

# INITIATION AU LOGICIEL MILLENIUM 3

## API CROUZET

### 1 Introduction

Le logiciel Millennium 3 sert à programmer l'Automate Programmable Industriel (API) Crouzet.



On nomme **Automate Programmable Industriel**, (**API**, en anglais **Programmable Logic Controller**, **PLC**) un dispositif similaire à un ordinateur, utilisé pour automatiser des processus comme la commande des machines sur une chaîne de montage dans une usine.

Là où les systèmes automatisés plus anciens emploieraient des centaines ou des milliers de relais et de cames, un simple automate suffit. On nomme automaticiens les programmeurs de ces Automates Programmables Industriels.

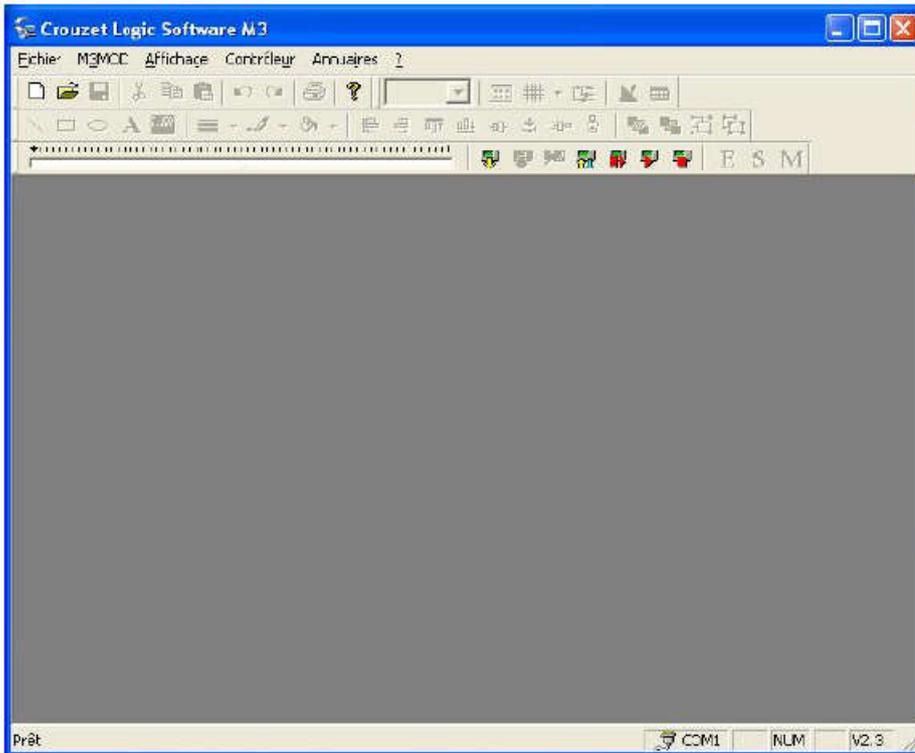
L'API est structuré autour d'une unité de calcul ou processeur (en anglais Central Processing Unit, CPU), d'une alimentation (depuis des tensions alternatives - AC ou continues - DC) et, de modules suivant les besoins de l'application, tel que:

- des cartes d'entrées - sorties (en anglais Input - Output, **I/O**) numériques (Tout ou rien)
- ou analogiques
- des cartes d'entrées pour brancher des capteurs (un capteur est un dispositif qui transforme l'état d'une grandeur physique observée en une grandeur utilisable,...), boutons poussoirs, ...
- des cartes de sorties pour brancher des actionneurs, voyants, vannes, ...
- des modules de communication pour dialoguer avec d'autres automates, des entrées/sorties déportées, des supervisions ou autres interfaces homme-machine
- des modules dédiés métiers, tels que de comptage rapide, de pesage...
- des modules d'interface pour la commande de mouvement, dits modules Motion, tels que démarreurs progressifs, variateur de vitesse, commande d'axes.

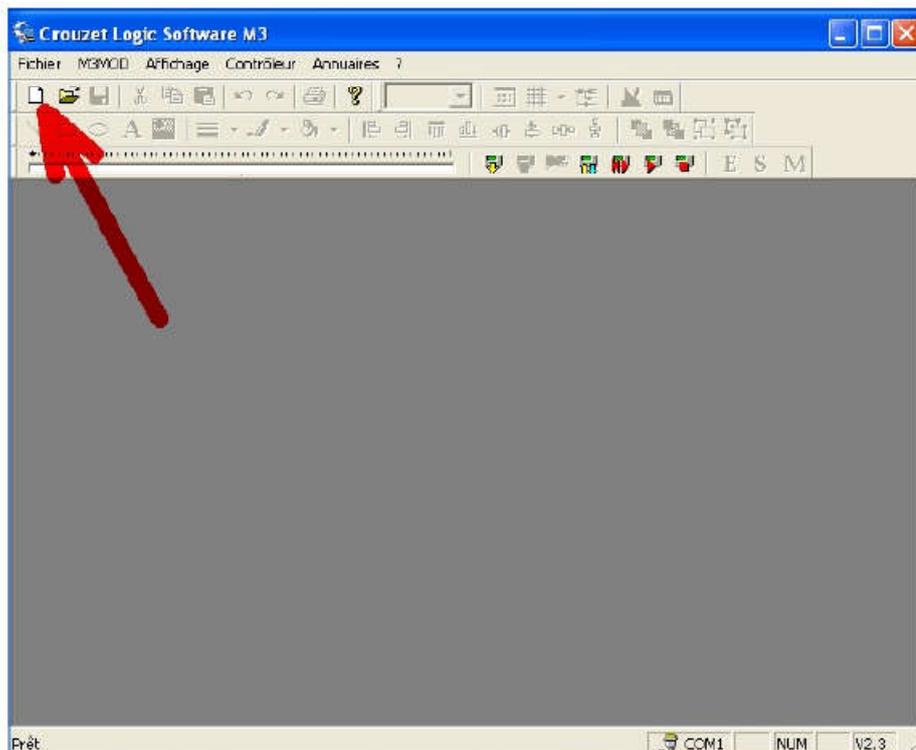
Nous allons étudier le fonctionnement du logiciel de programmation de cet automate. Une fois le programme élaboré et testé sur l'ordinateur, il pourra être transféré dans l'automate pour y être exploité.

## 2 CREATION D'UN NOUVEAU DOCUMENT DE TRAVAIL

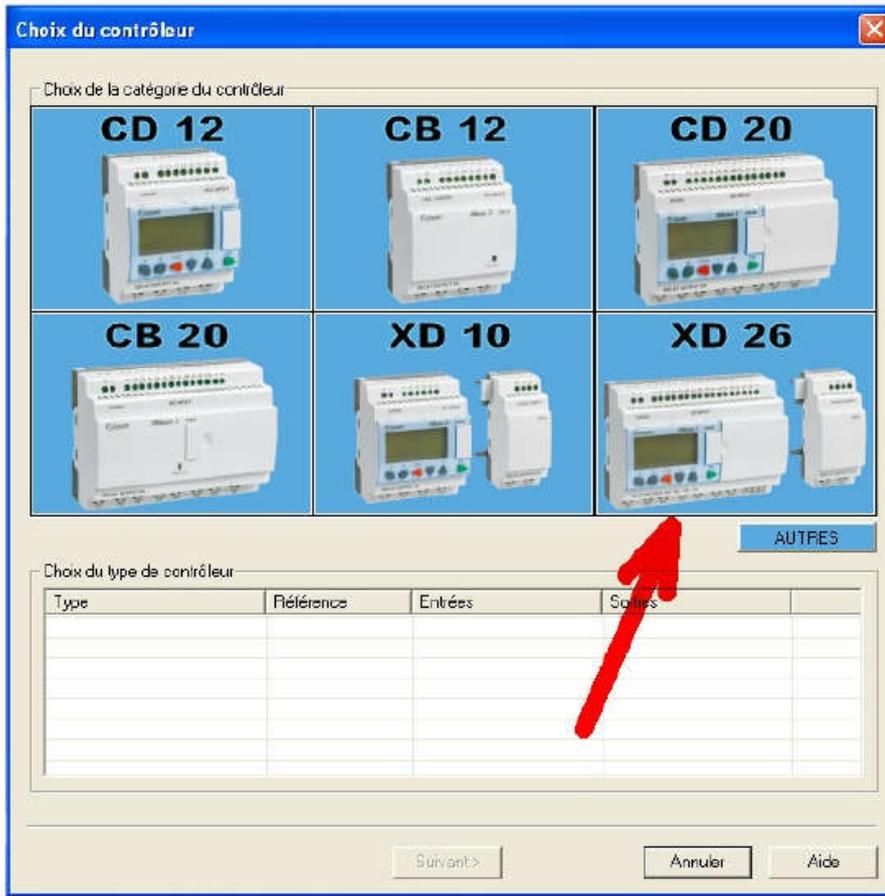
- a. Lancez le logiciel "Millenium 3 AC".



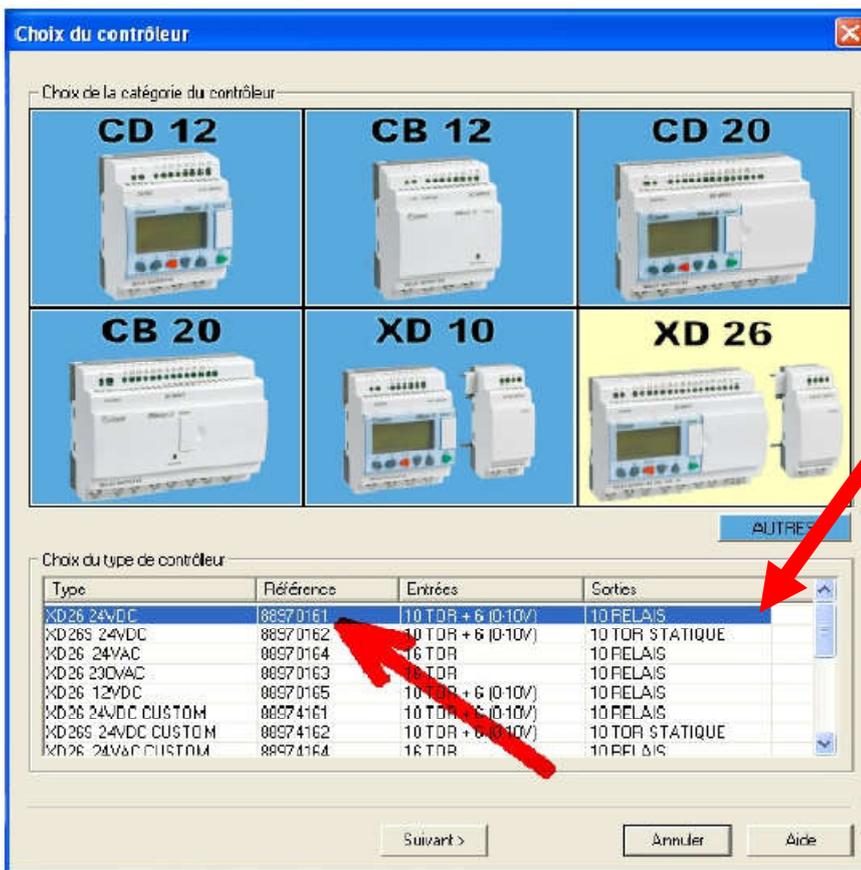
- b. Cliquez sur "Nouveau document".



c. Vous devez choisir le type d'Automate : Exemple : cliquez sur XD26.



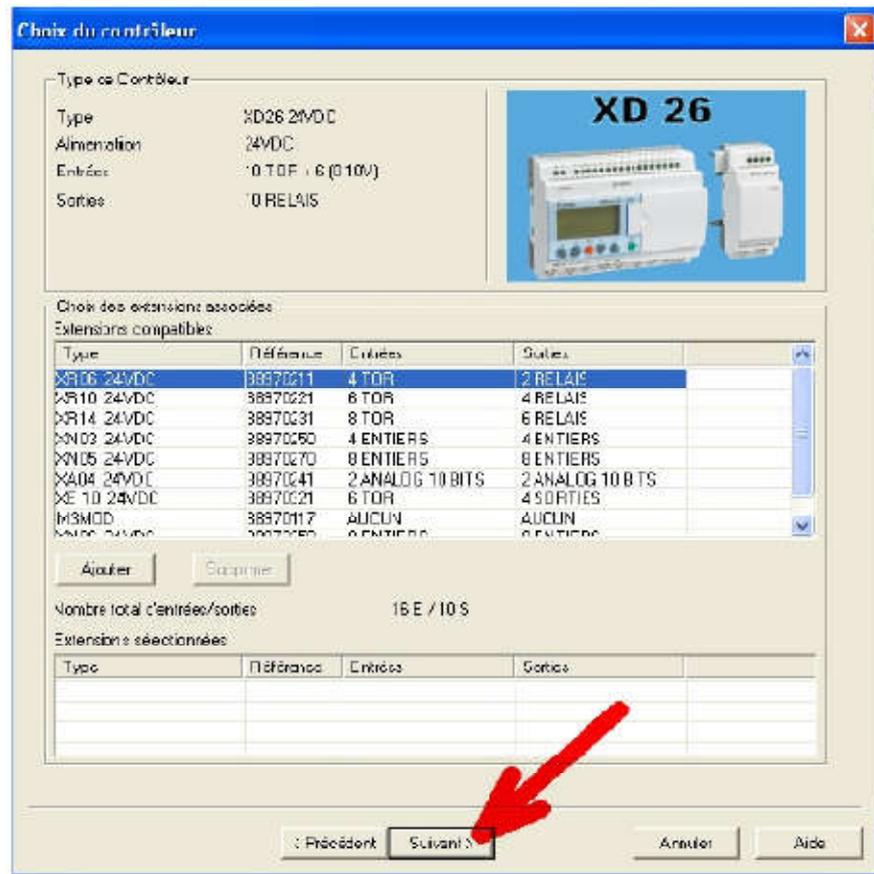
d. Précisez la référence exacte de l'Automate : Exemple : choisissez "88 970 161".



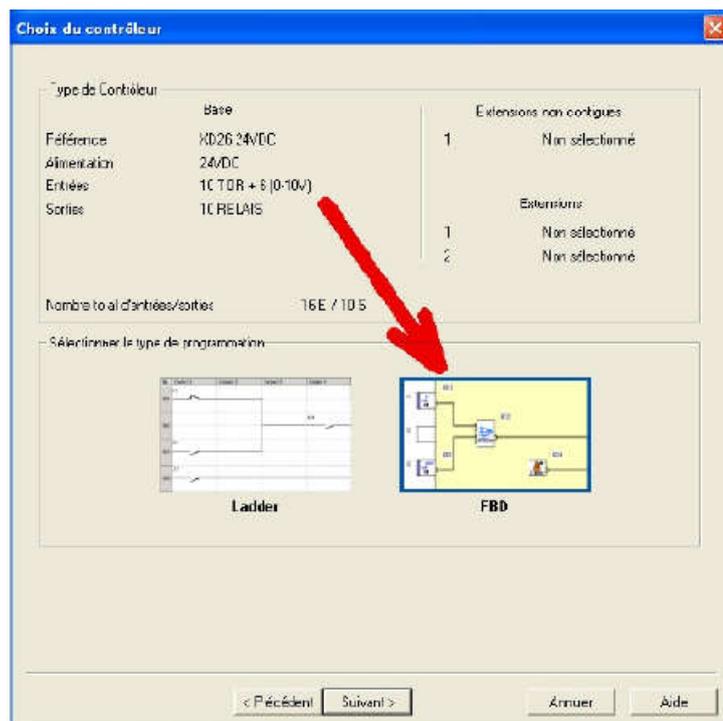
Ce type est caractérisé par :

- 10 entrées Tout Ou Rien (TOR ) et 6 entrées analogiques,
- 10 sorties Tout Ou Rien (TOR ).

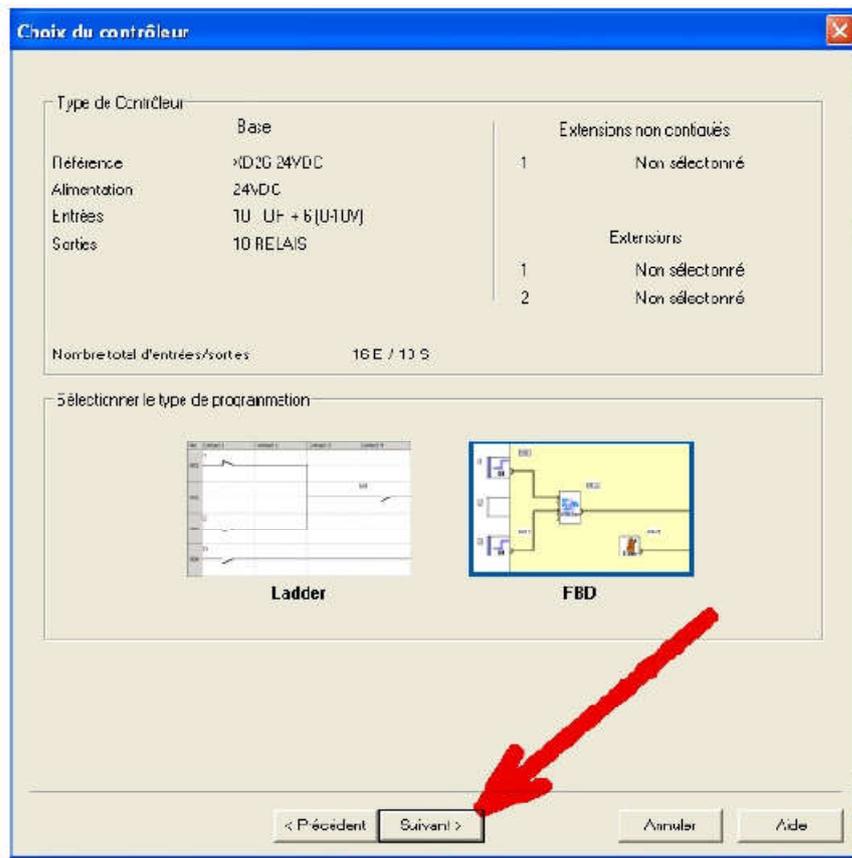
- e. Il est possible d'ajouter des extensions sur l'Automatique (communication, entrées, sorties, etc.). Pour cette application, nul besoin d'ajouter d'extension : cliquez sur **"Suivant"**.



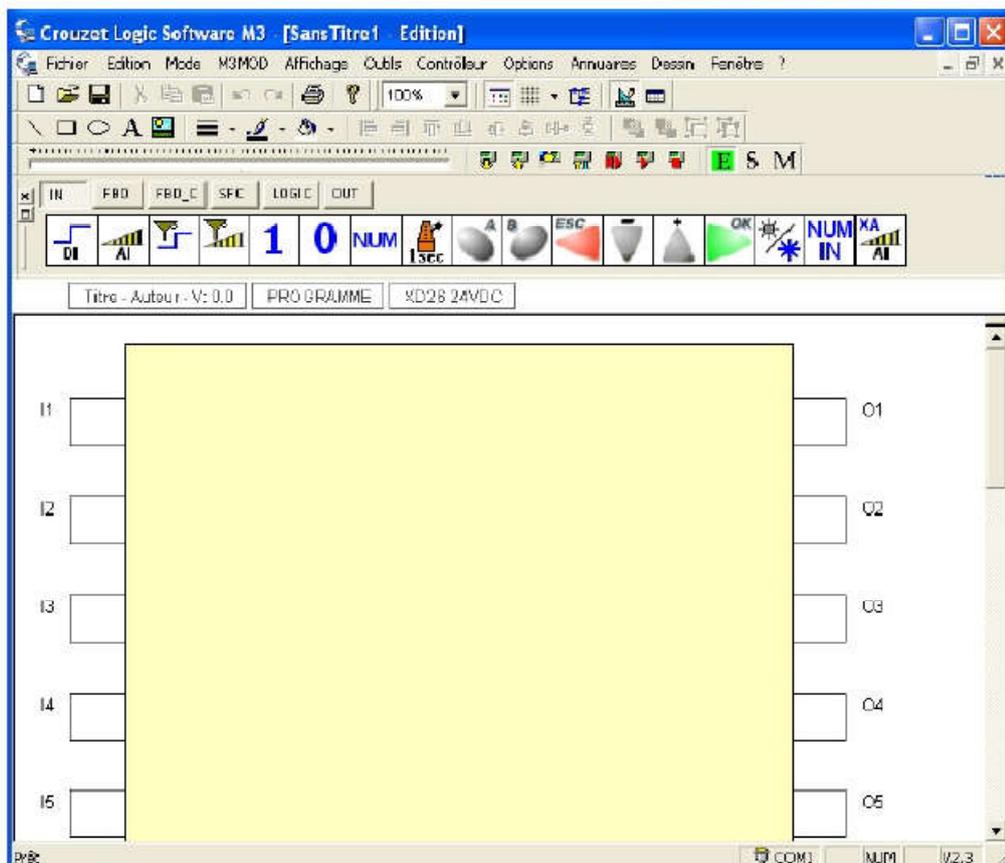
- f. Sélectionnez le type de langage de programmation - langage à contact (Ladder) ou FBD (Functional Block Diagram). Pour cette initiation, cliquez sur **"FBD"**.



g. Validez ce choix en cliquant sur "Suivant".

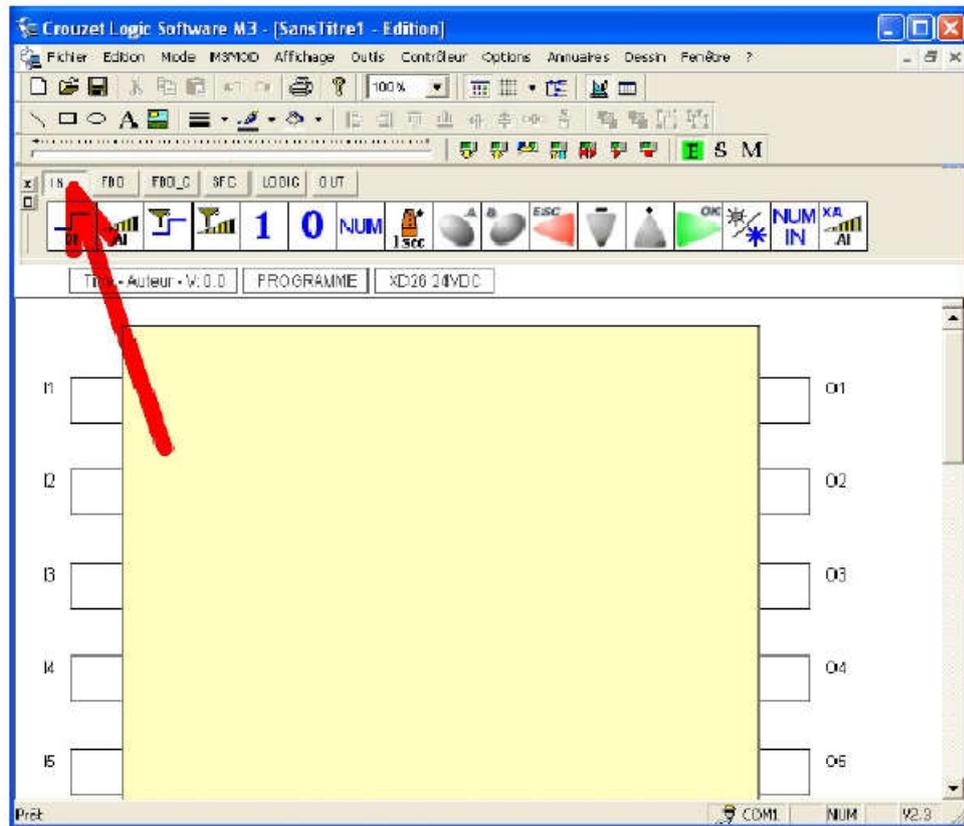


h. Voici le nouveau document configuré pour votre automate.

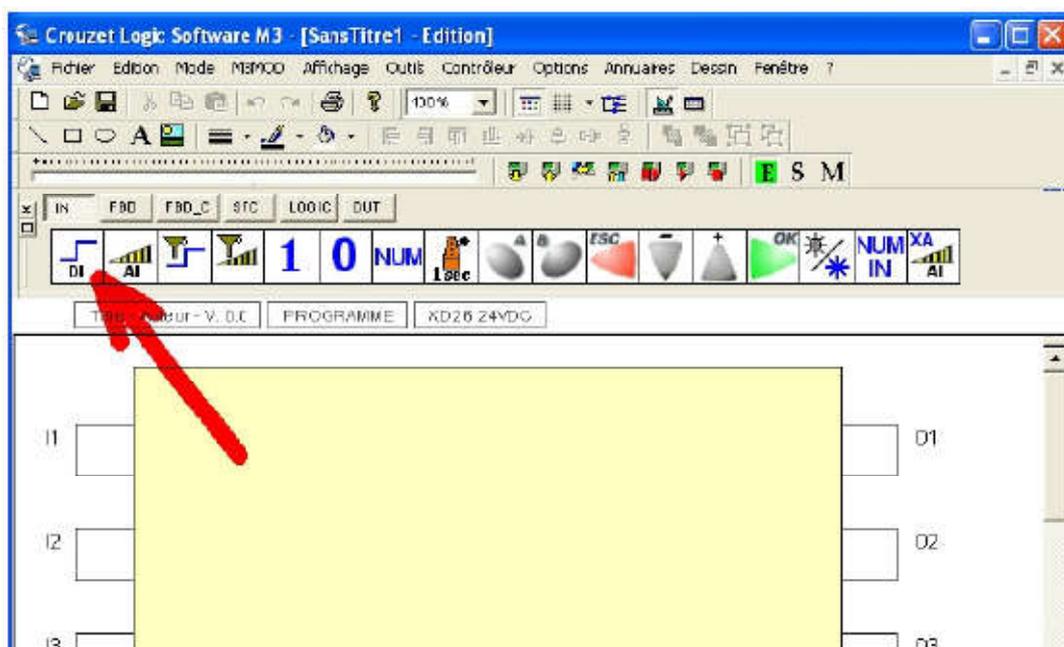


### 3 PROGRAMMATION DE BASE

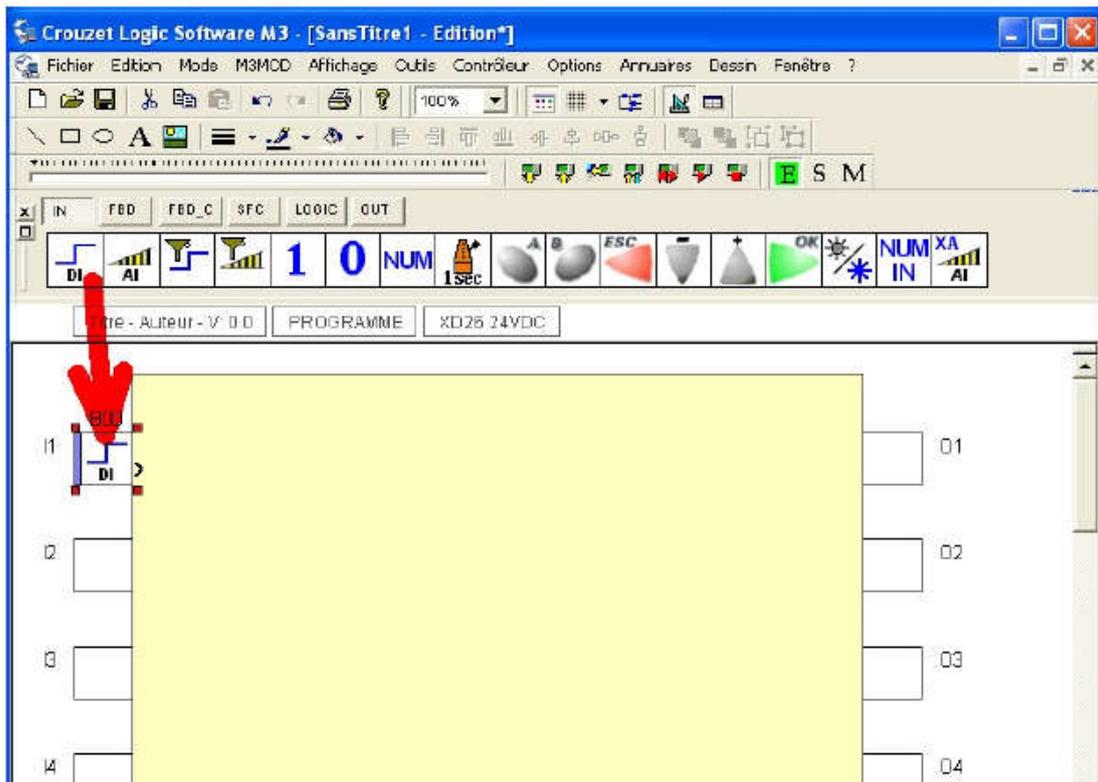
- a. Une barre d'outils nommée "Barre de fonctions" permet d'insérer dans le document des éléments d'entrées (IN), de fonctions (FBD), de grafcet (SFC), de logique (LOGIC) ou de sorties (OUT). Choisissez pour commencer le menu de fonctions "IN".



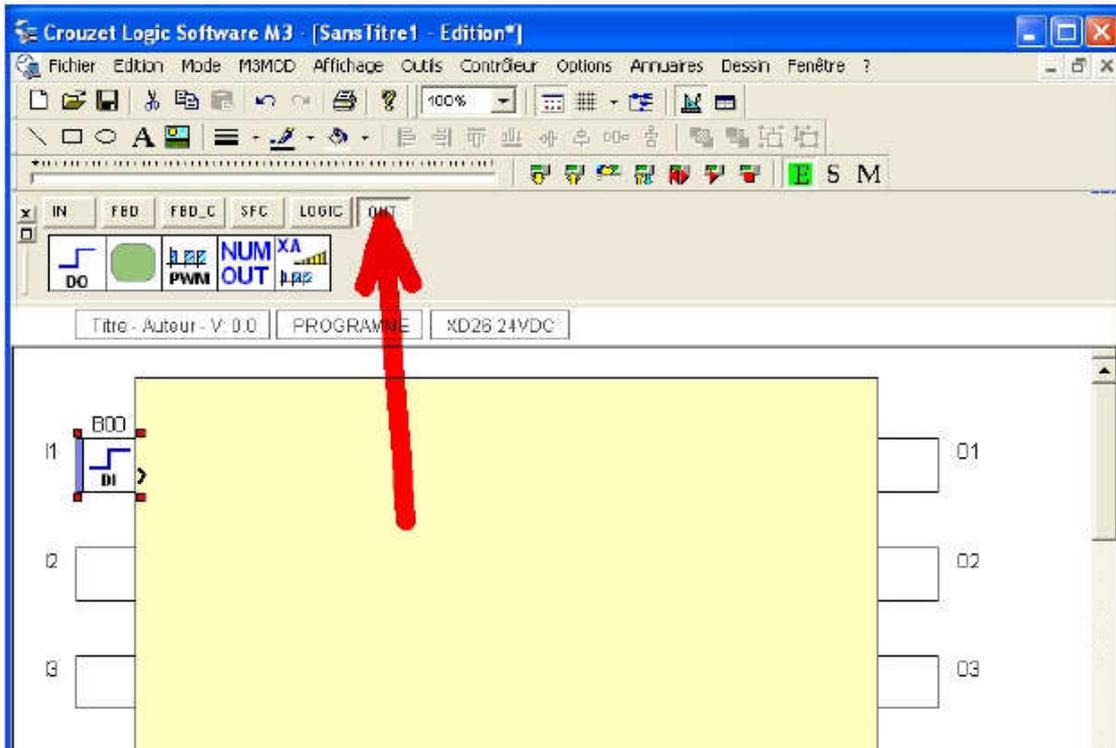
- b. Nous allons placer une entrée binaire (de type Tout Ou Rien - TOR) sur notre document : elle se nomme "DI" (Digital Input).



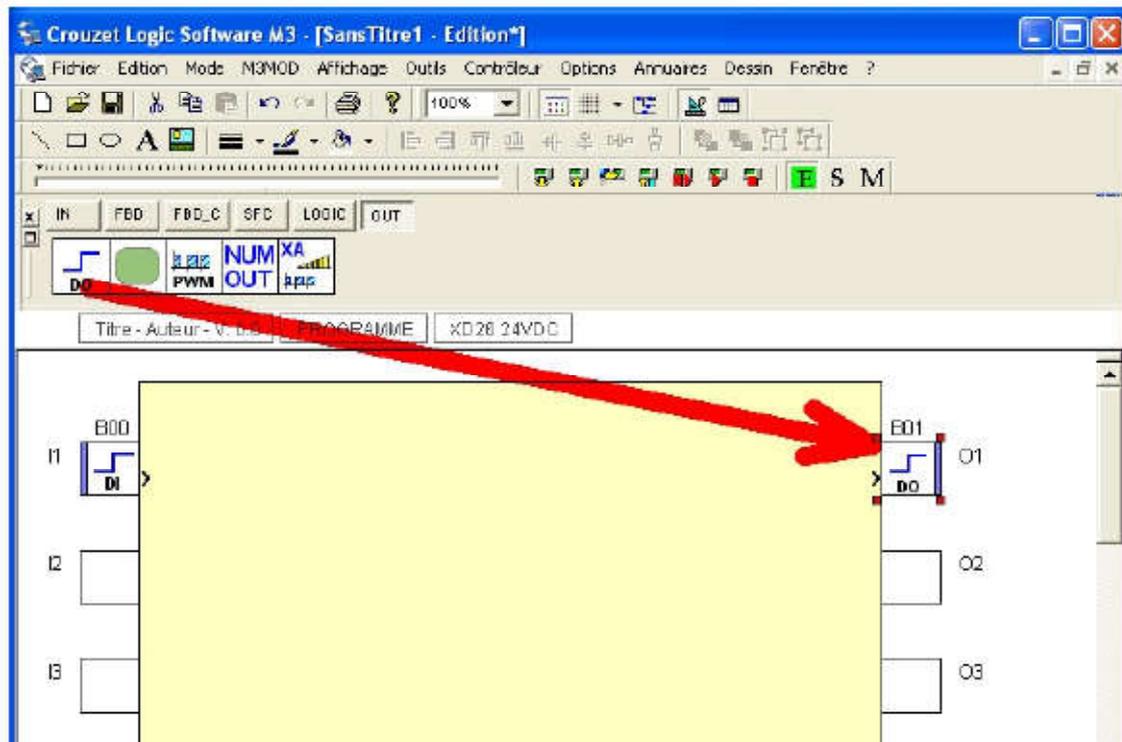
- c. Faites glisser l'icône sur une des bornes d'entrées de l'automate (I1 à IA).



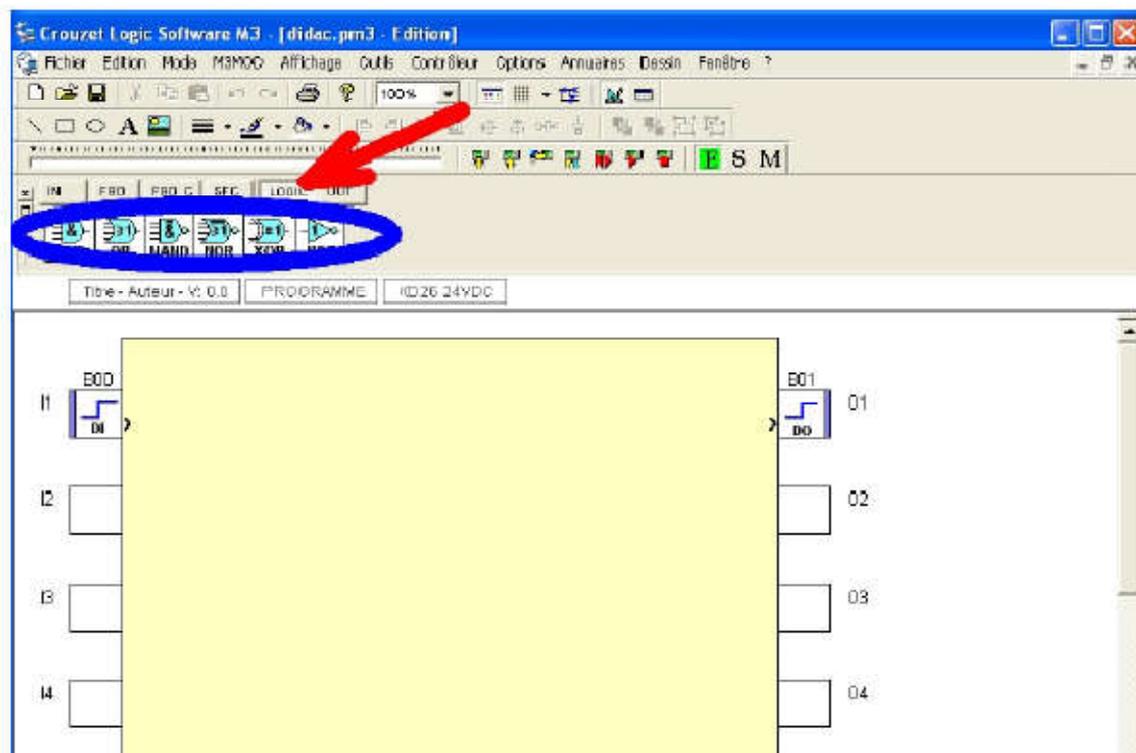
- d. Choisissez le menu de fonctions "OUT" (sorties).



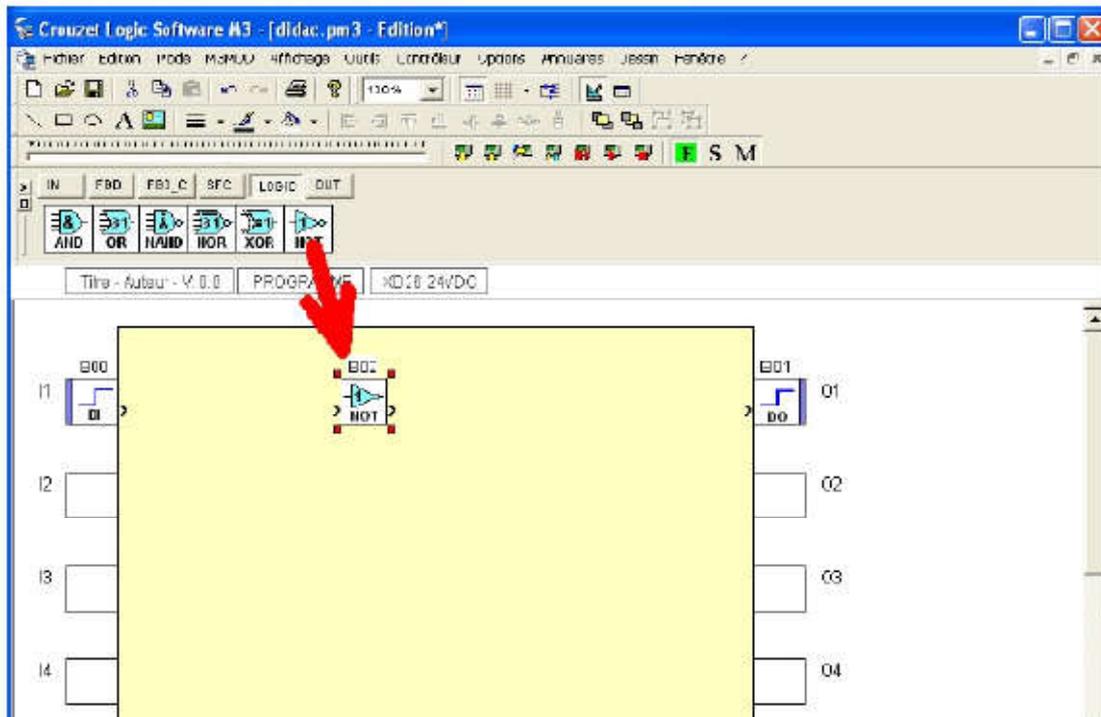
- e. Faites glisser l'icône de sortie "DO" (Digital Output) sur une des bornes de sortie de l'automate (O1 à OA).



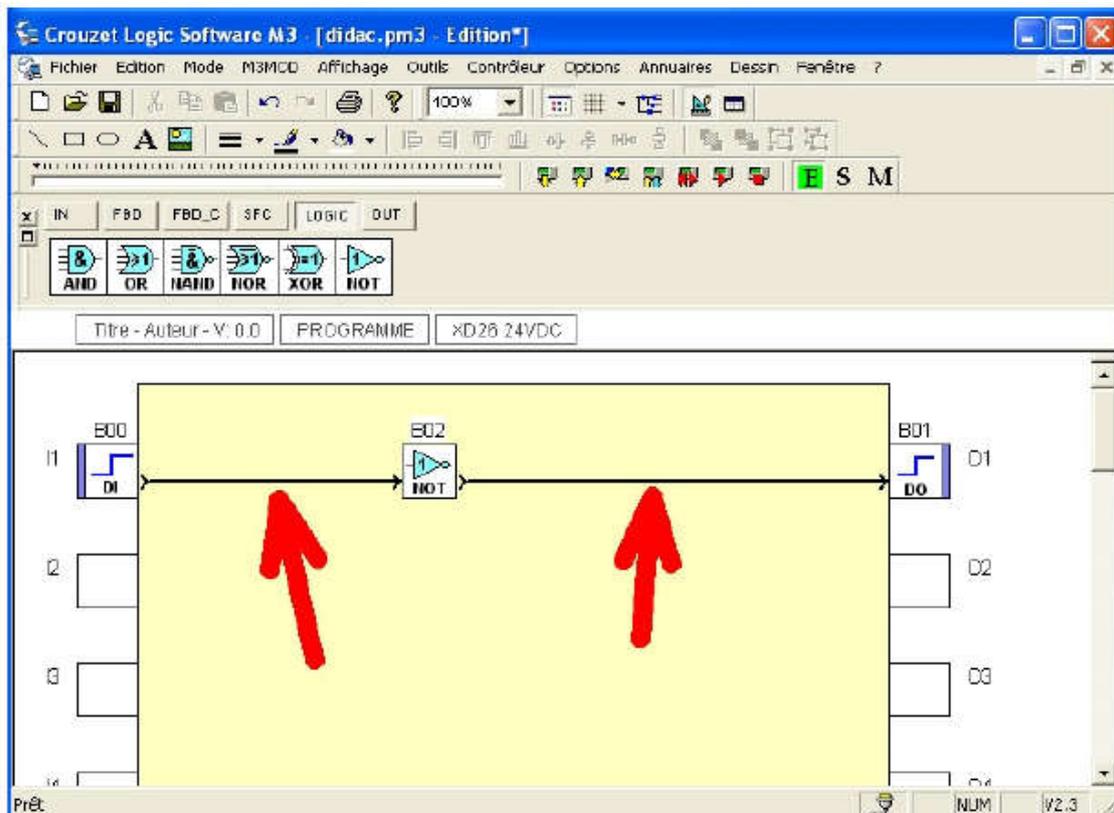
- f. Choisissez le menu de fonctions "LOGIC".



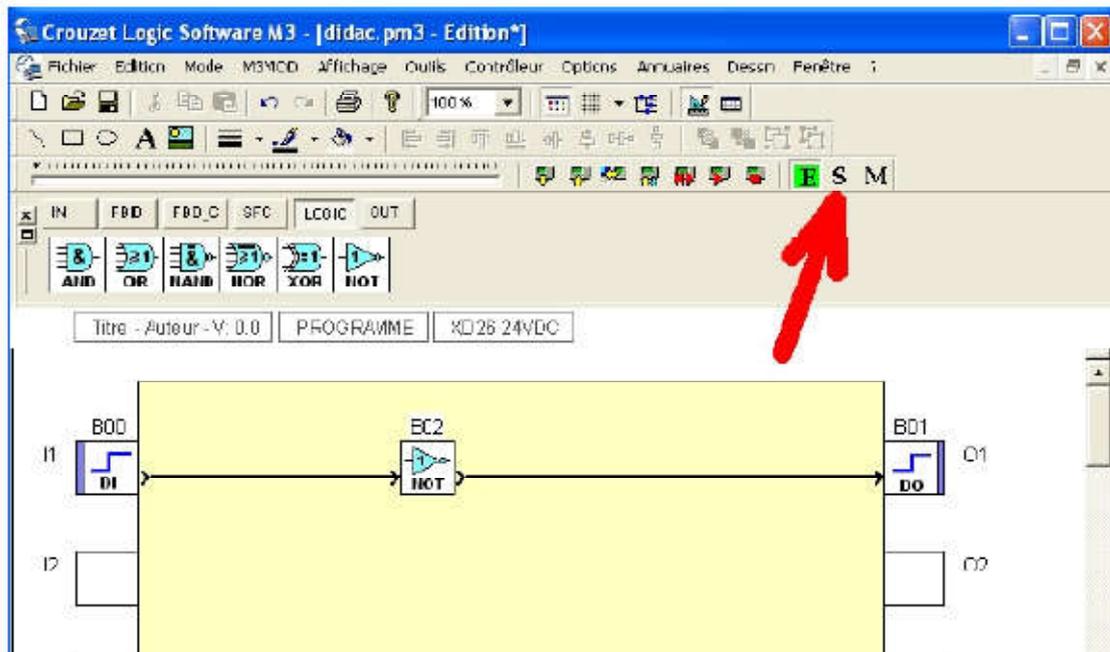
- g. Faites glisser l'icône de l'inverseur logique (NOT) sur l'espace de travail.



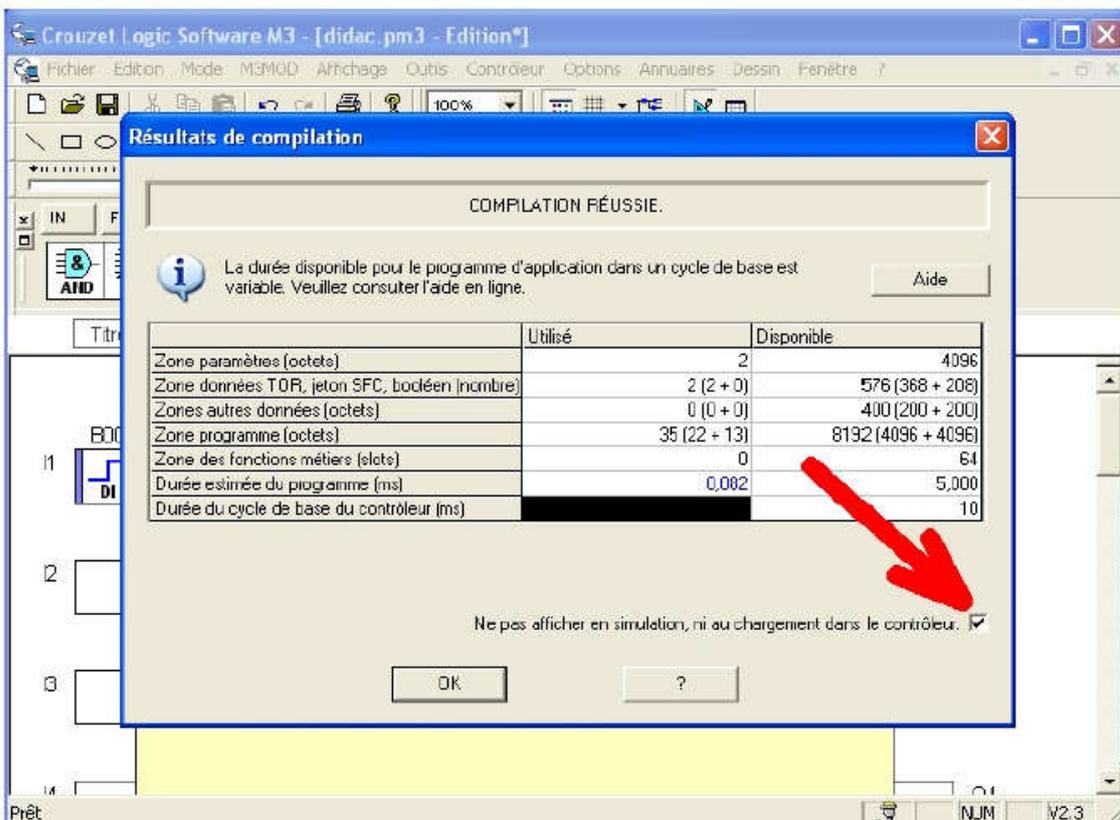
- h. Effectuez les liaisons en cliquant-déplaçant votre souris depuis les bornes d'entrée/sortie vers les connexions de l'inverseur logique.



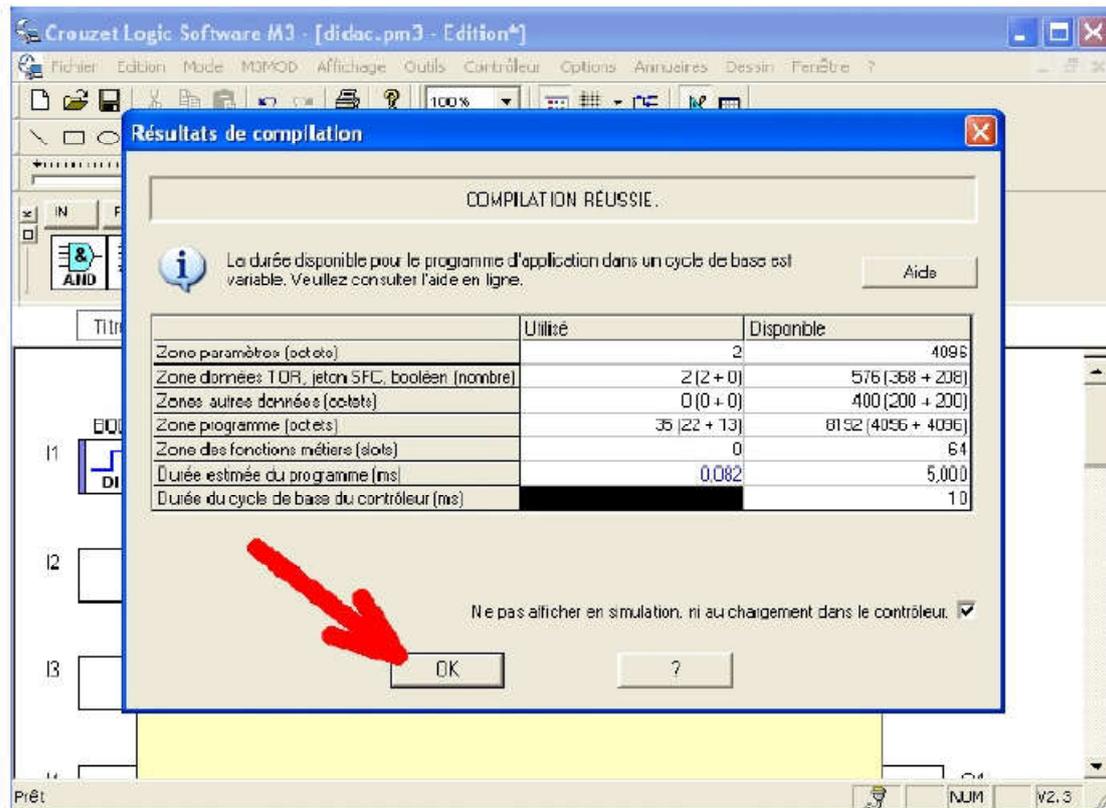
- i. Votre circuit de base est terminé.. Il faut maintenant vérifier son fonctionnement à l'aide du simulateur (bouton "S").



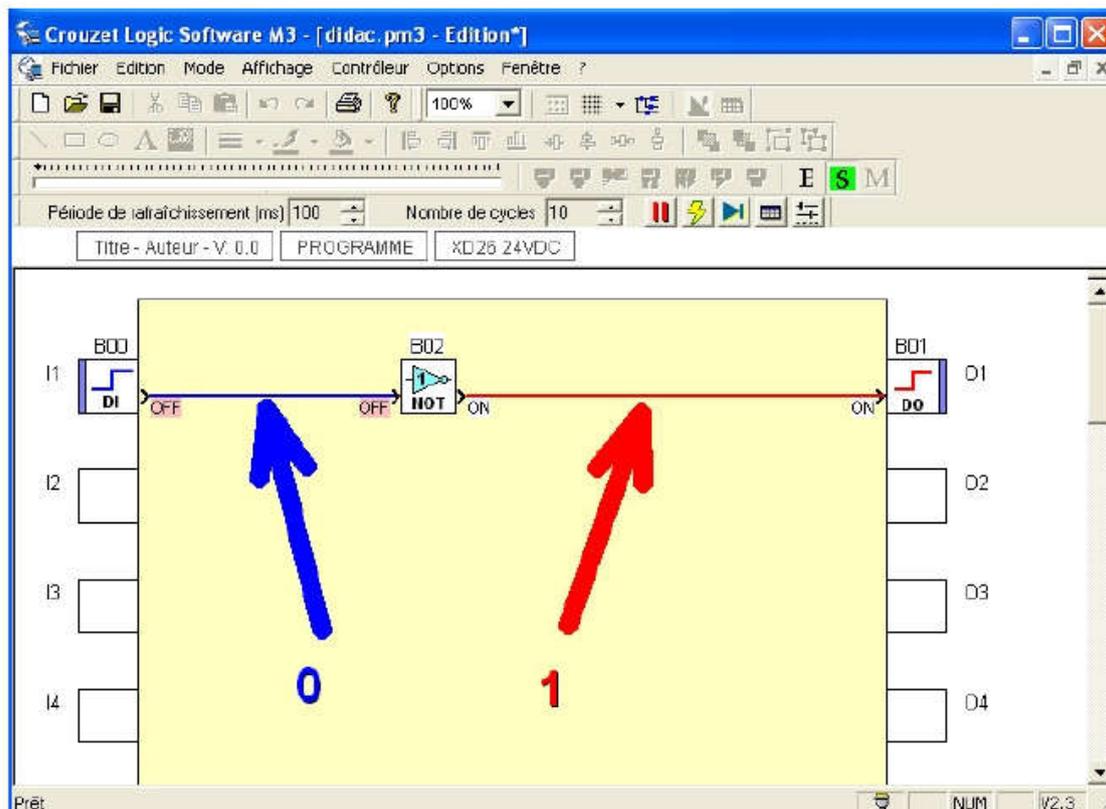
- j. Lorsque vous lancez la Simulation, il se peut que le résultat de la compilation du programme apparaisse : il atteste que le programme écrit a été correctement "traduit" pour être exploité par le simulateur. Si vous ne voulez plus voir cette fenêtre, cochez l'option correspondante :



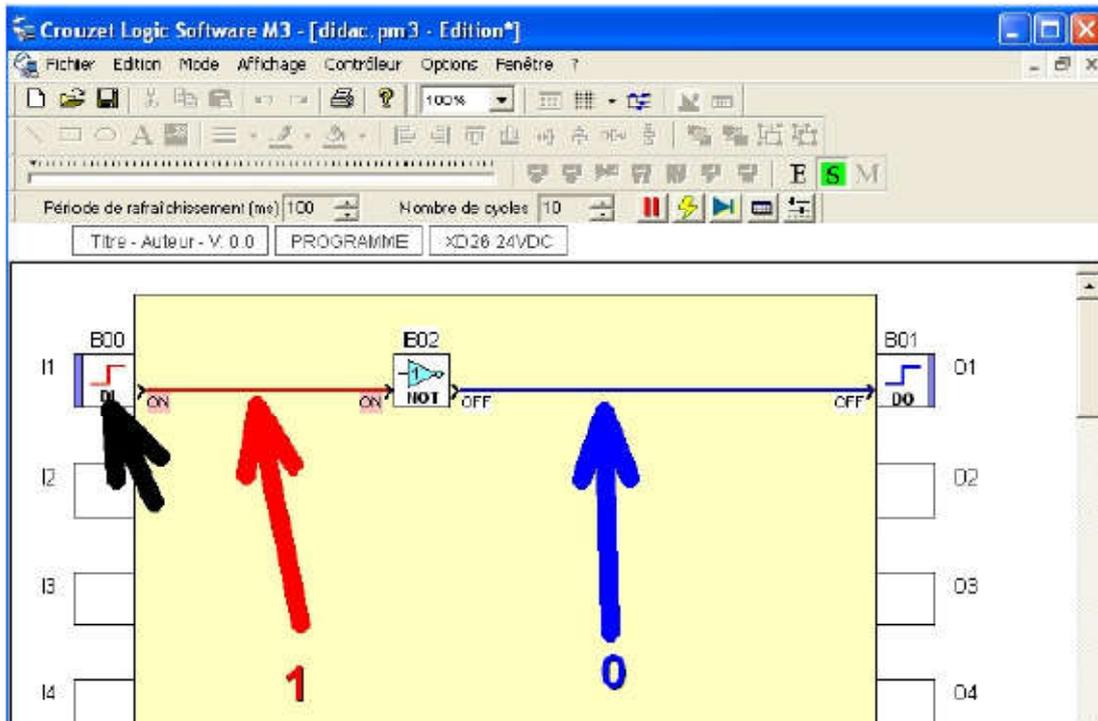
- k. Validez ensuite en cliquant sur "OK".



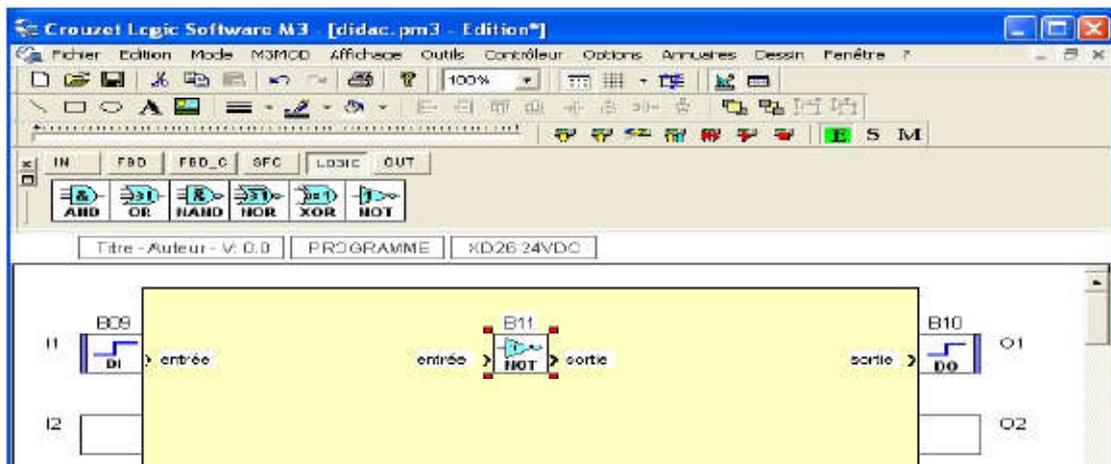
- l. La fenêtre montre alors le circuit dessiné en fonctionnement : une liaison en bleu indique un état logique 0 ; une ligne en rouge indique un état logique 1.



m. Pour changer l'état d'une entrée, il suffit de cliquer dessus :



n. 14. On peut aussi utiliser du texte à la place des fils de liaison. Il suffit de cliquer avec le bouton droit sur une liaison et de choisir : "Type de câblage / Texte" et de remplacer dans un second temps le texte par défaut par celui de votre choix (entrée 1, etc.)



### 4 EXERCICES D'APPLICATION 1 : CIRCUIT LOGIQUE

On vous demande de réaliser le schéma ci-dessous et de remplir la table de vérité ci-après à partir des essais effectués en mode Simulation.

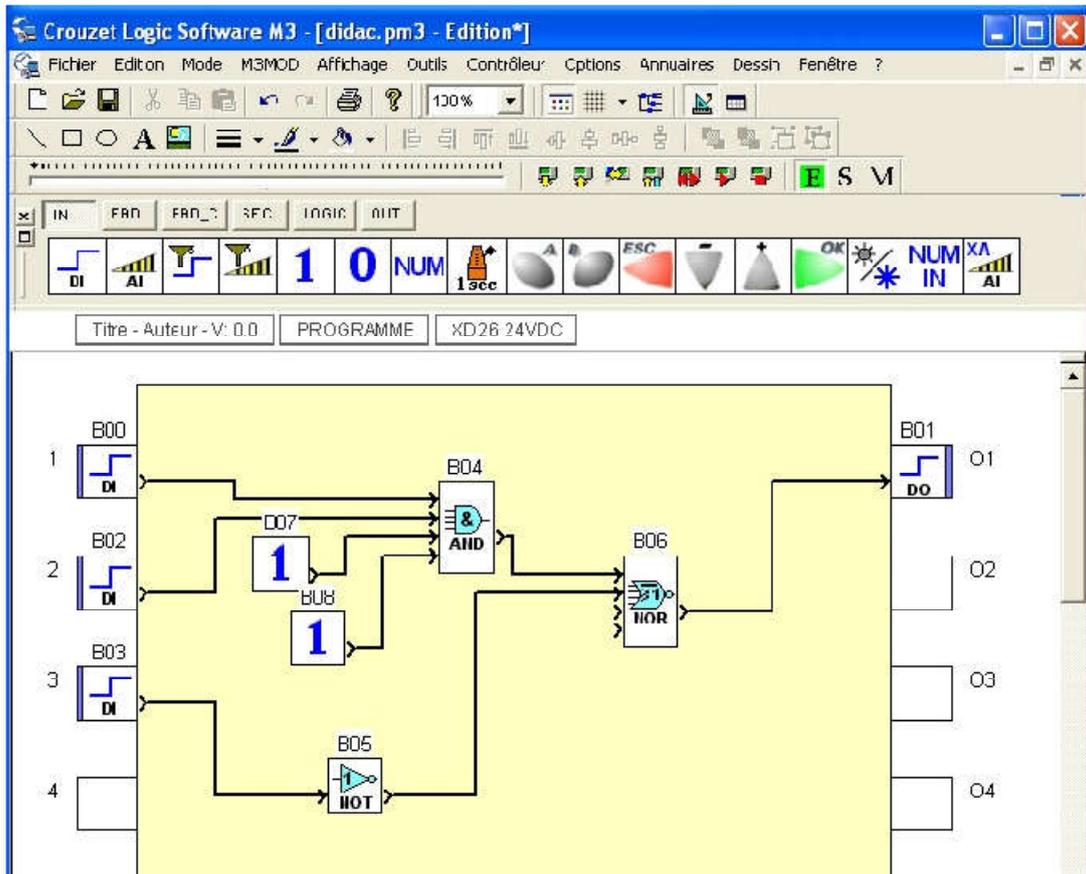
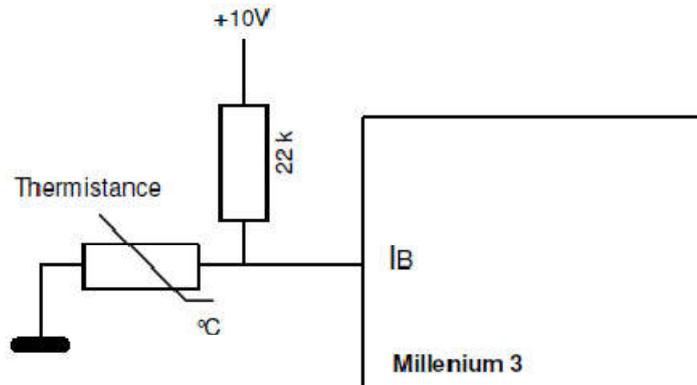


Table de vérité :

I1	I2	I3	Q1
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

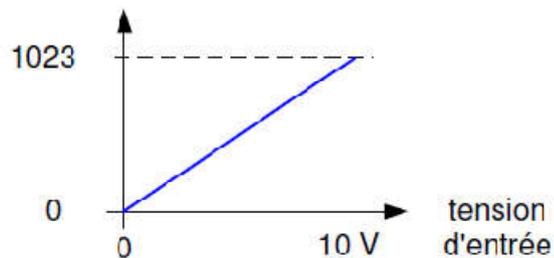
## 5 EXERCICES D'APPLICATION 2 : DETECTION DE SEUIL

Le modèle d'automate Millenium 3 étudié peut aussi gérer des entrées de type analogique (entrées IB à IG), variant entre 0 et 10 V, issue par exemple d'un capteur extérieur :



*Exemple de capteur de température placé sur l'entrée analogique IA du Millenium*

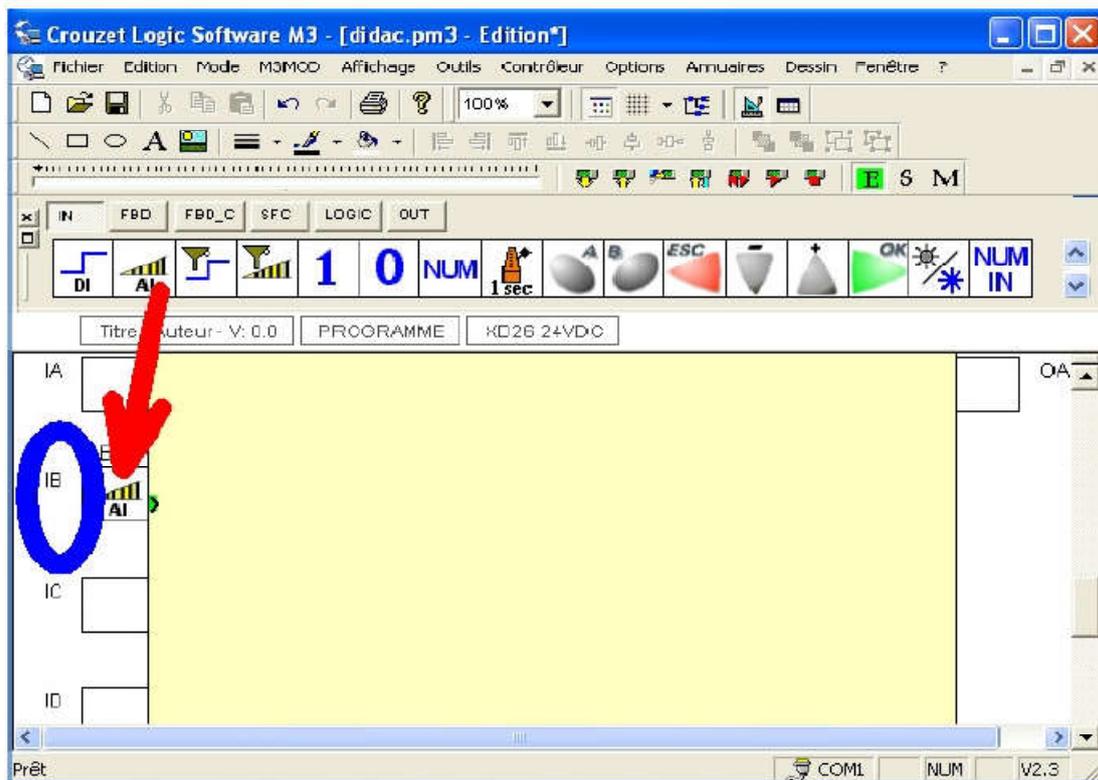
Dans cet exemple, la tension présente sur l'entrée IB du Millenium sera fonction de la valeur de la résistance du capteur, elle-même dépendante de la température. L'automate va effectuer une conversion de cette tension pour la transformer en une valeur numérique comprise entre 0 et 1023 (Conversion Analogique Numérique sur 10 bits) :



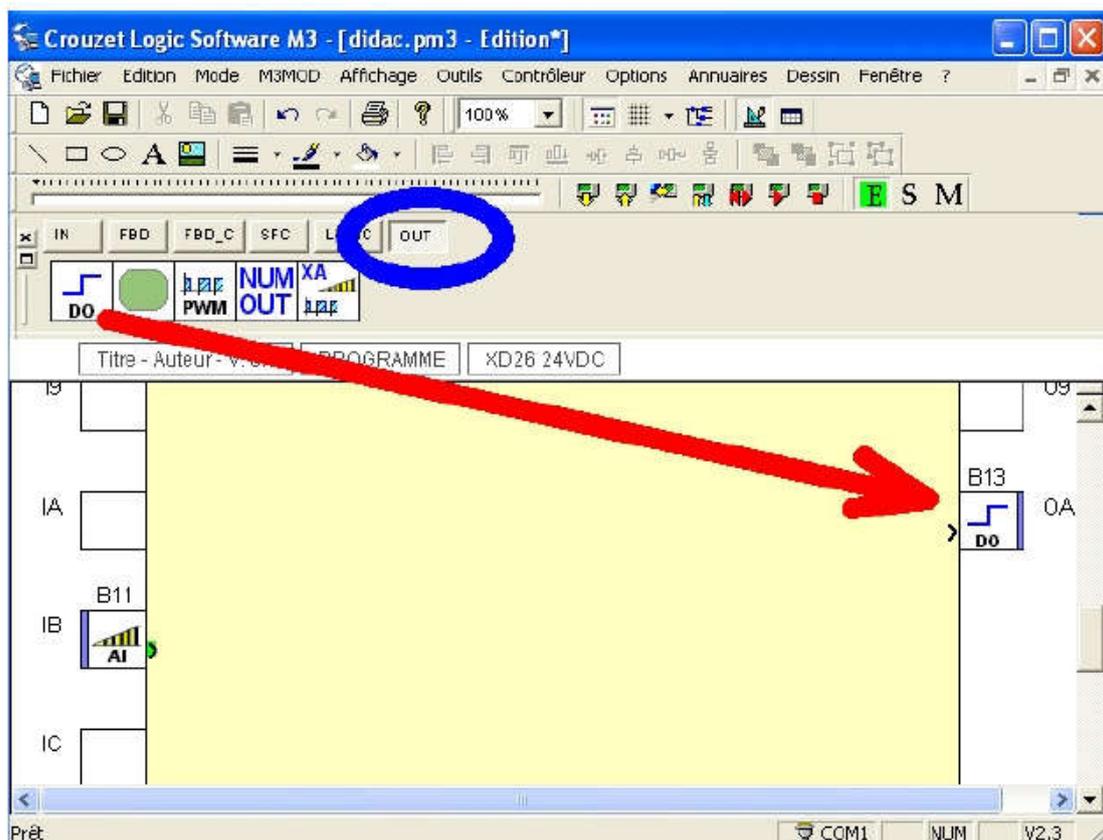
Il est alors possible d'exploiter cette information pour détecter un seuil de tension correspondant dans cet exemple à un seuil de température.

On choisira dans cet exercice de commander une sortie lorsque le nombre issu de la conversion de tension sera supérieur à 314.

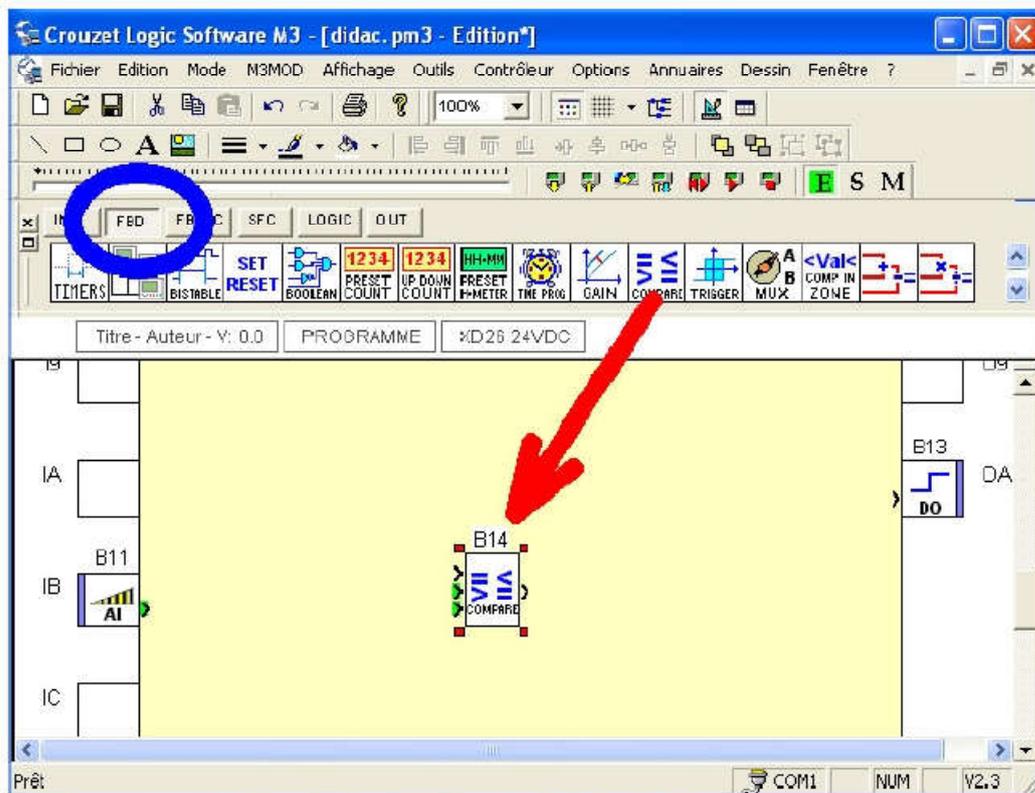
- a. Dans le menu de fonction "IN", déplacez l'entrée "AI" (Analog Input) sur l'entrée IB.



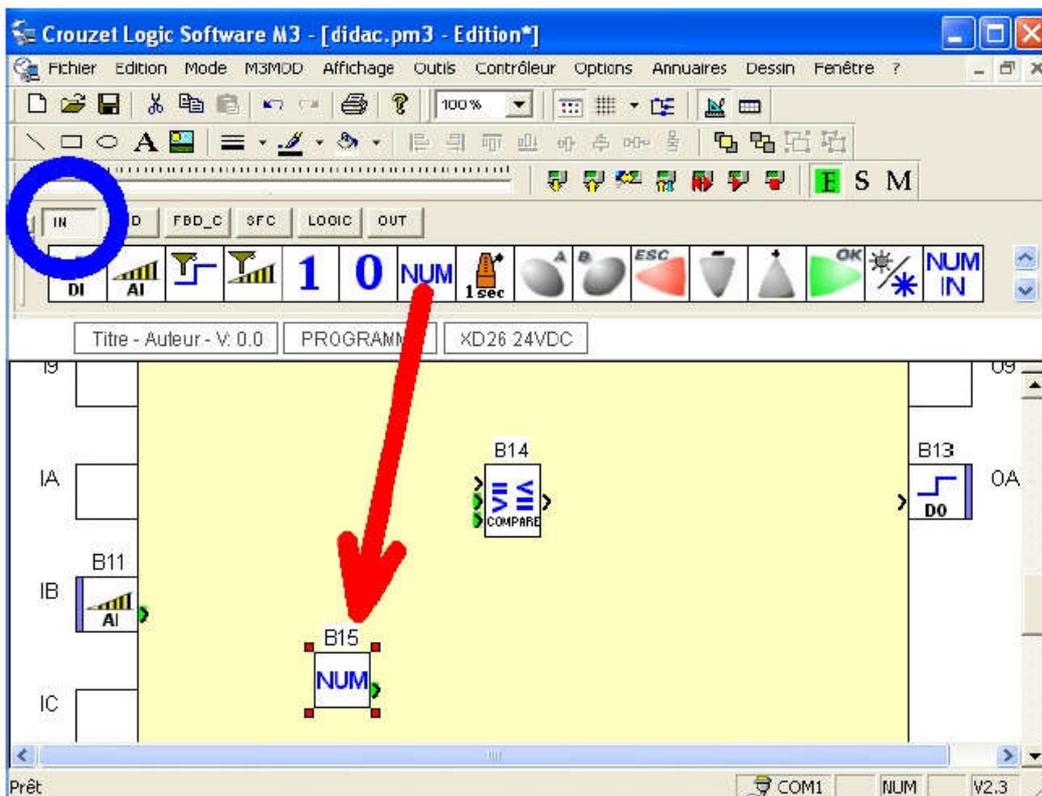
- b. Dans le menu de fonction "OUT", déplacez la sortie "DO" (Digital Output) vers la sortie OA.



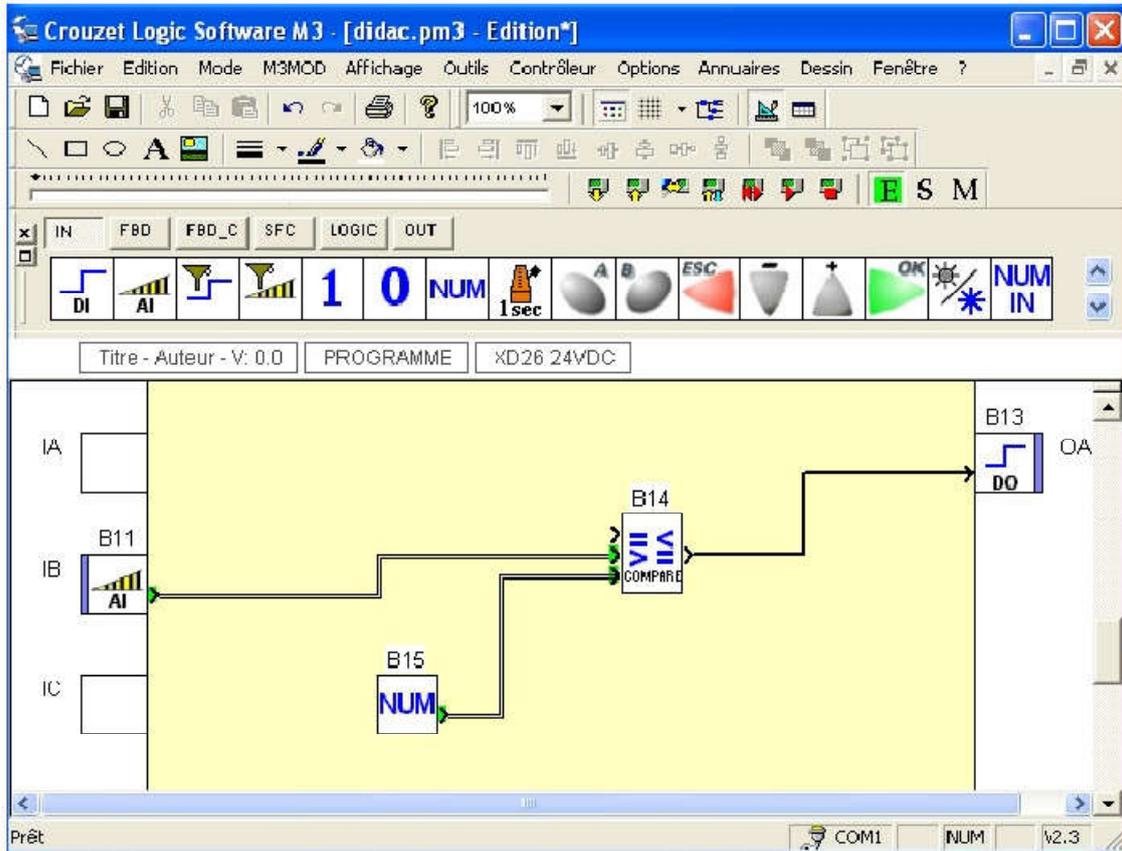
- c. Dans le menu de fonction "FBD", déplacez l'élément "Compare" sur la feuille de travail (cet opérateur compare 2 valeurs numériques).



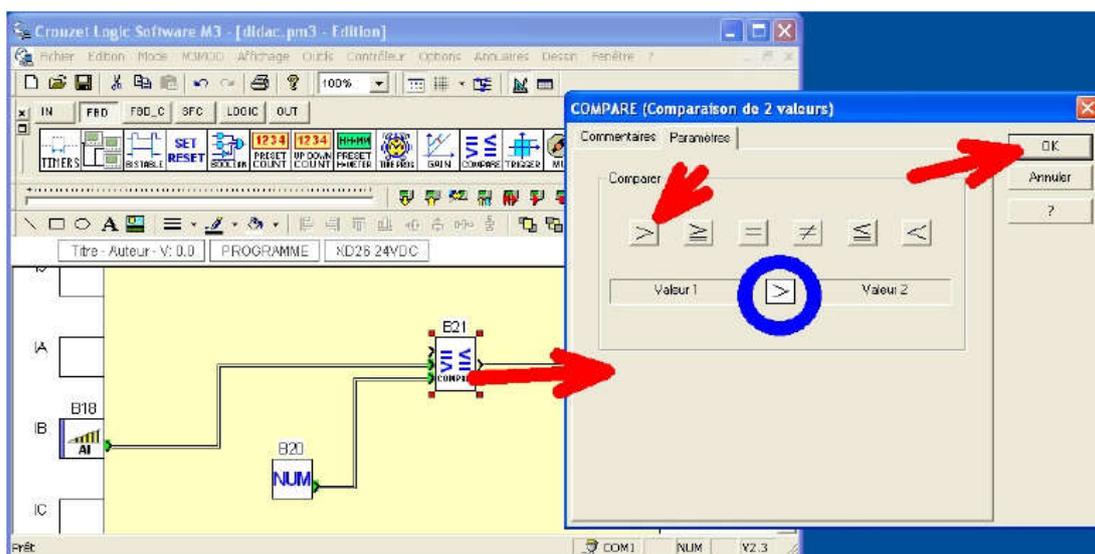
- d. Dans le menu de fonction "IN", déplacez l'élément "NUM" sur la feuille de travail. Cette entrée permet de délivrer un nombre compris entre -32768 et +32767.



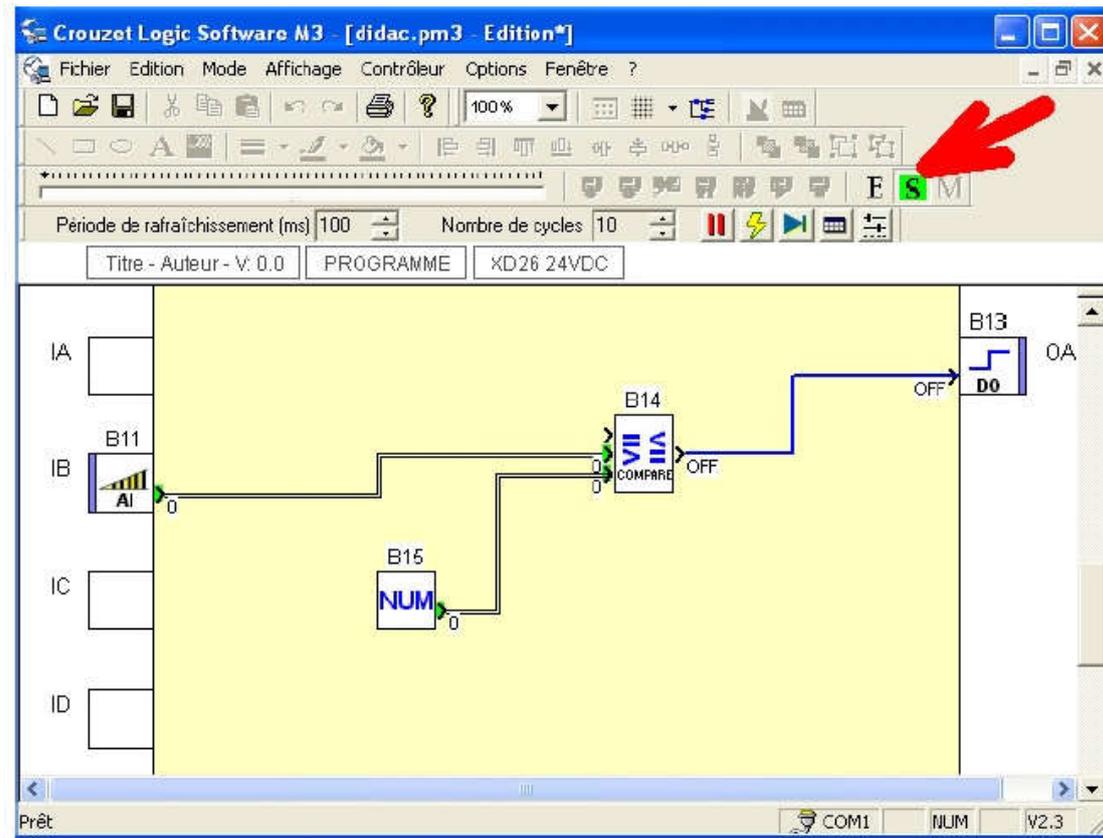
- e. Effectuez les liaisons entre les entrées-sortie et les divers éléments de la feuille. Vous remarquerez que les liaisons "numériques" (entre IB et "Compare" ainsi qu'entre "NUM" et "Compare") n'ont pas la même allure que les liaisons "binaires" (entre "Compare" et "OA").



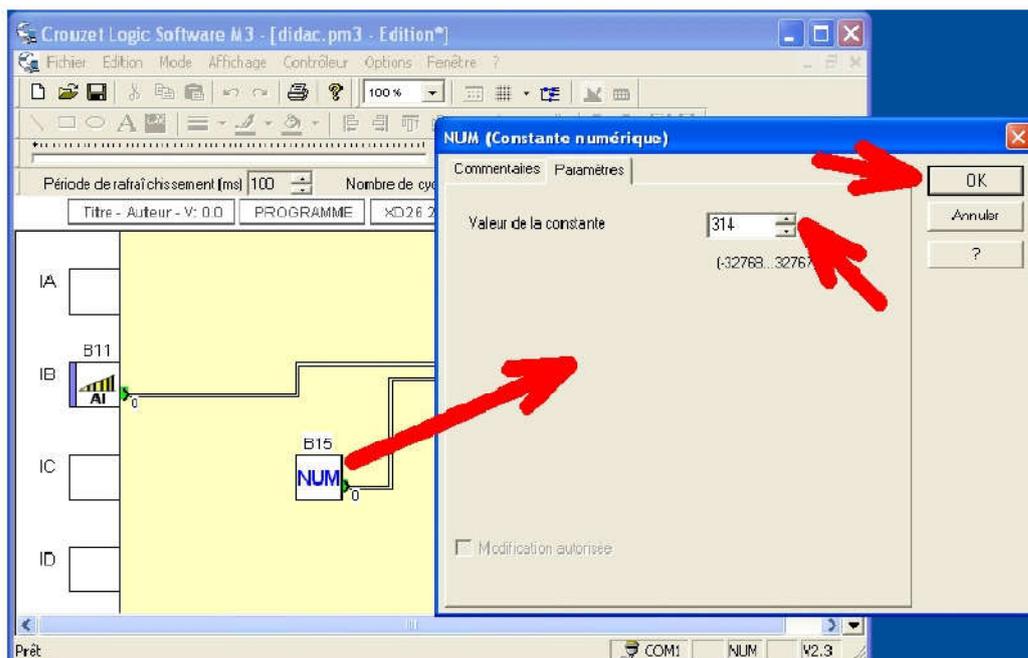
- f. En double-cliquant sur l'élément "Compare", vous ouvrez une fenêtre qui vous permet de choisir le type de comparaison recherché. Optez pour que la sortie soit validée quand "Valeur 1" (l'entrée IB) est supérieure à "Valeur 2" (entrée "NUM").



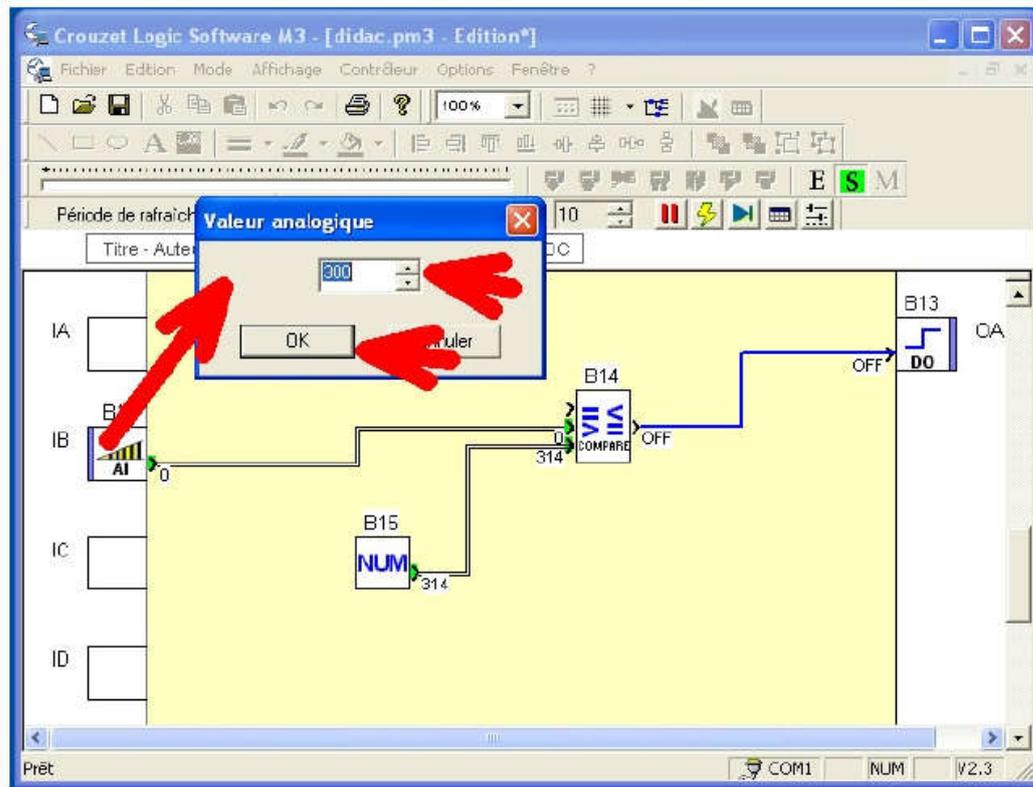
- g. Lancez la simulation (bouton "S"), les valeurs numériques des entrées sont indiquées à côté des liaisons (valeurs par défaut = 0). Les 2 valeurs étant égales, la sortie est sur "OFF".



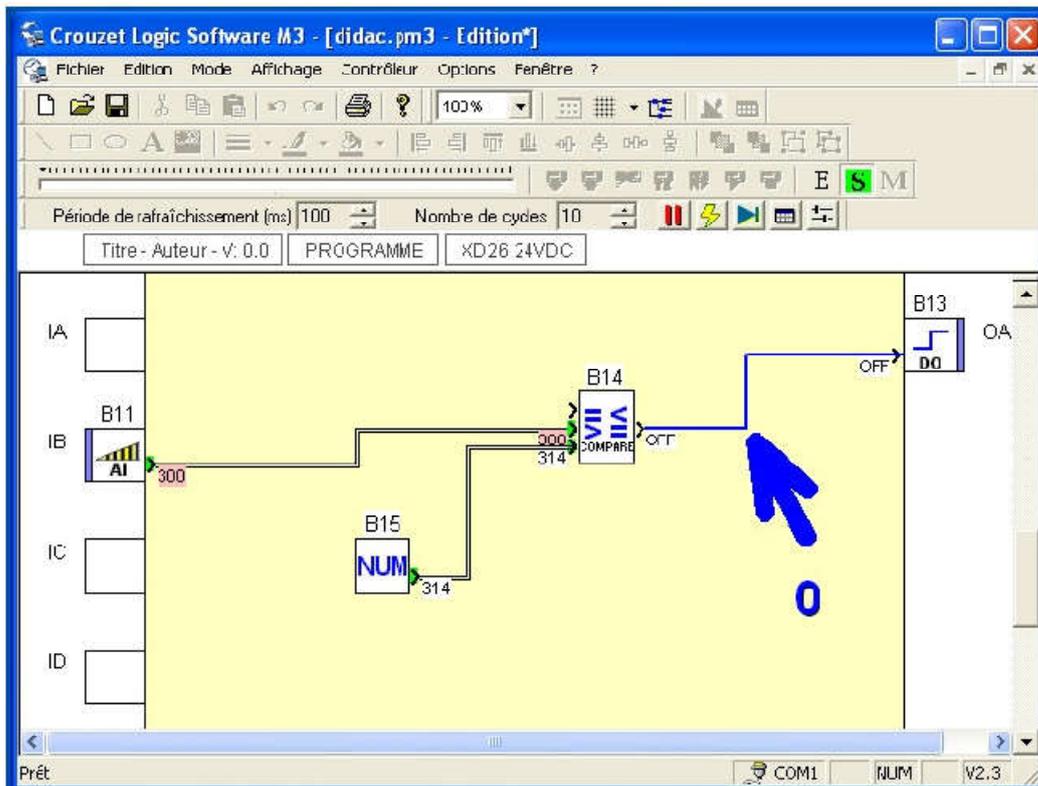
- h. En double-cliquant sur l'élément "NUM", vous ouvrez une fenêtre qui vous permet de choisir la valeur "numérique" délivrée par cet élément. Saisissez "314" puis validez.



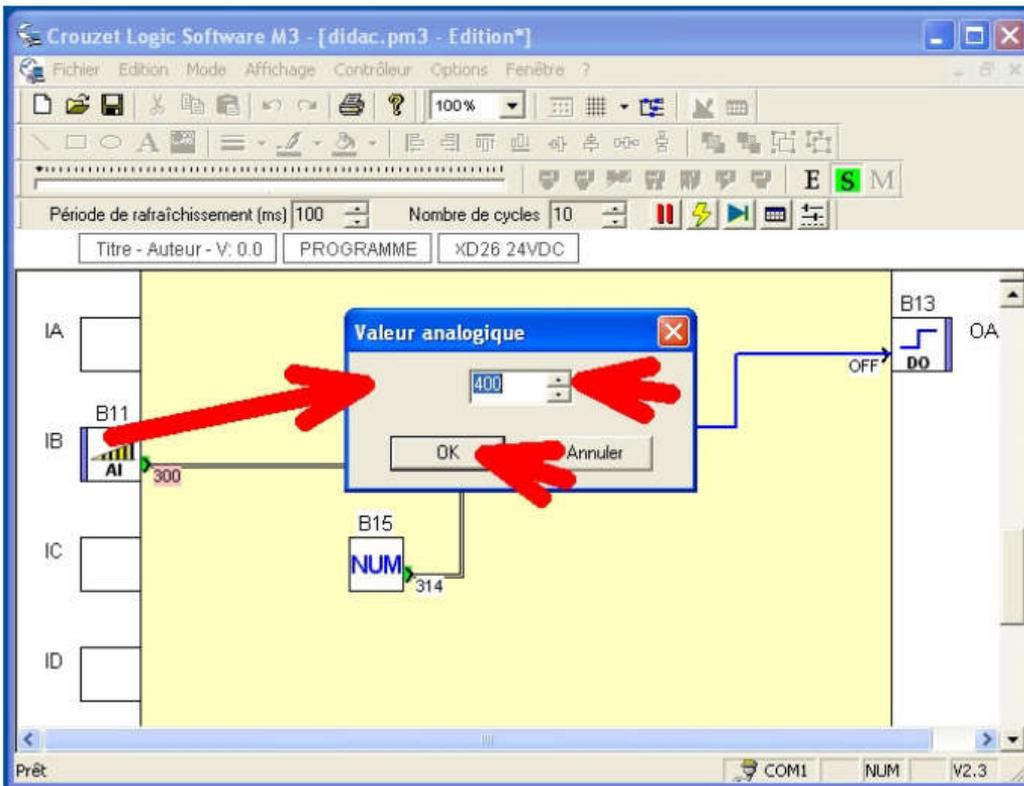
- i. Cliquez sur l'entrée analogique "IB" et imposez une valeur de 300 à cet élément puis validez.



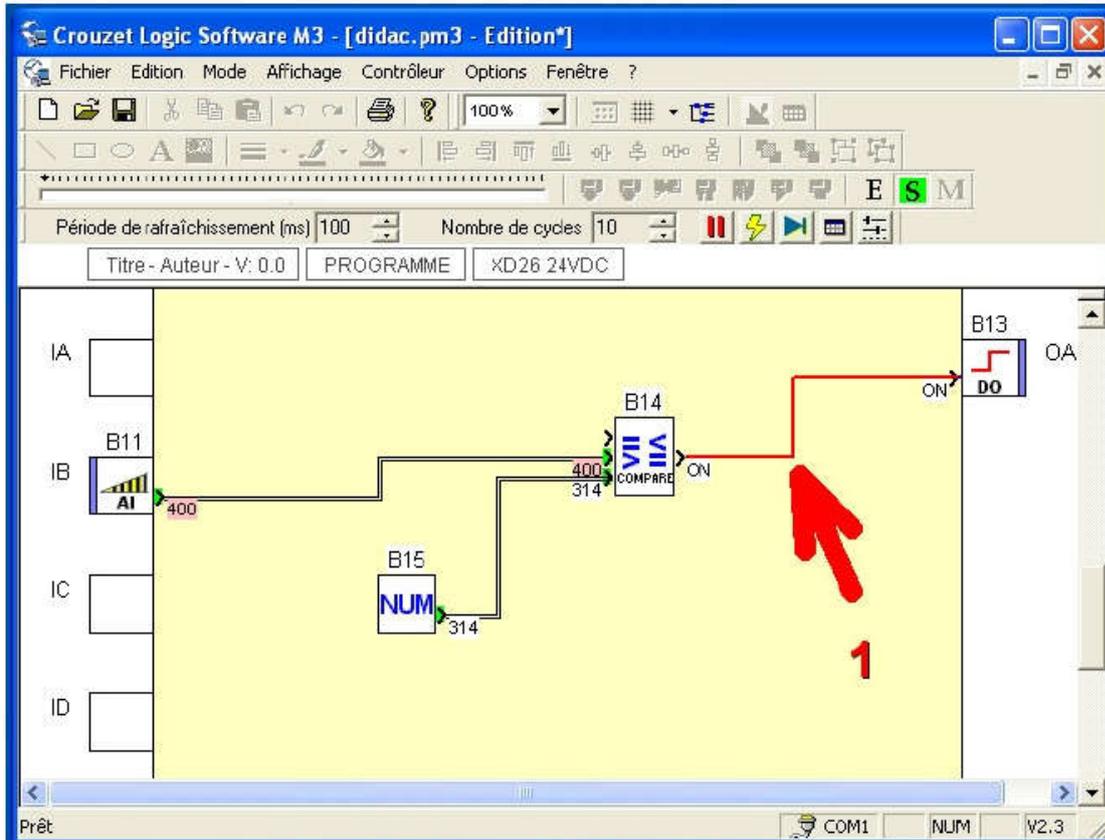
- j. On constate que la sortie reste à 0 car  $300 < 314$ .



k. Choisissez maintenant une valeur supérieure à 314 (400 pour notre exemple) :



l. On constate que la sortie est maintenant validée.



m.

## 6 EXERCICES D'APPLICATION 3 : CIRCUIT LOGIQUE

On vous demande de réaliser le schéma de commande, en langage FBD de trois actionneurs (moteur **M**, vérin **V** et électrovannes **Ev**) :

- a. commande séparée ( bouton poussoir " m" pour commander le moteur **M**, le fin de course " c1" pour commander la sortie de la tige du vérin **V** et " c2" pour commander la rentrée de la tige du vérin **V**
- b. commande alternée après de trois 3 secondes (moteur seul puis après 3 seconde lampe sera allumée et 3 seconde après le vérin se déplace )
- c. commande de deux actionneurs ( moteur et électrovanne) avec action conditionnée par un compteur.