

- des maths dans un API -

Question :

Est-il possible d'exécuter des opérations mathématiques dans un grafcet, sous Automgen ?

Réponse :

Oui !

Question :

Comment ?

Réponse :

Vous n'avez qu'à lire ce qui suit...

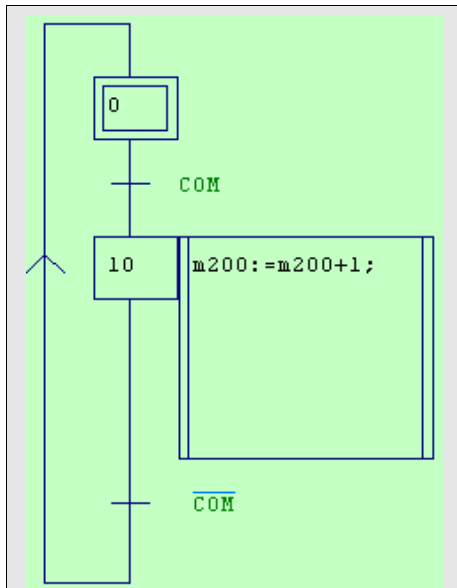


Figure 1 – Ce grafcet permet de voir comment intégrer une opération mathématique. Ici, une addition.

Il faut pour cela utiliser du code, autrement dit, un langage de programmation. On peut se rassurer, pour les opérations mathématiques, ce n'est pas très compliqué donc nul besoin de connaître un quelconque langage.

Comme un exemple vaut bien mieux que dix mille mots, en voici un

Le grafcet de la figure 1 permet de compter de façon perpétuelle, tant que la touche COM (pour commutateur) est pressée.

En étape 10, on a une boîte¹ dont la forme est un peu particulière : elle correspond à une action qui va

¹ Il est à noter qu'il est possible, au sein d'une même étape, d'avoir un rectangle d'action et un rectangle de code.

quelle partie ?	ça signifie quoi ?
m200	nom de la variable qui va contenir le résultat de l'opération.
:=	marque la notion d'affectation : le résultat de l'opération qui se trouve à droite de ce symbole se retrouvera dans la partie qui se trouve à gauche. Ne pas confondre ce symbole avec « = » qui teste une égalité.
m200+1	opération à réaliser.
;	marque la fin d'une ligne de code.

Tableau 1 – Que signifie exactement `m200:=m200+1;` ?

réaliser un petit programme. Ici, ce « petit programme » n'est qu'une ligne : `m200:=m200+1;` (voir tableau 1). On appelle ça une *ligne de code*, le *code* étant l'ensemble des lignes d'un programme.

Tant que l'on sera à l'étape 10, la variable m200 va être incrémentée (au rythme d'un oscillateur interne à l'automate).

Les variables internes utilisables ont un nom compris entre m200 et m9999 et sont des variables 16 bits (elles peuvent donc contenir un nombre compris entre 0 et 65535).

Le code pourra avoir autant de lignes que nécessaire ; il est donc possible de réaliser des calculs nécessitant plusieurs étapes. Les opérations les plus communes sont accessibles (voir plus loin).

Le monitoring

Pour visualiser la valeur de la

variable, il faut ajouter un *objet IRIS* :

- Dans l'arbre d'Automgen (à gauche), faire un clic droit sur Iris ;

- Choisir Ajouter un objet IRIS 2D ;

- Aller dans la branche Objets prédéfinis ~ Eléments de pupitre ~ Afficheurs, cliquer sur l'afficheur désiré, puis sur Ouvrir l'objet ;

- Faire un clic droit sur l'objet pour le déplacer ou le re-dimensionner (figure 2) ;

- L'icône « flèche gauche » permet

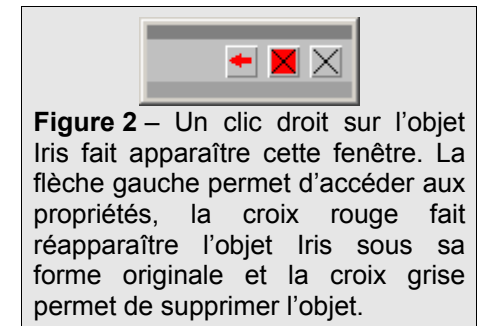


Figure 2 – Un clic droit sur l'objet Iris fait apparaître cette fenêtre. La flèche gauche permet d'accéder aux propriétés, la croix rouge fait réapparaître l'objet Iris sous sa forme originale et la croix grise permet de supprimer l'objet.

d'accéder aux propriétés. Cliquer sur cette icône et aller dans l'onglet Liens ;

- Dans la case Variable ou symbole, saisir le nom de la variable à visualiser (ici m200) et valider par OK.

On peut maintenant lancer le grafcet, valider la réceptivité COM (on clique dessus si on est en simulation),

et voir la variable m200 évoluer.

Dans cet exemple, la valeur de la variable m200 n'est jamais réinitialisée (on peut le faire en cliquant sur l'afficheur de *monitoring* que l'on vient d'ajouter). Voyons un second grafcet, présenté à droite du premier (figure 3), qui initialise la variable lors de l'étape initiale.

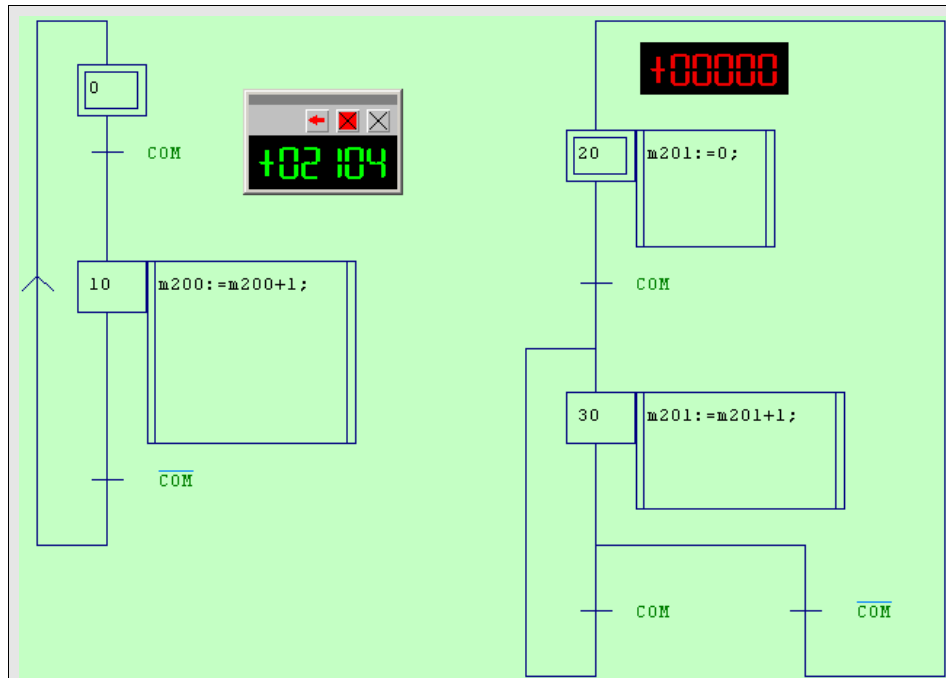


Figure 3 – Le grafcet de droite entraîne une réinitialisation de la variable de comptage m201, ce que ne permet pas celui de gauche. On a aussi ajouté le *monitoring* des deux variables utilisées. Ici, rien n'a été fait de façon esthétique (couleur, emplacement, organisation) car cette figure permet simplement d'avoir un aperçu des possibilités.

opérateur	opération
+	addition
-	soustraction
*	multiplication
/	division
^	élévation à une puissance

Tableau 2 – Les opérations élémentaires sont accessibles. Pour plus de possibilités il faut faire appel au langage de bas niveau. Consulter pour cela le « manuel de référence du langage » d'Automgen, au chapitre 1.10.3.

Comme un deuxième exemple vaut bien mieux que dix mille mots de plus, en voici un deuxième

A l'étape 20, on affecte la valeur 0 à la variable m201 (ce qui correspond à un *reset*) et en étape 30 on compte tant que COM est validé (boucle de gauche). Dès que COM n'est plus validé, on passe par la droite et on revient à l'étape initiale 20. A ce moment, la variable m201 est réinitialisée à la valeur 0 et le grafcet attend que COM soit de nouveau validé. Si l'on connaît la fréquence de travail de l'automate, ce grafcet permet, par exemple, de connaître le temps d'activation du commutateur COM (en fait, l'équivalent d'un chronométrage).

Note : ces exemples de comptage auraient pu utiliser la syntaxe suivante :

RCx pour réinitialiser la variable de comptage Cx (x de 0 à 9999) ;
+Cx pour incrémenter la variable Cx.

Les opérations possibles

Il est évidemment possible d'exécuter plusieurs opérations sur une variable. On peut, dans une ligne de code, entrer une opération à multiples opérateurs (la logique des priorités mathématiques est respectée). Une autre méthode est d'utiliser plusieurs lignes de codes qui correspondent chacune à une opération à exécuter (figure 4, page suivante).

L'ensemble des opérations communes accessibles est listé dans le tableau 2.

Les tests

Très souvent, on fait une opération pour faire travailler le grafcet

différemment selon le résultat obtenu. Il faut donc intégrer des tests à notre grafcet (tableau 3). Le grafcet de la figure 5 utilise ce principe : on veut remplir un flacon de parfum en faisant

faire plusieurs allers et retours à une pompe doseuse d'une contenance de base. Une variable est incrémentée à chaque cycle de remplissage ; de son état dépend l'évolution du grafcet.

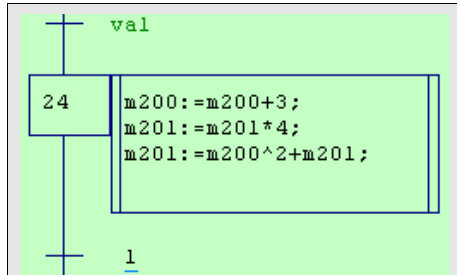


Figure 4 – Quatre opérations sont exécutées ici : deux additions, une multiplication et une élévation au carré.

prof.maquaire@free.fr
http://prof.maquaire.free.fr

opérateur	test
=	égalité
! ou <>	différence
<	infériorité (non signé)
>	supériorité (non signé)
<<	infériorité (signé)
>>	supériorité (signé)
<=	infériorité ou égalité (non signé)
>=	supériorité ou égalité (non signé)
<<=	infériorité ou égalité (signé)
>>=	supériorité ou égalité (signé)

Tableau 3 – L'ensemble des tests disponibles. Plus de détails sont disponibles dans le « manuel de référence du langage » d'Automgen, au chapitre 1.3.6.

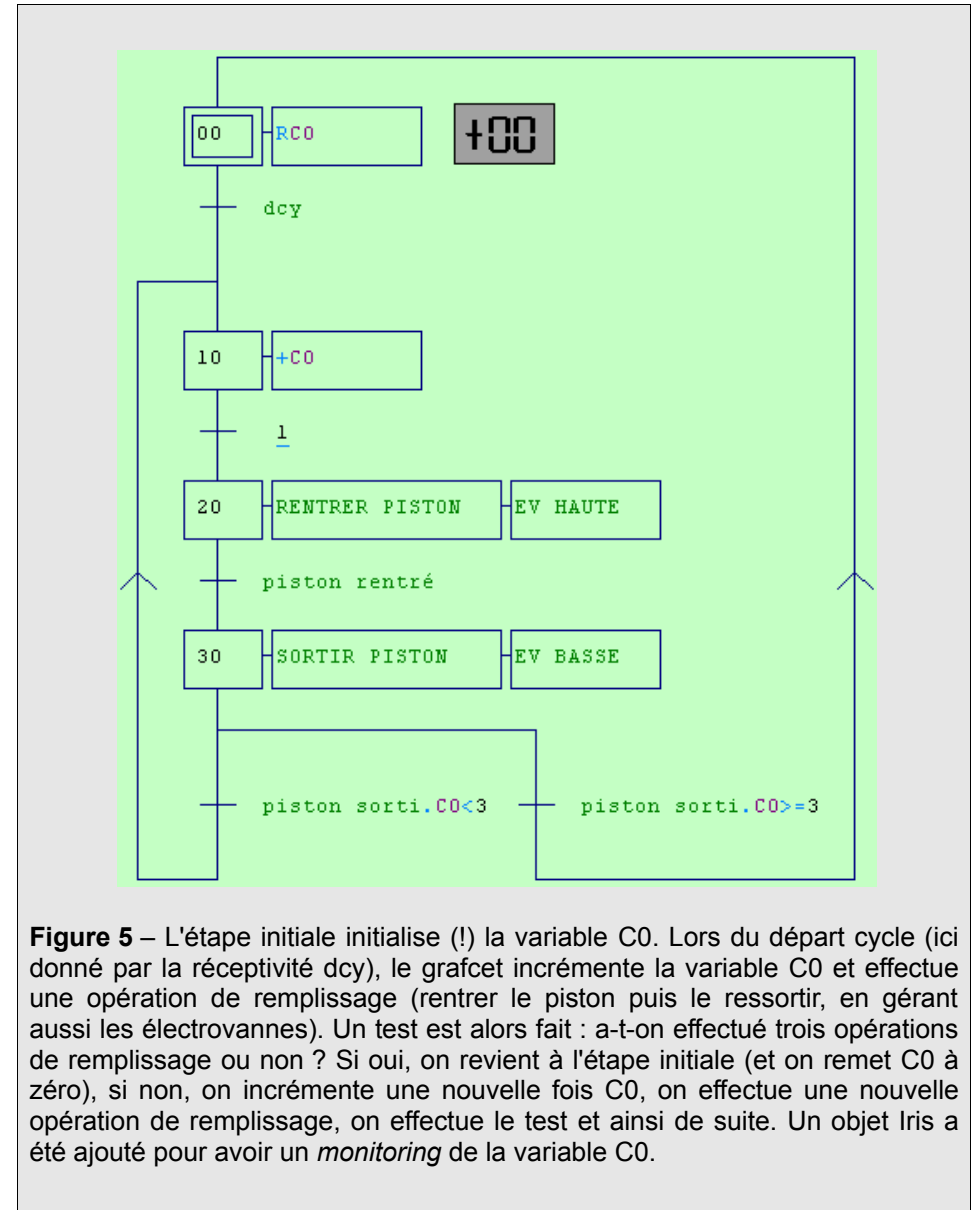


Figure 5 – L'étape initiale initialise (!) la variable C0. Lors du départ cycle (ici donné par la réceptivité dcy), le grafcet incrémente la variable C0 et effectue une opération de remplissage (rentrer le piston puis le ressortir, en gérant aussi les électrovannes). Un test est alors fait : a-t-on effectué trois opérations de remplissage ou non ? Si oui, on revient à l'étape initiale (et on remet C0 à zéro), si non, on incrémente une nouvelle fois C0, on effectue une nouvelle opération de remplissage, on effectue le test et ainsi de suite. Un objet Iris a été ajouté pour avoir un *monitoring* de la variable C0.