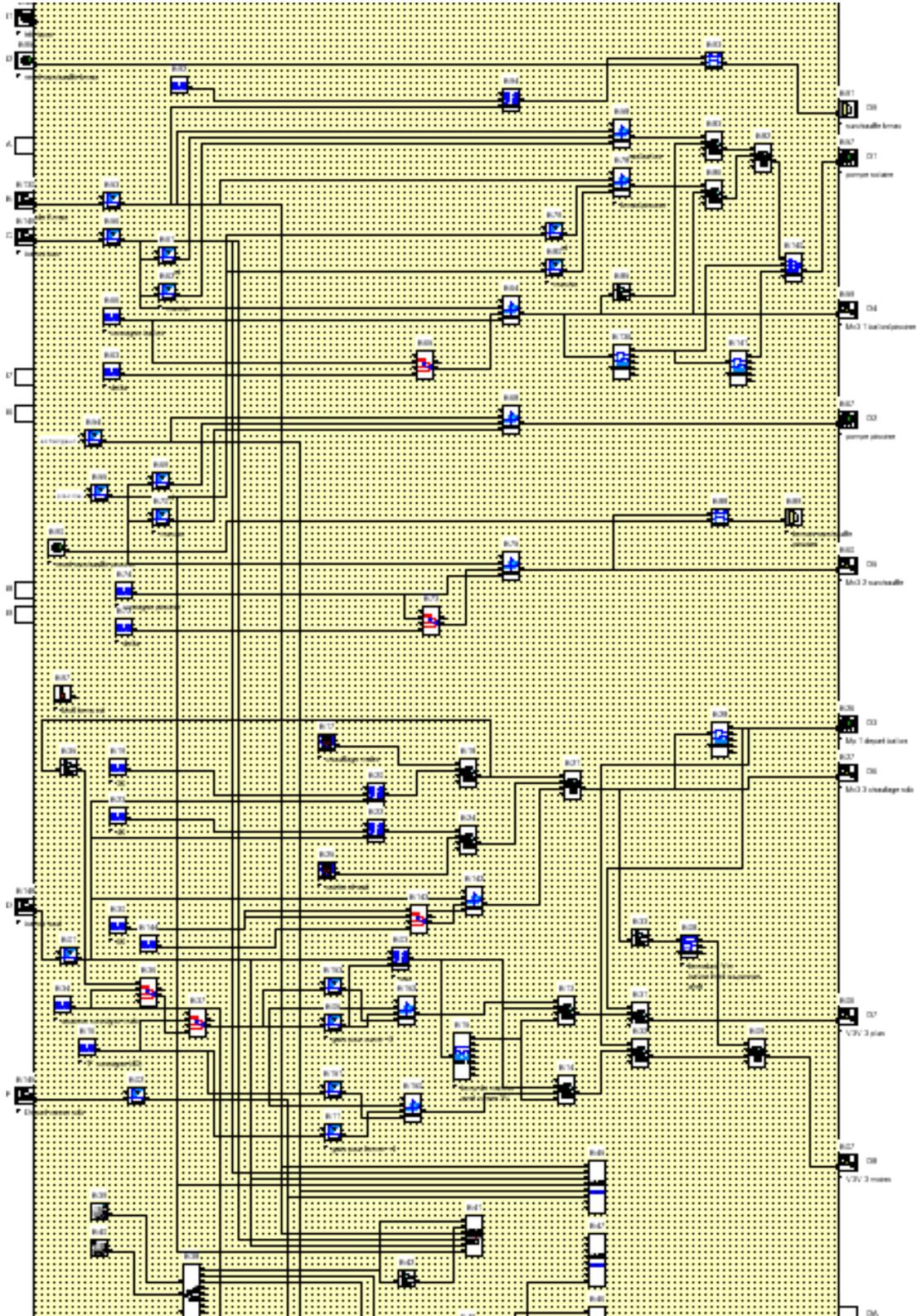


Alors, c'est certes très subjectif, mais de mon point de vue, un programme présenté comme ça, il faut moins de 2 minutes pour en comprendre les principales fonctions alors qu'avant cela aurait probablement pris beaucoup de temps.

Comme le principe de l'organisation d'un programme est toujours le même, je me permets de vous exposer en dernier l'apparence du programme au départ, et l'apparence du programme une fois réorganisé. Notons que dans les 2 cas le zoom est à 25%, et que sur le programme brut il n'est pas possible de l'afficher sur une seule page. Je précise que le programme que j'ai pris pour exemple est un programme qui n'est pas terminé et qu'en conséquence on ne peut pas considérer qu'il aura cet aspect brut dans sa version finale (on fait souvent l'organisation à la fin, pour éviter d'être embêté avec les objets graphiques pendant le câblage).

Programme réorganisé

Programme brut



Conclusion

Je n'ai nullement la prétention d'avoir la meilleur des solutions pour organiser un programme, néanmoins retenons les bonnes pratiques comme les liens textes et les regroupements.

Le programme « réorganisé » ci-dessus sera de retour dans la toute dernière phase du tutoriel car il est un très bon exemple de mise en place des MACROs. En effet, il comporte de « nombreuses » répétitions de mêmes groupes de blocs (hystérésis avec deux blocs gain en 5 exemplaires, et hystérésis avec addition en 3 exemplaires). C'est en utilisant les MACROs sur ces « groupes » que l'on comprendra la réelle puissance de cet outil et ce, tant sur le plan fonctionnel qu'organisationnel.