

**Centre d'intérêt** : Logique séquentielle

**Durée** : 2 h00

# LEGO MINDSTORMS

**Programmation d'un robot exploration**

**Objectif** : Analyse et amélioration d'un diagramme d'états pilotant une Partie Opérative.

# ROBOT MINDSTORMS



Robot aspirateur Roomba



Robot tondeuse à gazon



Robot nettoyeur de piscine

## **1- Mise en situation :**

De nombreux robots domestiques autonomes ont un fonctionnement dit « d'exploration » : ils doivent couvrir un maximum de la surface d'une zone considérée. Par exemple, le robot aspirateur Roomba doit aspirer la plus grande surface possible d'un logement avant que ses batteries ne soient vides, auquel cas, il retourne à la base de chargement.

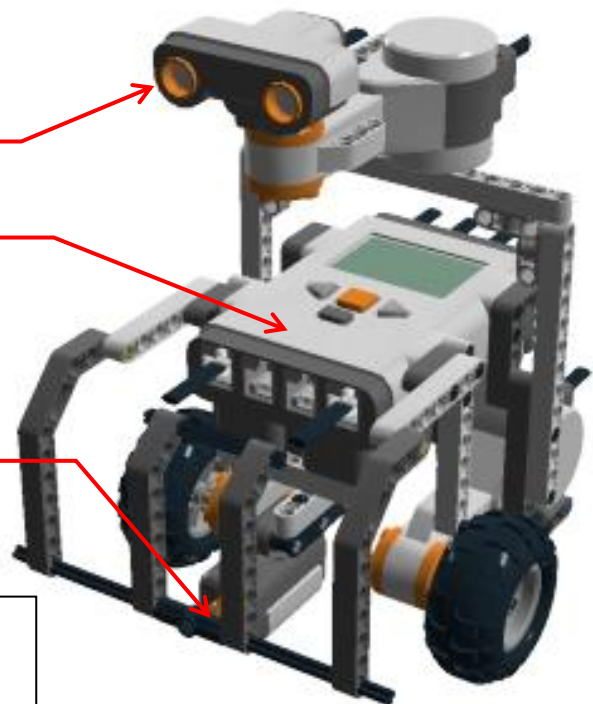
On utilise ici, pour étudier ces modes de déplacements un robot construit à l'aide d'éléments Lego Mindstorms.

Capteur à ultrasons

Brique NXT

Capteur à contact de détection d'obstacles (capteur tactile)

N.B. : Un capteur du robot utilisé n'est pas représenté sur cette image : il sera étudié ultérieurement



Il est constitué :

- D'une Partie Commande : la brique NXT que l'on peut programmer depuis l'ordinateur via un câble USB. On utilise dans ce TP le langage Diagramme d'états et le logiciel Automgen.
- D'une Partie Opérative équipée :
  - des actionneurs suivants :
    - Moteur roue gauche
    - Moteur roue droite
    - Moteur d'orientation du capteur à ultrasons
  - des capteurs suivants :
    - Capteur à contact (ou capteur tactile)
    - Capteur à ultrasons

**Dans ce TP, on ne modifie pas la construction du robot, mais on modifie le programme de la partie commande.**

## **2- Observation du fonctionnement du robot**

On propose un diagramme d'états très simple pour tester le fonctionnement :

Ouvrir une session sur le PC en tant que « prof » (pas de mot de passe)

Lancer le programme AUTOMGEN, icône :

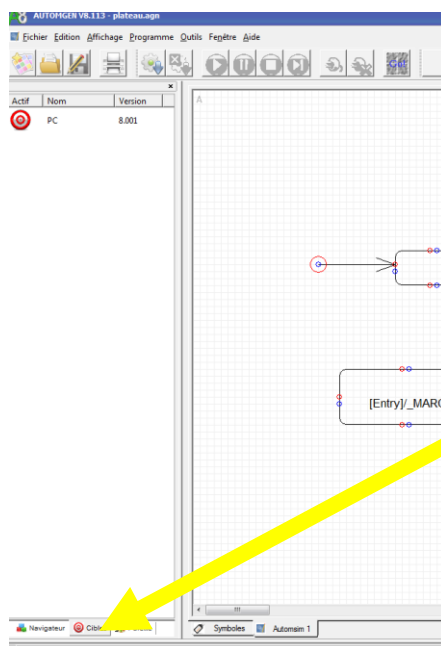


**Charger les fichiers nécessaires depuis le site de la classe**

**Menu : Fichier / Ouvrir le fichier « lego.AGN »**

**dans le répertoire Mes documents \ Lego Mindstorms \**

**Choix de la cible :**



**Si le diagramme d'états ne s'affiche pas, double cliquer sur *Lego* dans l'onglet Automsim**

**Sélectionner l'onglet Cibles puis double-cliquer sur la ligne RCX/NXT**

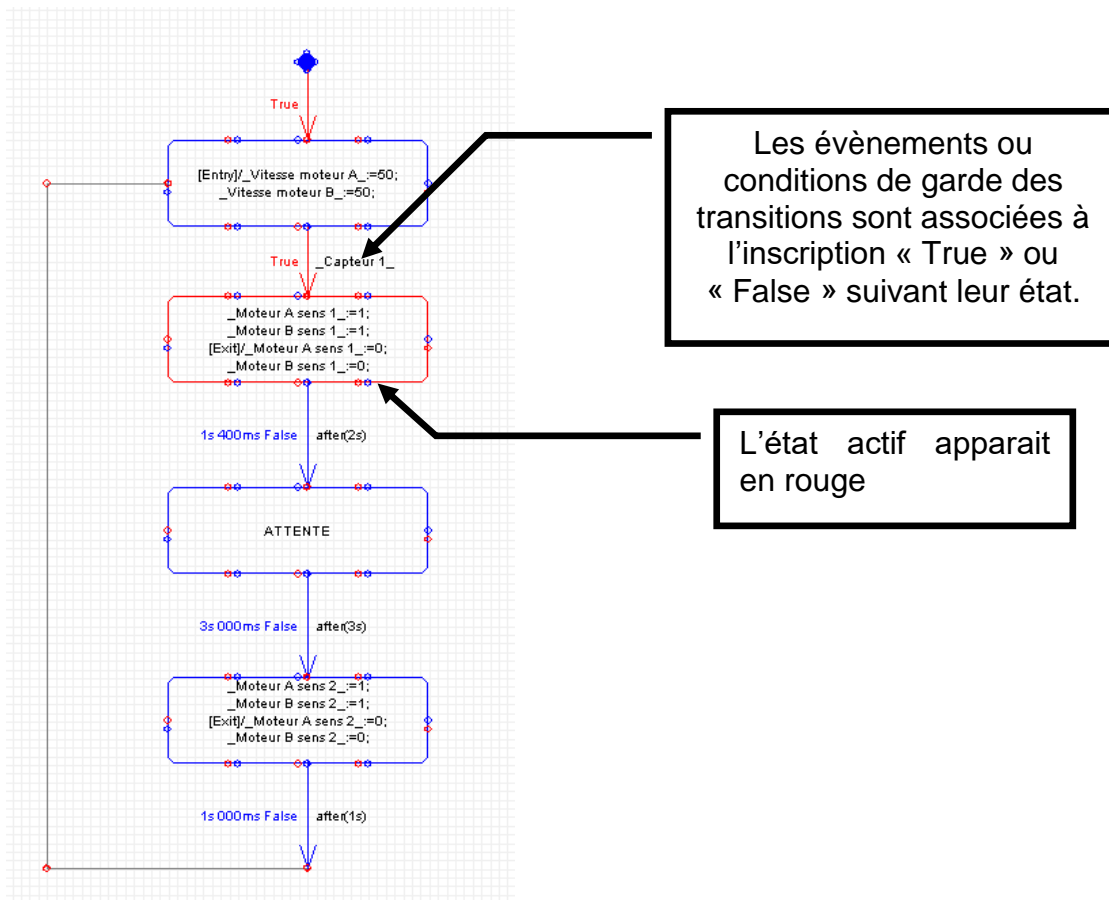
## CHARGEMENT DU PROGRAMME SUR LA PARTIE COMMANDE

Allumer le robot avec le bouton orange, vérifier qu'il est connecté à un port USB.

## BIEN SURVEILLER LE ROBOT PENDANT LE CHARGEMENT (ET APRES) POUR EVITER QU'IL NE TOMBE

Cliquer sur l'icône « GO ! » :  ... et attendre quelques secondes le chargement

Le diagramme d'états est alors animé :



### Remarque :

Il est possible de modifier la taille d'affichage du diagramme d'états avec les commandes Zoom +



et Zoom - :



Lancer le fonctionnement du robot en appuyant sur le capteur tactile (capteur 1) situé à l'avant du robot.

Observer le comportement du robot et les évolutions du diagramme d'états correspondantes.

**Remarque :**

1/ la vitesse des moteurs se règle avec les variables **Vitesse moteur A** (roue droite), **Vitesse moteur B** (roue gauche), et **Vitesse moteur C** (orientation du capteur à ultrasons, vu plus tard), avec une valeur entre **0** (arrêt) et **100** (vitesse maxi).

2/ **Capteur 1** est une variable booléenne associée au capteur à contact détecteur d'obstacles

Cliquer sur l'icône « GO ! » :  pour arrêter l'animation.

### 3- Mise au point d'un diagramme d'états de commande de fonctionnement simple

On désire obtenir le fonctionnement suivant :

- Le robot ne démarre que si l'on appuie sur le deuxième capteur à contact situé sur la partie supérieure du robot (**Capteur 3**)
- Le robot avance en ligne droite à vitesse réduite (50%)
- Si un obstacle est détecté avec le capteur à contact :
  - o le robot s'arrête
  - o recule de 10 cm **environ**
  - o tourne de 90° vers la droite
  - o et reprend son fonctionnement initial

**Q 1 Proposer un diagramme d'états correspondant à ce fonctionnement sur le document réponse (compléter l'amorce de diagramme d'états proposé).**

Tester la solution proposée, pour cela :

Modifier le diagramme d'états initial sur le logiciel Automgen en tenant compte de la remarque ci-dessous.

**Pour en savoir plus sur l'utilisation d'Automgen : se reporter à la partie « CREER ET MODIFIER UN DIAGRAMME D'ÉTATS » du Dossier Technique (plus loin dans cette pochette)**

**Remarque importante :**

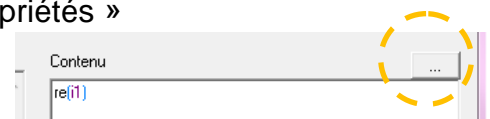
Pour éviter les erreurs lors de la saisie des noms de variables, on peut les choisir dans une liste et procéder de la manière suivante :

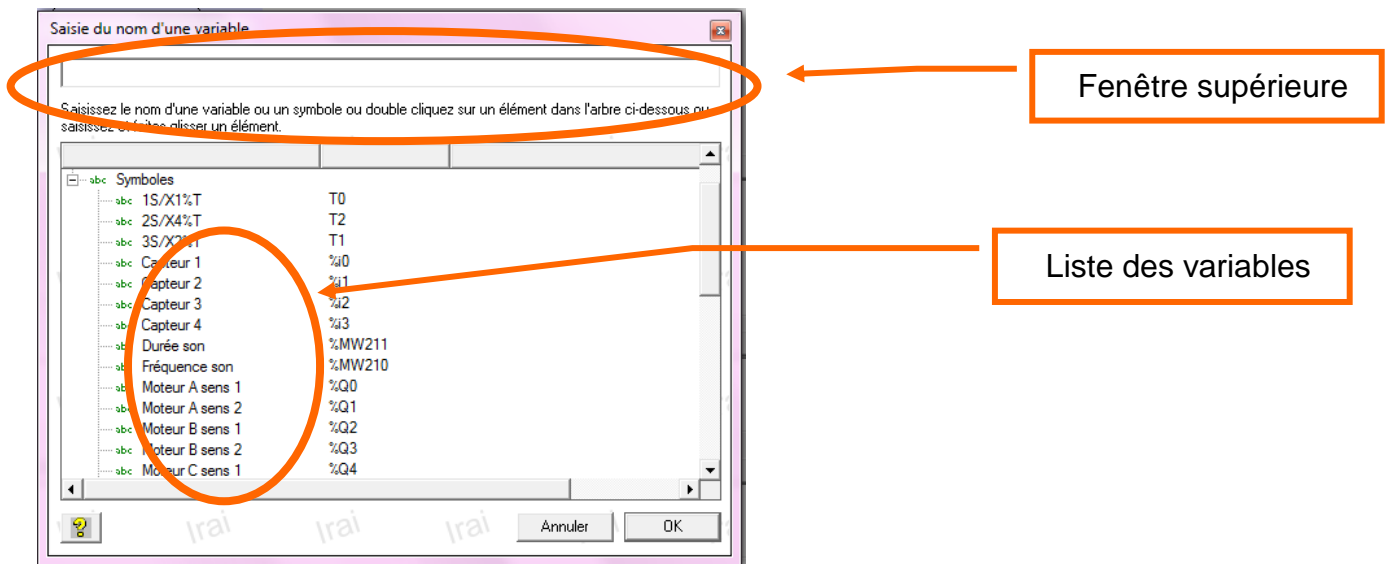
1- Cliquer-droit sur la transition ou sur l'état, et choisissez « propriétés »

2- En haut de la zone « contenu » ou « Entry » ou « Exit »

(suivant les cas), cliquer sur la case  qui apparaît

3- Choisir « Symboles » et double-cliquer dans la liste proposée la variable souhaitée pour la faire apparaître dans la fenêtre supérieure. Dans cette fenêtre on peut ajouter les opérateurs logiques désirés ( +, ., / etc.) ou les virgules entre deux ordres.





**Vérifier le diagramme d'états et le télécharger sur le robot :**

Cliquer sur l'icône « GO ! » :  et attendre quelques secondes le chargement :  
**BIEN TENIR LE ROBOT PENDANT LE CHARGEMENT POUR EVITER QU'IL NE TOMBE**

**Q 2 Faire valider par le professeur**

#### **4- Première évolution du diagramme d'états de commande**

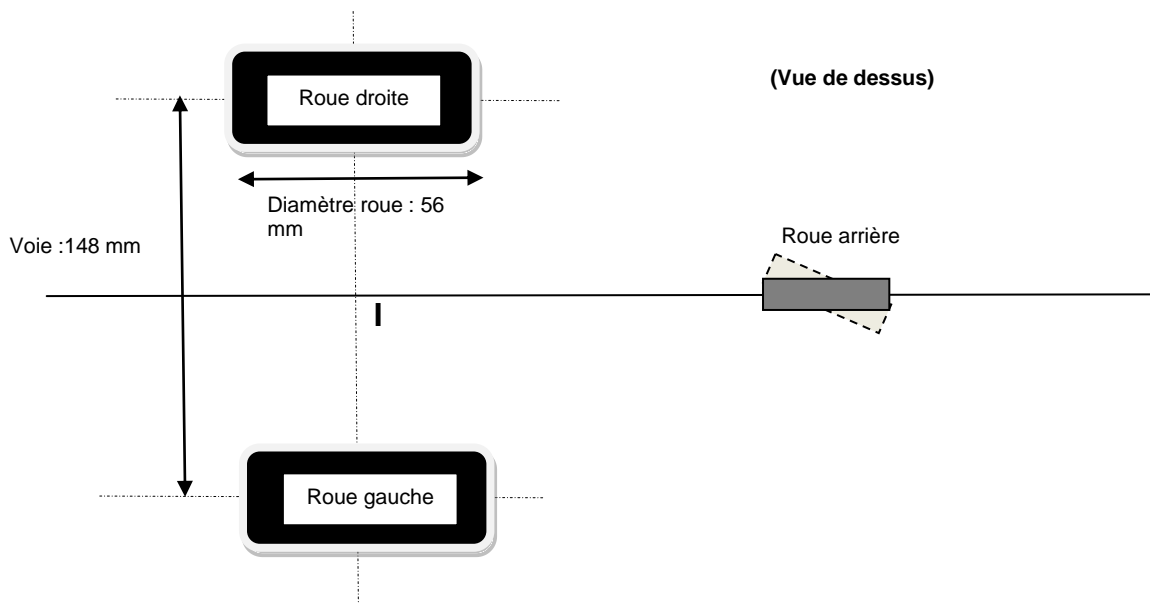
Pour obtenir un déplacement précis du robot, on peut utiliser le capteur de rotation situé sur chaque roue.

Les variables associées sont **Position moteur A**, **Position moteur B** et **Position moteur C**  
 Ces valeurs renvoient l'angle de rotation en degrés de la roue.

Remarques :

- 1- On doit mettre à zéro la valeur de cette variable en utilisant l'ordre : **\_Raz position moteur A\_** (respectivement B, C).
- 2- On teste la valeur du capteur de la manière suivante : **Position moteur A<<180** ou **Position moteur A>>70** (180 ou 70 sont donnés à titre d'exemple)


Les dimensions du robot sont les suivantes :



**Q 3** Quel est l'angle de la roue droite et de la roue gauche à imposer pour un déplacement de 10 cm ? Pour une rotation d'un quart de tour autour du point I ?

**Q 4** Modifier sur le document réponse le diagramme d'états précédent qui permet d'avoir un déplacement précis de 10cm lorsque le robot recule et une rotation précise de 90° (effectuer les modifications en couleurs sur le document-réponse).

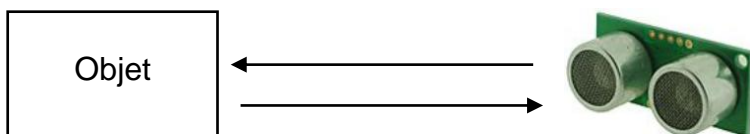
Vérifier le diagramme d'états et le télécharger sur le robot :

Cliquer sur l'icône « GO ! » :  et attendre quelques secondes le chargement :  
**BIEN TENIR LE ROBOT PENDANT LE CHARGEMENT POUR EVITER QU'IL NE TOMBE**

**Q 5** Faire valider par le professeur

## 5- Deuxième évolution du diagramme d'états de commande

### Le capteur à ultrasons :



**Principe :** On mesure le temps mis par une onde ultrasonore, réfléchiée par un obstacle, entre l'émetteur et le récepteur.  
 On peut ainsi mesurer la distance entre le capteur et l'obstacle.  
 Vitesse de propagation :  $340 \text{ m.s}^{-1}$

Exemple d'application : radar de recul sur les automobiles :



Sur le robot Lego, le capteur à ultrasons est associé à la variable **Valeur numérique capteur 4** : cette valeur correspond à la distance à l'obstacle en cm.

Dans le langage diagramme d'états, pour utiliser cette valeur on peut placer des tests qui se traduisent en valeur booléenne (vrai ou faux).

Exemple : **Valeur numérique capteur 4 > 25** est vrai si la distance mesurée par le capteur à ultrasons est supérieure à 25cm.

**On désire obtenir le fonctionnement suivant :**

- Le robot avance en ligne droite à vitesse réduite (50%)
- Si un obstacle est détecté avec le capteur à contact le robot doit s'arrêter, reculer de 10cm et s'arrêter,
- Si un obstacle est détecté à moins de 25 cm par le capteur à ultrasons le robot doit ralentir (25% de vitesses)
- Si un obstacle est détecté à moins de 15 cm par le capteur à ultrasons le robot doit s'arrêter,

**Q 6 Proposer un diagramme d'états correspondant à ce fonctionnement**

**Q 7 Faire valider par le professeur**

### **6- Troisième évolution du diagramme d'états de commande**

Le capteur à ultrasons est placé sur le moteur C, au sommet du robot : on peut donc orienter son axe de mesure.

Si l'on a besoin de stocker la valeur numérique d'une mesure, on peut utiliser l'une des deux variables numériques **Valeur1** ou **Valeur2**.

Pour affecter une valeur à ces deux variables, on est obligé, dans l'action d'un état, d'utiliser la syntaxe suivante :

**{%MW212:=%MW206 ;}**

Avec :



- %MW212 : associée à Valeur1
- %MW213 : associée à Valeur2
- %MW206 : associée à Valeur numérique capteur 4

**On désire obtenir le fonctionnement suivant :**

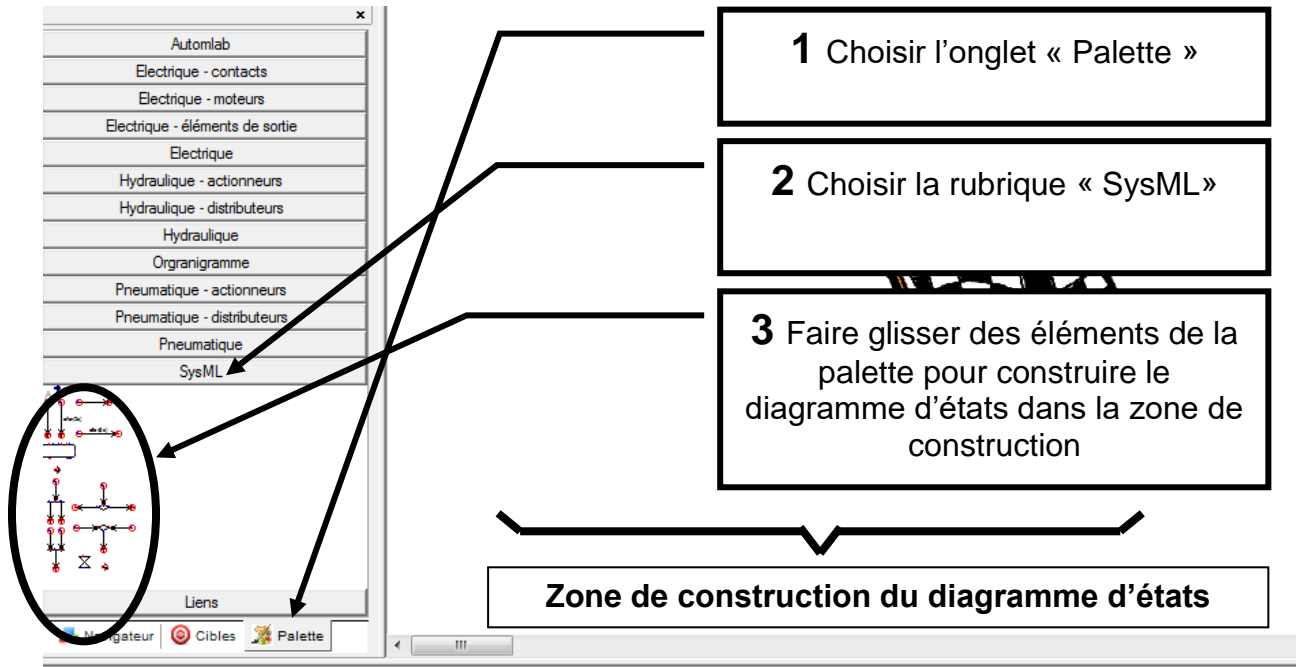
- **En plus du fonctionnement précédent**, si le robot est à l'arrêt à cause d'un obstacle (détecté par contact ou à distance), il doit mesurer la distance qui le sépare d'un autre obstacle à sa droite et à sa gauche :
  - Le robot doit choisir le chemin qui lui permet de parcourir le plus de trajet
  - Le changement de direction se fait par angle droit

**Q 8 Proposer un diagramme d'états correspondant à ce fonctionnement : apporter les modifications en couleurs au diagramme d'états précédent sur le document-réponse**

**Q 9 Faire valider par le professeur**

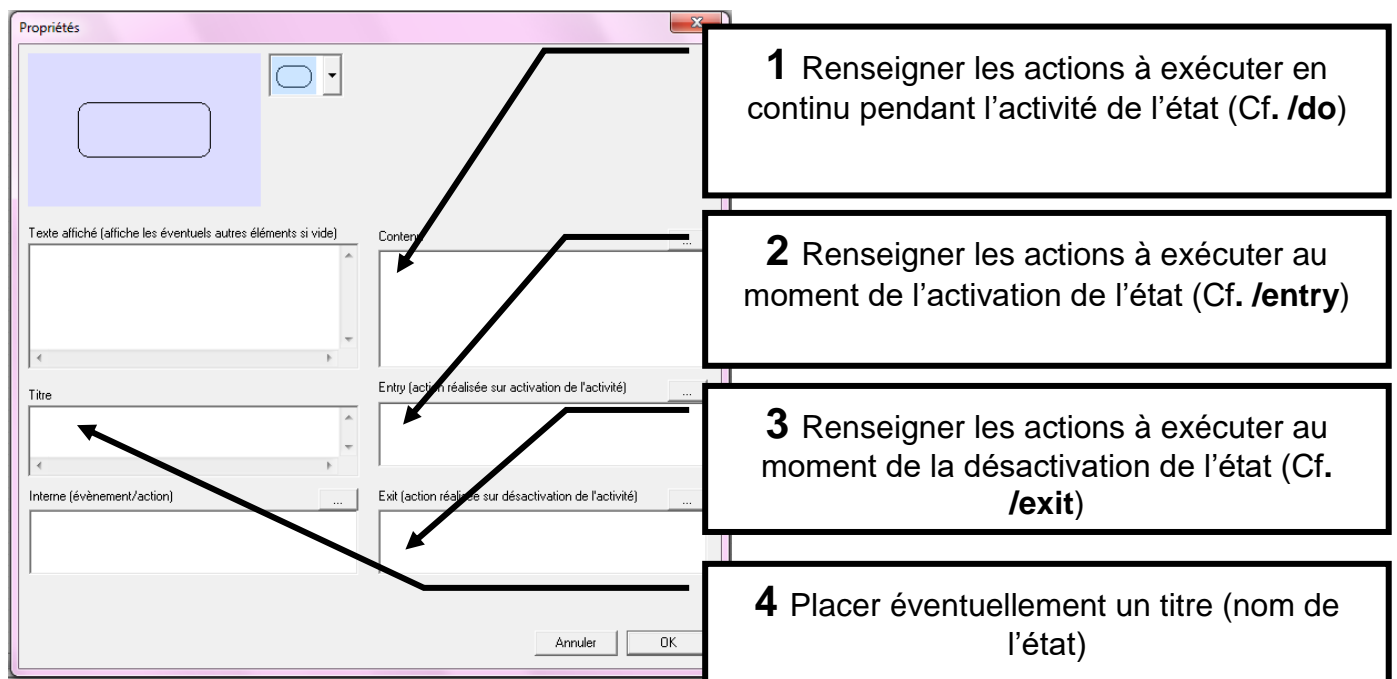
# CREER ET MODIFIER UN DIAGRAMME D'ÉTATS SUR ATOMGEN

Les éléments de construction du diagramme d'états se placent à la souris sur la zone de construction.



Puis remplir :

- **les actions associées aux états:** double-cliquer sur l'état, la fenêtre suivante apparaît :

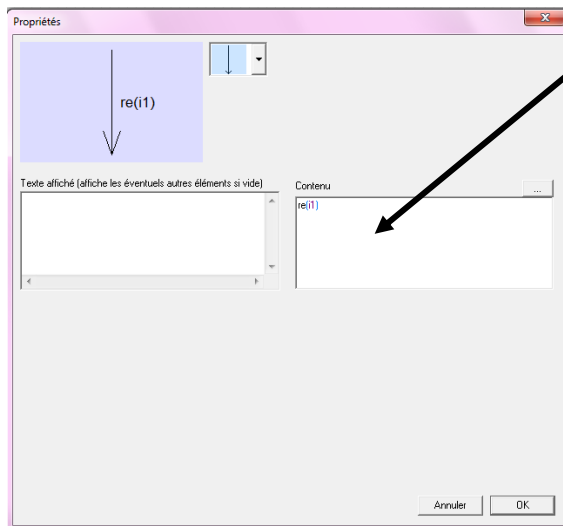


## Toutes les actions se terminent par le signe « ; »

Types d'actions :

- **Affectation :**  
mise à « un » :  $O_i:=1;$   
mise à « zéro » :  $O_i:=0;$
- **Actions multiples :** revenir à la ligne :  $\begin{array}{l} 01:=1; \\ 04:=0; \end{array}$
- **Incrémentation / décrémentation :** sur les variables numériques :  $C := C+1;$   
 $C := C-1;$

- **Les événements ou conditions de garde sur les transitions:** cliquer-droit sur la transition, et en choisissant *propriétés*, la fenêtre suivante apparaît :



Dans « contenu », on renseigne la condition logique pour franchir la transition.

**N.B. :**

le font montant se note « re() » :  $\uparrow a = re(a)$   
le font descendant se note « fe() » :  $\downarrow a = fe(a)$   
le complémentaire se note « / » :  $\bar{a} = /a$

Exemple :

$\bar{a}. \uparrow c + \downarrow b$  se note :  $/a.re(c) + fe(b)$

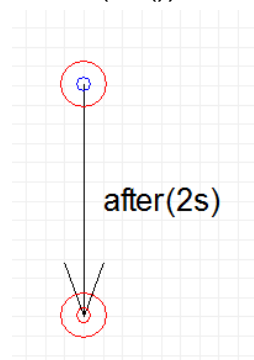
Un test sur une variable numérique s'écrit :  $c < 3$  ou  $c = 4$

Remarque :

- les fonctions logiques sont : ET (.), OU (+), NON (/), front montant ( re()) et front descendant (fe()) .



On peut utiliser des transitions avec temporisation :

pour des durées en dixième de seconde, donner la valeur sans le « s ». Exemple : « after(30) » est identique à « after(3s) ».



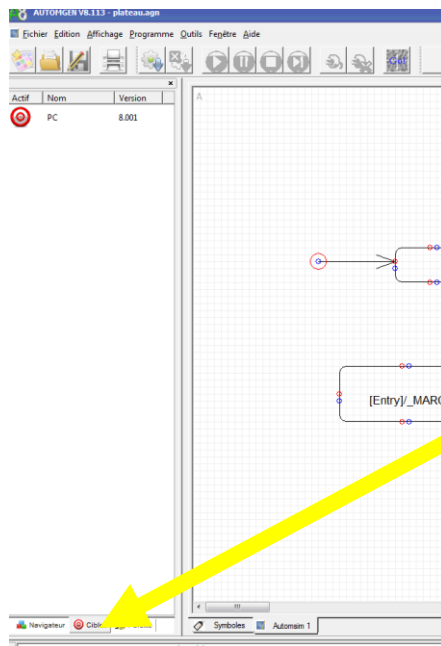
**Rappel :** on peut utiliser l'activité d'un état comme variable : exemple :  $[in\ ATTENTE] = 1$  si l'état intitulé « ATTENTE » est actif, et vaut 0 sinon. Bien noter le « in » qui signifie que la variable est « interne » à la partie commande. Le nom de l'état doit être renseigné dans la case « titre » de celui-ci.

## EDITION :


On peut sélectionner à la souris des zones entières, puis copier (  ), coller (  ), effacer (touche Supr du clavier)

## CHARGER UN DIAGRAMME D'ÉTATS SUR L'AUTOMATE ET L'EXECUTER

Choix de la cible :



Sélectionner l'onglet Cibles  
puis double-cliquer sur la ligne  
RCX/NXT

Pour tester la solution sur le robot : Cliquer sur l'icône « GO ! » :  pour charger le nouveau diagramme d'états dans l'automate et visualiser l'animation à l'écran.

Pour déconnecter l'ordinateur du robot et arrêter la visualisation à l'écran des états actifs :

Cliquer sur l'icône « GO ! » :  .