

# Utilisation des blocs fonctionnels sous Automgen

## I - Qu'est-ce qu'un bloc fonctionnel ?

Les blocs fonctionnels permettent d'associer à un élément graphique un ensemble d'instructions écrites en langage littéral ou sous forme graphique (logigramme ou grafcet). Ils sont surtout utilisées pour éviter d'écrire le même bout de programme qui se répète plusieurs fois dans le schéma (par exemple pour la fonction Ou-Exclusif : on la crée une seule fois et on peut l'utiliser plusieurs fois dans un logigramme). Les blocs fonctionnels s'utilisent à l'intérieur de schémas de types logigramme ou ladder, ils possèdent de une à n entrées booléennes et de une à n sorties booléennes. Des exemples sont disponibles dans le répertoire `C:\Program Files\IRAI\AUTOMGEN7103\exemples\Bf`.

## II - Comment créer un bloc fonctionnel ?

Pour dessiner un bloc fonctionnel il faut utiliser l'assistant en cliquant droit sur le fond du folio puis en allant dans l'onglet « Blocs-fonctionnels ». On choisit alors le nom, la taille, le nombre d'entrées et le nombre de sorties du bloc.

### Exemple 1 : création de la fonction Ou-Exclusif

Créez dans Automgen un nouveau projet et ajoutez-y un nouveau folio (de type **Normal**) que vous nommerez **principal**. Cliquez droit sur le fond du folio **principal** puis avec l'assistant créez un bloc fonctionnel nommé **exclusif.GR7** possédant 2 entrées et 1 sortie (augmentez la largeur si le nom ne peut pas être saisie en entier).

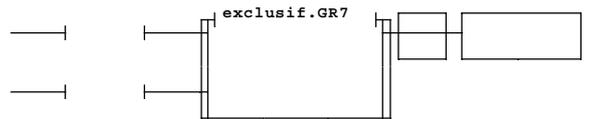


Figure 1 : le folio principal

Ajoutez les délimiteurs d'équation sur les entrées du bloc fonctionnel (raccourcis 6 et 7) ainsi qu'une interface et un rectangle d'action sur la sortie (raccourcis O, w et y) afin d'obtenir le logigramme de la Figure 1. Il faut maintenant programmer le bloc fonctionnel **exclusif** afin que l'équation de sa sortie soit un Ou-Exclusif entre les deux entrées. Pour cela, créez un second folio, éditez les propriétés de ce nouveau folio, donnez-lui comme nom **exclusif** et choisissez comme type **Bloc-fonctionnel**. Sur ce folio **exclusif** nous allons créer le logigramme interne du bloc **exclusif** donnant l'équation de sa sortie en fonction de ses deux entrées. Mais comment s'appellent les entrées et la sortie du bloc fonctionnel ? La première entrées du bloc fonctionnel s'appelle `_{i0}_`, la seconde entrée s'appelle `_{i1}_`, et la sortie s'appelle `_{o0}_`. Cette syntaxe particulière utilisant le caractère underscore (tiret bas obtenu par la touche 8 du clavier) et les accolades permet de ne pas confondre les entrées/sorties du bloc fonctionnel avec les entrées/sorties de l'automate.

Maintenant que vous connaissez le nom des entrées et de la sortie du bloc fonctionnel, créez sur le folio **exclusif** le logigramme de la fonction Ou-Exclusif comme sur la Figure 2.

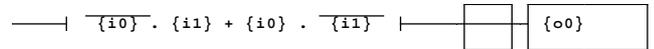


Figure 2 : le folio exclusif

**ATTENTION :** n'oubliez pas de saisir **les caractères underscores avant et après les accolades** même s'ils n'apparaissent pas sur la Figure 2 ni sur votre logigramme une fois qu'il est validé.

Enfin pour tester cette fonction Ou-Exclusif sous forme d'un bloc fonctionnel, mettez deux entrées de l'automate sur les entrées du bloc (par exemple **i8** et **i9**) et une sortie de l'automate à la sortie du bloc (par exemple **o5**) comme sur la Figure 3. Cliquez sur **Go !** et testez.

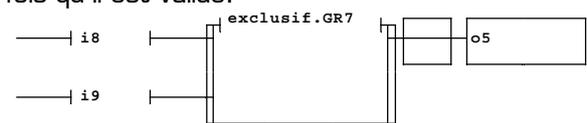


Figure 3 : test du bloc fonctionnel

**IMPORTANT :** ne pas confondre les entrées `_{i0}_` et `_{i1}_` qui représentent les deux premières entrées d'un bloc fonctionnel avec les entrées **i0** et **i1** qui sont les entrées réelles de l'automate. De même pour les sorties, la syntaxe `_{o0}_` [qui n'a de sens que dans un folio de type « Bloc-fonctionnel »] désigne la première sortie du bloc fonctionnel et ne doit pas être confondue avec la syntaxe **o0** qui représente la première sortie réelle de l'automate.

### Exemple 2 : création d'une fonction temporisation

Nous allons ici réaliser une fonction logique possédant une seule entrée et une seule sortie. L'état logique de la sortie devra suivre l'état logique de l'entrée, mais après un temps de retard de 1 seconde. Par exemple, si l'entrée passe de 0 à 1, la sortie passera aussi de 0 à 1 mais une seconde plus tard. Nous verrons en un second temps comment paramétrer cette durée de 1seconde afin de la modifier facilement sans toucher au grafcet.

Créez dans Automgen un nouveau projet, ajoutez-y un folio de type **Normal** que vous nommerez **principal** et un folio de type **Bloc-fonctionnel** que vous nommerez **tempo**. Sur le folio **principal**, créer un nouveau bloc fonctionnel nommé **tempo.GR7** possédant 1 entrée et 1 sortie et intégrez-le dans un logigramme identique à la Figure 4. Sur le folio **tempo**, créer le grafcet de la Figure 5 sans oublier de saisir les caractères underscores avant chaque accolade ouvrante et après chaque accolade fermante. Ce grafcet fait basculer la sortie `_{o0}_` du bloc fonctionnel avec 1 seconde de retard par rapport au basculement de l'entrée `_{i0}_`. Cliquez sur **Go !** et testez cette fonction temporisation en activant l'entrée **i4** de l'automate et en observant l'évolution de la sortie **o6** de l'automate.

**Remarque :** lorsque votre application est en cours d'exécution, un clic sur le nom du bloc fonctionnel dans le folio **principal** permet d'ouvrir de folio **tempo** et de pouvoir suivre en temps réel l'évolution du grafcet.



Figure 4

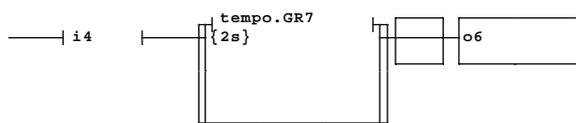


Figure 6



Figure 8

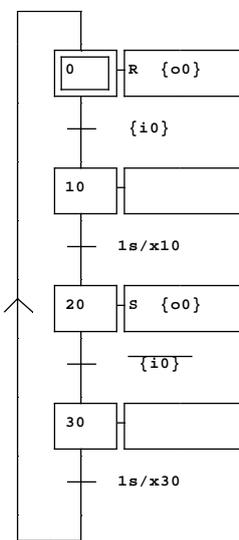


Figure 5

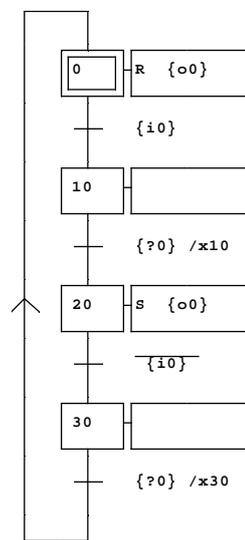


Figure 7

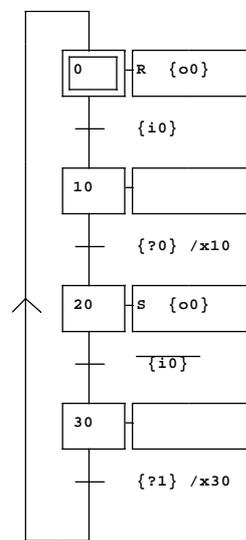


Figure 9

Nous allons maintenant voir comment modifier cette temporisation de 1 seconde. Bien sûr on peut aller modifier directement les réceptivités  $1s/x10$  et  $1s/x30$  dans le grafcet décrivant le bloc fonctionnel. Mais la modification directe du grafcet présente 3 inconvénients :

- \* dans le cas d'un grafcet complexe, ou décomposé en plusieurs grafcet esclaves, on perd du temps à rechercher où se trouve cette temporisation
- \* si la temporisation est utilisée plusieurs fois (c'est le cas ici) il faut penser à modifier une à une chaque temporisation sans en oublier
- \* et si un jour on veut tester une nouvelle valeur de temporisation il faut recommencer toutes ces modifications ...

Pour toutes ces raisons, il serait plus simple de pouvoir inscrire la valeur de la temporisation une seule fois sur le bloc fonctionnel comme sur la Figure 6, ce qui permettrait de la modifier facilement sans éditer le grafcet. Les valeurs écrites sur les blocs fonctionnels des Figure 6 et Figure 8 s'appellent **des paramètres**. On remarque qu'un paramètre s'écrit **entre accolades** sur le bloc fonctionnel. Mais comment récupérer dans le grafcet du folio **tempo** la valeur du paramètre écrit entre accolades sur le bloc fonctionnel ? Pour référencer un paramètre externe écrit sur un bloc fonctionnel, il faut utiliser la syntaxe suivante :

- \* `_{?0}_` représente le premier paramètre passé au bloc fonctionnel
- \* `_{?1}_` représente le second paramètre passé au bloc fonctionnel
- \* `_{?2}_` représente le troisième paramètre passé au bloc fonctionnel, etc.

Modifiez le logigramme du folio **principal** comme sur la Figure 6 sachant que pour écrire un paramètre sur un bloc fonctionnel il faut cliquer sur le coin inférieur gauche du bloc (un clic sur le coin supérieur gauche permet de modifier le nom du bloc fonctionnel). Modifiez ensuite le grafcet du folio **tempo** comme sur la Figure 7, en remplaçant la temporisation constante de  $1s$  par le paramètre `_{?0}_`. Cliquez sur **Go !** et testez cette nouvelle fonction temporisation en essayant différentes valeurs passées en paramètre au grafcet.

Enfin, si on veut donner une valeur différente à chacun des deux retards (retard à la mise à 1 et retard à la mise à 0), il faut utiliser deux temporisations différentes et configurées avec deux paramètres distincts. C'est ce que font le logigramme de la Figure 8 et le grafcet de la Figure 9. Testez-les maintenant.

### III - Ce qu'il faut retenir et désormais appliquer

Pour créer un bloc fonctionnel on utilise **l'assistant**. Le « programme interne » décrivant le fonctionnement d'un bloc fonctionnel doit être dessiné sur **un folio de type « Bloc-fonctionnel »**, et peut être soit un logigramme soit un grafcet. L'appel de ce « programme interne » se fait avec l'extension **« .GR7 »** dans le nom du bloc fonctionnel. Dans le grafcet tracé sur le folio de type « Bloc fonctionnel » il faut entourer les symboles par des caractères underscores [tiret bas obtenu par la touche 8 du clavier]. Dans le folio de type « Bloc fonctionnel », les entrées du bloc fonctionnel sont appelés `_{i0}_`, `_{i1}_`, `_{i2}_`, etc. et les sorties du bloc fonctionnel sont appelés `_{o0}_`, `_{o1}_`, etc.

De manière générale, pour référencer une variable externe d'un bloc fonctionnel il faut utiliser la syntaxe suivante :

- \* `_{in}_` pour référencer l'entrée booléenne **n**
- \* `_{on}_` pour référencer la sortie booléenne **n**
- \* `_{?n}_` pour référencer le paramètre **n**

Enfin, dans l'onglet « Blocs-fonctionnels » de l'assistant, il existe un ensemble de **blocs fonctionnels prédéfinis** qui ne demandent qu'à être utilisés, mais dont les « programmes internes » sont cette fois inaccessibles. La plupart de ses blocs fonctionnels prédéfinis demandent **des paramètres** [à écrire entre accolades] dont le rôle est commenté sur le bloc lui-même. **Tout texte écrit sur un bloc fonctionnel hors accolades est un simple commentaire.**