

**1 - Lecture (et compréhension) du document sur l'automate.** Nous utilisons le Micro1 car les automates Siemens ne sont pas encore accessibles. Familiarisez-vous avec le matériel et logiciel, soit directement sur l'automate, mais aussi en installant le simulateur sur votre PC, sous Linux (cherchez-le sur moodle, décompressez le dans un répertoire, appelez-le via la commande « dosemu »)

**2 – Combinatoire :** Nous allons étudier un distributeur de café. Les produits sont tous en poudre. Trois produits principaux sont disponibles : café (noté K), café décaféiné (D), chocolat (C). Toute distribution fournira un de ces trois produits principaux. En plus, deux produits secondaires sont proposés : du lait (L) et du sucre (S). Le chocolat est toujours servi avec du lait et du sucre. Le "Capuccino" contient une dose K et une dose C (avec obligatoirement une dose de lait ainsi que de sucre).

La machine dispose d'un clavier à 16 touches, délivrant 5 signaux tout ou rien : le premier (nommé "validation") vaut 1 quand une touche est appuyée; les 4 autres correspondent au numéro de la touche (en binaire, nommées  $e_3, e_2, e_1$  et  $e_0$ ). Dans ce TP, nous supposons que la validation se fait par le bouton relié à l'entrée 7, le numéro de touche sera entré par **les entrées 0 à 3**. Le tableau de commande de notre distributeur propose les 10 options ci-contre (les 6 autres touches ne sont pas connectées, profitez-en pour simplifier les équations) :

produit	n° de touche
café non sucré	0
café sucré	1
café au lait	2
café au lait sucré	3
décaféiné non sucré	4
décaféiné sucré	5
décaféiné au lait	6
décaféiné au lait sucré	7
chocolat	8
capuccino	9

Déterminez (sur papier) les équations de cinq sorties (nommées K, D, C, L et S) définissant les produits nécessaires à la préparation choisie, en fonction des 5 entrées (ou alors regardez dans vos notes de cours). Comme dit en cours, analysez chaque sortie séparément, commencez par K, sans traiter les autres, en utilisant un tableau de Karnaugh, en ne tenant pas compte (pour ce tableau) de l'entrée de validation. Mais vous en tiendrez quand même compte dans l'équation définitive. Programmez K, essayez, et montrez le moi (si je suis disponible) puis traitez les 4 autres sorties D, C, L et S. Puis programmez l'automate pour commander ces 5 sorties (sur 200 à 204) en utilisant principalement les instructions LOD, OUT, AND, OR, NOT, AND LOD, OR LOD.

**3 – séquentiel :** D'abord, résolvez complètement le problème séquentiel vu en cours (portique pouvant déplacer des pièces d'un point à un autre) : au départ cycle, on monte (pour soulever la pièce), puis on avance, on descend, puis monte à nouveau, on recule, et enfin on redescend. Vous programmerez cela, en utilisant les instruction SET et RST (et bien sûr les autres instructions combinatoires utilisées à la question précédente).

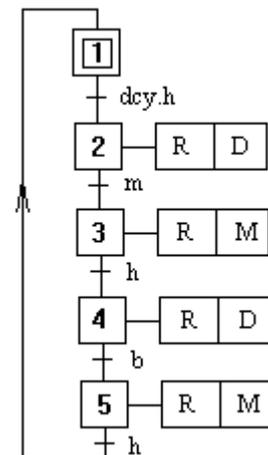
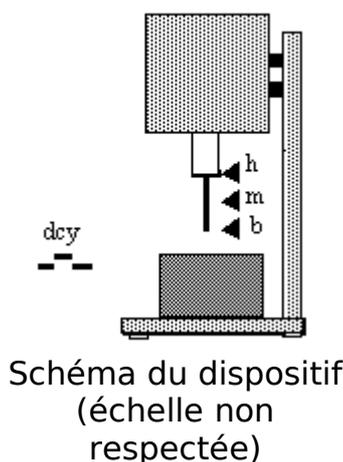
**4 – séquentiel (Grafcet) en langage mnémotechnique :** Vous allez maintenant acquérir progressivement une méthode permettant de traiter tout Grafcet sur tout automate programmable (il lui suffit de posséder les fonctions de base du combinatoire (ET, OU, NON) et les bascules).

**A - Perçage avec débouillage**

On utilise

trois sorties : montée M (200)  
descente D (201)  
rotation R (202)

quatre entrées : haut h (0)  
milieu m (1)  
bas b (2)  
départ cycle dcy (3)



**Cahier des charges** : Au départ cycle (condition initiale broche en haut), descente de la broche jusqu'en m, remontée en h (pour débourrer, c.a.d sortir les copeaux), descente jusqu'en b puis remontée en h, pour attendre un nouveau dcy. Le foret doit tourner tout au long du cycle. Je vous impose le Grafcet à utiliser. Programmez ce Grafcet en respectant les consignes : chaque étape est représentée par une mémoire interne (400 et plus) qui vaudra 1 lorsque l'étape est active, 0 sinon. Le programme se décomposera en 3 parties : initialisation, évolution puis affectation des sorties, détaillées ci-dessous.

- A l'initialisation, on allume l'étape initiale et on éteint les autres mémoires. Utilisez l'entrée 7 comme bouton Reset.

- L'évolution est traitée **transition par transition**. Une transition est franchie lorsque l'étape précédente est active et que sa réceptivité est vraie. On désactive alors l'étape précédente et on active la suivante.

- Pour la clarté du programme, les sorties seront traitées obligatoirement après de l'évolution : on modifie les sorties en fonction des étapes actives (et pas des capteurs !).

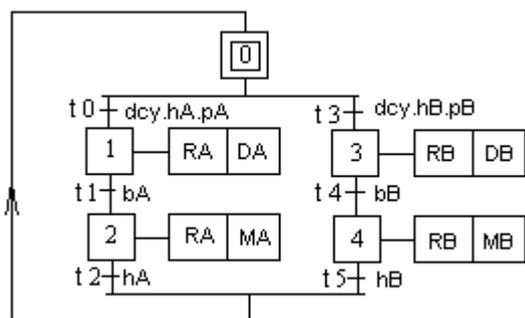
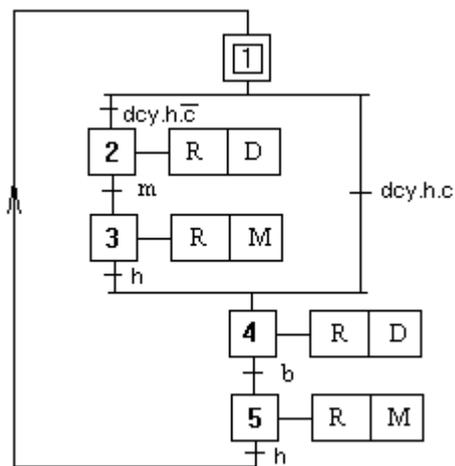
**B** - Rajoutons un interrupteur bistable c (sur l'entrée 4), commandé par l'opérateur, qui, s'il est appuyé, court-circuite le débouillage. Modifiez votre programme (il me semble qu'il suffit d'ajouter une transition). Le Grafcet est donné ci-contre (je suis gentil, non ?).

**C** - On dispose désormais de deux postes de perçage A et B (pour simplifier on ne prévoira plus de débouillage), avec un seul dcy mais deux capteurs de présence. Au dcy, on démarre les perçages pour les postes prêts (c'est à dire en position haute avec présence de pièce). Le Grafcet est donné ci-dessous. La méthode de programmation utilisée jusqu'ici ne respecterait pas les règles du Grafcet. Une bonne solution permettant de traiter ce cas (et d'ailleurs tous les cas) est de prévoir également une mémoire pour toutes les transitions (ici t0 à t5). On décompose la phase d'évolution en trois :

- calcul de toutes les transitions (définir si elles sont franchissables ou non) sans faire évoluer le Grafcet (ceci permet de respecter la règle 4 du Grafcet).

- désactivation de toutes les étapes à désactiver en fonction des transitions franchissables (mais pas encore de changement des sorties)

- activation de toutes les étapes à activer, toujours en fonction des transitions franchissables (ceci permet de respecter la règle 5 du Grafcet).



CAPTEURS :

hA haut A  
bA bas A  
hB haut B  
bB bas B  
pA présence pièce A  
pB présence pièce B  
dcy

ACTIONNEURS :

MA montée A  
DA descente A  
RA rotation A  
DB descente B  
MB montée B  
RB rotation B