

Niv : RobIA3	R519 : AUTOMATISMES SECURITE	Cours n°2
Rép : R519	LES MODES DE MARCHES ET D'ARRETS	Page 1 sur 2

Programmation GRAFCET

1) TYPES DE PROGRAMMATION

Le Grafcet peut se programmer de 3 manières différents :

- En utilisant des diagrammes SFC : cette méthode ne présente pas de difficultés majeures quand l'on maîtrise la syntaxe du Grafcet (dont la structure est reprise) et la programmation en ladder (utilisée pour chaque réceptivité). La programmation des sorties et des différents forçages sera différente en fonction du logiciel. L'inconvénient est que toutes les marques d'automates ne la proposent pas dans leur logiciel de programmation.
- En utilisant un langage graphique (Ladder, FBD) première possibilité : chaque étape correspondra à une mémoire booléenne (1 bit) qui sera à 1 lorsque l'étape est active et à 0 lorsque l'étape est inactive. Cette méthode est à privilégier pour les automates d'entrée de gamme et avec un Grafcet très court (mais de plus de 2 étapes pour éviter certains désagréments).
- En utilisant un langage graphique (Ladder, FBD) deuxième possibilité : chaque Grafcet correspond à une mémoire, le plus souvent Word (16 bits, mais qui pourrait être Integer...), dont la valeur correspond à l'étape active (sauf pour les Grafcet en ET où il faudra autant de mémoires que de boucles en ET). Cette méthode est à privilégier dès que la complexité des Grafcet augmente. Cette méthode est possible avec toutes les marques d'automates sauf certains automates d'entrée de gamme.

2) LA PROGRAMMATION PAR MOTS

Les règles qui suivent sont à privilégiées pour une bonne organisation de votre programme :

- Chaque Grafcet sera programmé dans un Block (Siemens ; POU, Schneider...)
- A chaque Grafcet sera affecté une mémoire Word ; exception faite des Grafcet ayant des boucles en ET : à chaque boucle en ET sera affectée une mémoire Word.
- L'activation d'une étape i se fera dans un Network unique (Siemens ; Rung, Schneider...)
- La mise aux étapes initiales se fera, chez Siemens, dans le Block OB100 (start cycle ; seule exception à la règle précédente, utilisation de %S13 chez Schneider...)
- L'affectation d'une valeur d'étape dans une mémoire « Grafcet » se fait avec une instruction Move (Siemens ; Operate, Schneider...)
- Le test d'une étape (active ou pas) se fait avec une instruction de comparaison (égalité = OU pas égalité <>)
- L'activation d'une étape dépendra de la règle 2 du Grafcet : le franchissement d'une transition se produit lorsque toutes les étapes, immédiatement précédentes reliées à cette transition, sont actives et que la réceptivité associée à cette transition est vraie.
- Les sorties seront programmées dans un seul Block, séparé
- L'activation d'une sortie n se fera dans un Network unique (Siemens ; Rung, Schneider...)
- Les sorties ne dépendent que des étapes qui les activent (sauf pour les sorties conditionnelles qui dépendent aussi de ces conditions)

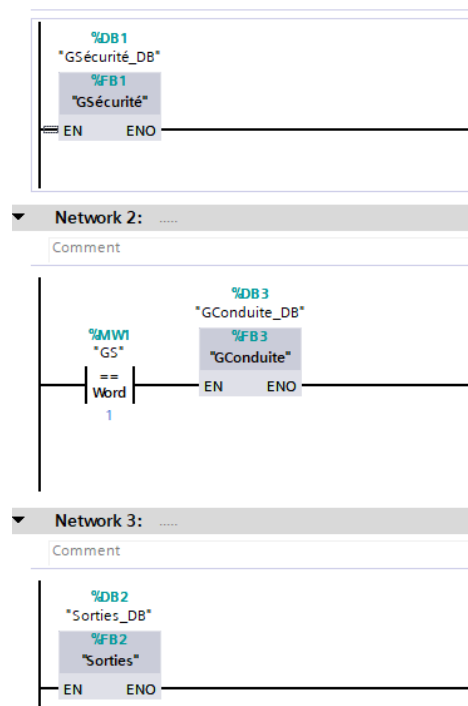
3) EXEMPLE DE PROGRAMMATION PAR MOTS

L'organisation de la programmation va se faire avec un block Main (OB1) qui va activer les Grafcet ; un block Startup (OB100) qui met les Grafcet à leur(s) étape(s) initiale(s) ; un block FB par Grafcet et un block FB pour toutes les sorties

Main [OB1]
Startup [OB100]
GConduite [FB3]
GSécurité [FB1]
Sorties [FB2]

Exemple de Main :

Dans le Network 2 a été introduit un « Figeage » du GC
Si l'on ne se trouve pas dans l'étape 1 du GS



Exemple de Startup :

Exemple de FB (dans le Network 1 est introduit un « Forçage »)

