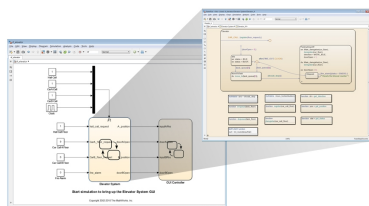
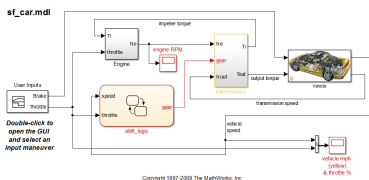
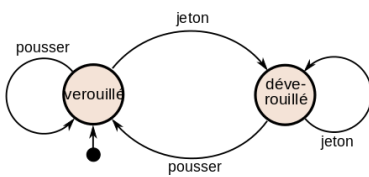


## Chapitre 2

### Systèmes séquentiels

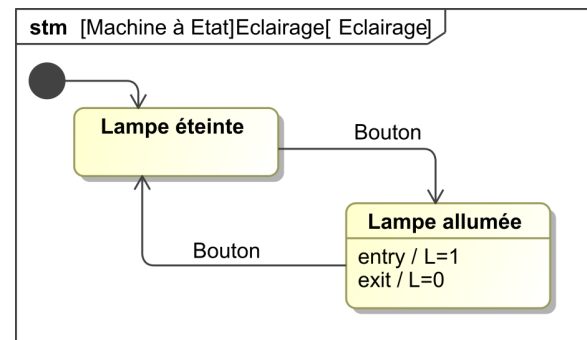
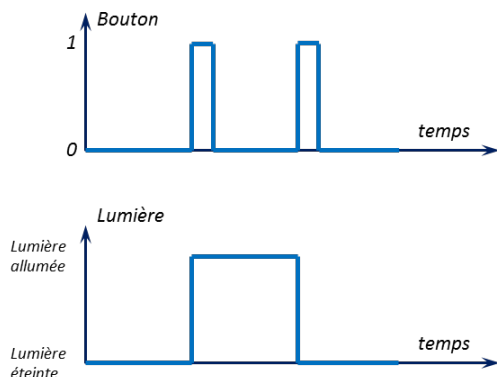


<b>1</b>	<b>La fonction mémoire</b>	<b>2</b>
1.1	Le chronogramme . . . . .	2
1.2	Notion de mémoire . . . . .	2
1.3	Réalisation d'une mémoire . . . . .	3
<b>2</b>	<b>Modélisation du comportement des systèmes séquentiels (1)</b>	<b>3</b>
2.1	Construction et lecture d'un diagramme d'états . . . . .	3
2.2	Utilisation des pseudo-états . . . . .	4
<b>3</b>	<b>Structures algorithmiques</b>	<b>4</b>
3.1	Affectation . . . . .	4
3.2	Groupe d'instructions . . . . .	4
3.3	Fonctions et procédures . . . . .	5
3.4	Structure alternative (conditionnelle) . . . . .	5
3.5	Structure répétitive (itérative) . . . . .	5

## 1 La fonction mémoire

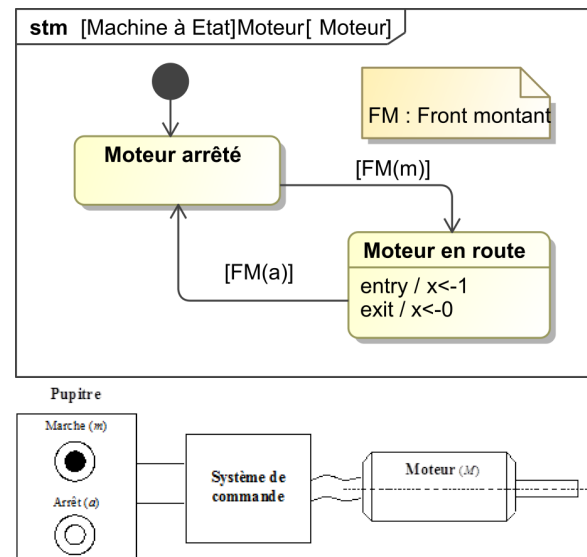
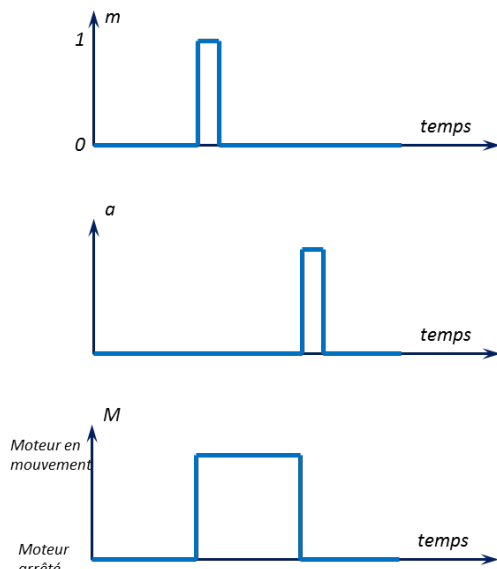
### 1.1 Le chronogramme

Le chronogramme est un outil permettant d'afficher l'état des entrées et des sorties en fonction du temps. Le chronogramme suivant indique le fonctionnement d'une lampe. L'interrupteur «va-et-vient» est relié à un relais.



Ainsi, pour une même combinaison des entrées, on observe des états différents de la sortie.

Le même constat peut être dressé à partir du fonctionnement d'un moteur (*Exemple : fonctionnement du système Doshydro*).



Ici aussi, on remarque qu'une même combinaison des entrées ( $a = m = 0$ ) peut conduire à un état de marche ou d'arrêt du moteur. Une variable interne, ici, une mémoire, permet de mémoriser l'état du moteur.

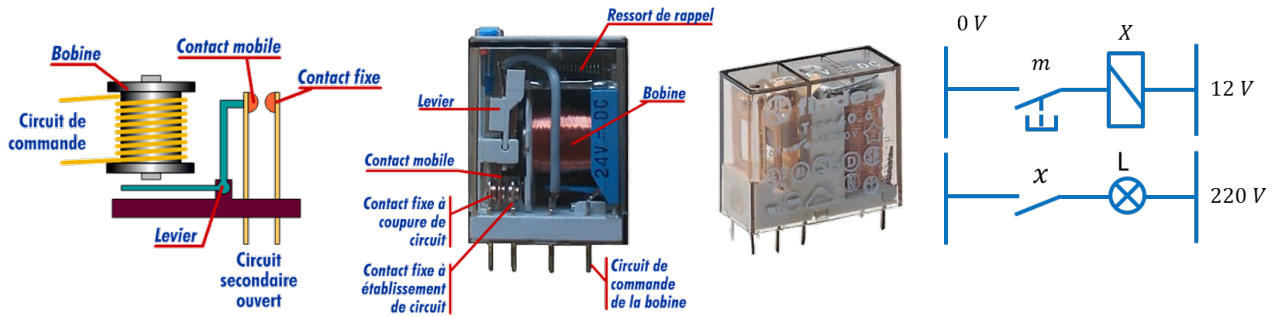
### 1.2 Notion de mémoire

La mémorisation des états est à la base de l'existence des systèmes séquentiels. En notant  $x$  la variable interne d'un système, on lui associe une mémoire.

On peut noter l'existence des mémoires permanentes (qui conservent l'état des variables en cas de coupure d'énergie) et les mémoires volatiles (qui perdent l'état des variables en cas de coupure d'énergie).

Les mémoires peuvent être réalisées de façon optique (blue-ray), électropneumatiques (distributeurs électropneumatiques bistables), magnétiques (disques durs), électriques (grâce à des transistors, mémoire flash), électromagnétique (relais).

Dans les systèmes automatisés industriels, le relais auto maintenu électromagnétique est un élément clef de la mémorisation des états.



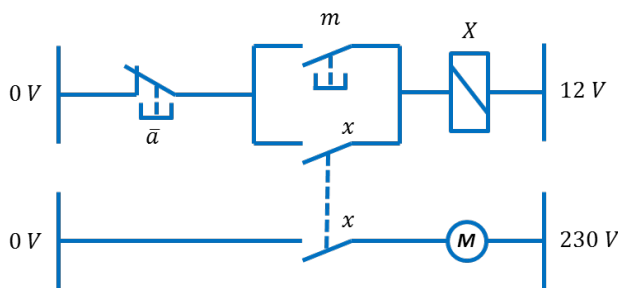
Lorsque l'utilisateur presse le bouton  $m$ , le relais  $X$  est activé. Ainsi, la variable  $x$  est mise à 1 ce qui provoque l'éclairage de la lampe  $L$ . À ce stade, si on relâche l'interrupteur  $m$  le relais  $X$  est désactivé, la variable  $x$  passe à l'état 0 et la lampe s'éteint.

**Objectif** Comment réaliser une mémoire par l'utilisation d'un relais? Que se passe-t-il lors de l'appui simultané sur marche et arrêt.

### 1.3 Réalisation d'une mémoire

#### Exemple : mémoire à effacement prioritaire

Dans ce cas, le moteur s'allume lorsque  $m$  est pressé. Il est éteint lorsque  $a$  est pressé ou lorsque  $m$  et  $a$  sont pressés simultanément.

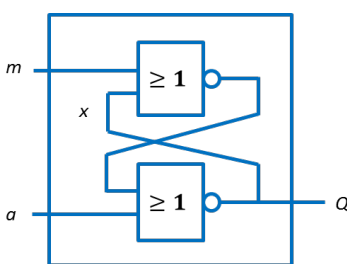


Il est possible de traduire le fonctionnement du moteur par l'équation booléenne suivante :

$$M = \bar{a} \cdot (m + x)$$

Ainsi dans un premier temps, en appuyant sur  $m$  le relais est actionné,  $x$  passe donc à 1 dans le circuit de puissance (allumant ainsi le moteur) **et** dans le circuit de commande. On effectue ainsi un **auto maintien**. On comprend aisément qu'un appui simultané sur  $a$  et  $m$  provoque une désactivation du relais et donc un arrêt du moteur.

Lorsqu'on utilise une technologie électronique, il est courant d'utiliser de réaliser un câblage en utilisant des cellules NOR.



m	a	x	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Il n'est pas directement possible d'écrire directement la table de vérité de la sortie  $Q$  en fonction de  $a$  et de  $b$ . Il faut avoir recours à une variable interne  $x$ .

On retrouve bien l'équation précédente :

$$Q = \bar{m} \cdot \bar{a} \cdot x + \underbrace{m \cdot \bar{a} \cdot \bar{x} + m \cdot \bar{a} \cdot x}_{m \cdot \bar{a}}$$

$$Q = \bar{a} (\bar{m} \cdot x + m) = \bar{a} (x + m)$$

## 2 Modélisation du comportement des systèmes séquentiels (1)

Les outils privilégiés pour décrire le fonctionnement d'un système séquentiel seront les diagramme SysML. Le diagramme de séquence permet de présenter la succession d'étapes liés à un cas d'utilisation d'un point de vue utilisateur. Le diagramme d'état et le diagramme d'activité permettent de décrire l'évolution interne du système.

### 2.1 Construction et lecture d'un diagramme d'états

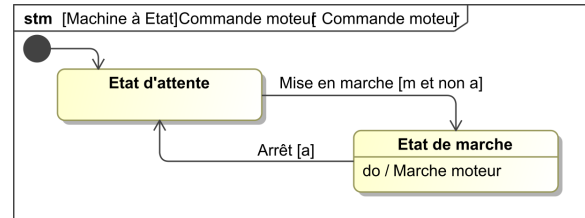
**Méthode** Pour chaque diagramme, il faut définir un état représenté par un nœud. Au cas où aucune activité n'est souhaitée, c'est un état d'attente.

Il faut ensuite identifier les cas d'évolution du système, c'est-à-dire les transitions possibles d'un état à un autre.

On trace alors un lien entre les nœuds correspondants. Chaque transition peut être qualifiée par un événement, une condition de garde et un effet.

### ■ Exemple *Commande tout ou rien d'un moteur*

Deux états internes du système cohabitent. Dans un cas le moteur est en marche, dans l'autre il est à l'arrêt. Les conditions de passage d'un état à un autre se font à l'aide des variables d'entrée arrêt « a » et marche « m ». L'activité « Marche moteur » peut être détaillée dans un diagramme d'activité. Elle comporterait l'action « distribuer l'énergie ». Son exécution ne nécessite pas de temps significatif, elle est associée à un ordre de commande en direction du préactionneur du moteur (un contacteur électrique à arrêt prioritaire).



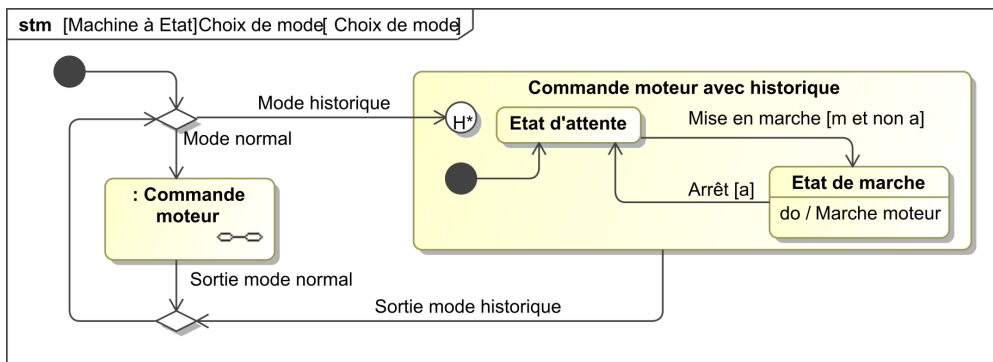
## 2.2 Utilisation des pseudo-états

**Méthode** Les pseudo-états permettent d'utiliser des fonctionnalités avancées sans trop compliquer le diagramme d'état avec le formalisme de base. Il est alors possible de montrer des sélections de séquences d'états (évolutions avec alternatives), des structures parallèles (évolutions simultanées), etc.

### ■ Exemple *Commande tout ou rien (TOR) d'un moteur*

Un diagramme d'états « choix de mode » de niveau hiérarchique supérieur à celui décrit précédemment (« commande moteur »), permet de le démarrer selon deux modes :

- « mode normal » : moteur à l'arrêt dans l'« état d'attente » ;
- « mode historique » : dernier état avant que l'on sorte du diagramme « commande moteur » (« état d'attente » ou « état de marche »).



## 3 Structures algorithmiques

### 3.1 Affectation

L'affectation d'une valeur à une variable peut se faire à l'aide d'une action. Cela ne prend pas de temps significatif.

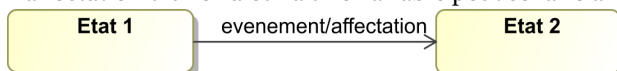


Diagramme d'états – Affectation

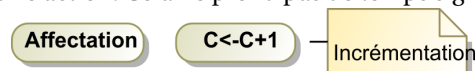
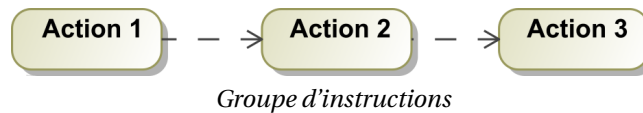


Diagramme d'activité – Affectation

### 3.2 Groupe d'instructions

Un groupe ou un bloc d'instructions peut être une séquence d'un diagramme d'activité. Cela correspond à une succession d'actions et / ou d'activités.



### 3.3 Fonctions et procédures

La décomposition d'un algorithme en fonctions et procédures, permet :

- d'une part, de scinder une problématique générale en plusieurs problématiques élémentaires;
- d'autre part, de pouvoir réutiliser des sous-programmes réalisant des tâches élémentaires.

Une procédure comporte une succession d'instructions mais ne renvoie rien. À la fin de l'exécution d'une fonction, il y a le retour d'une valeur, d'une liste, d'un objet, etc.

### 3.4 Structure alternative (conditionnelle)

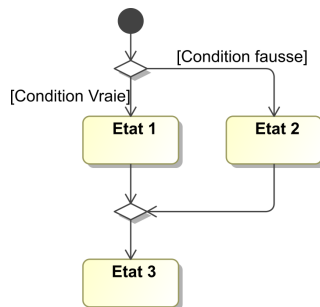


Diagramme d'états – Structure alternative complète

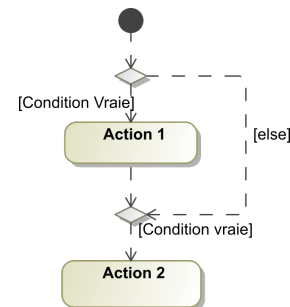


Diagramme d'activité – Structure alternative avec saut

### 3.5 Structure répétitive (itérative)

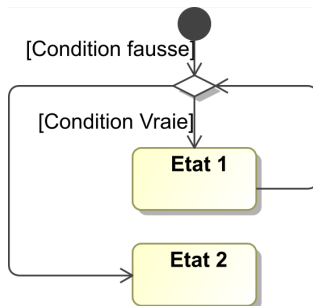


Diagramme d'états – Tant que condition vraie faire ...

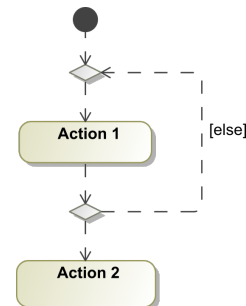
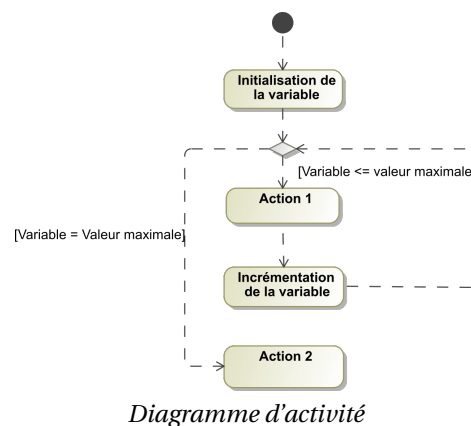
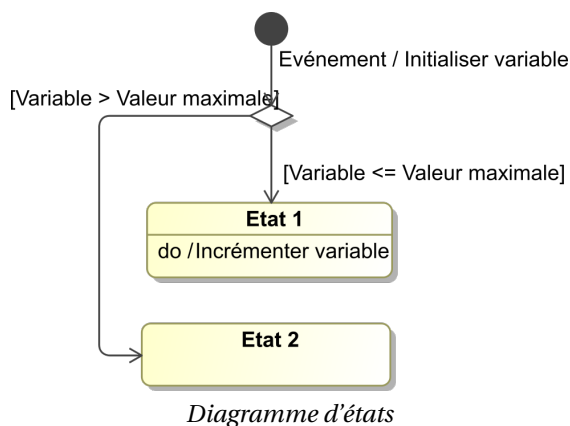


Diagramme d'activité – Répéter... jusqu'à condition vraie

■ **Exemple** Pour variable = valeur initiale, jusqu'à valeur maximale, faire... ■



## Références

[1] Stéphane Genouël, *Systèmes Séquentiels – Fonction mémoire*, Cours de MPSI – PCSI, Lycée Chateaubriand, Rennes, <http://stephane.genouel.free.fr/>.

[2] Patrick Beynet et Al., *Sciences Industrielles pour l'Ingénieur, MPSI – PCSI*, Éditions Ellipses.

- 
- [3] Pierre Debout, *Automatique – Systèmes à événements discrets*, Cours de PCSI.
- [4] Thierry Schanen, *Guide des automatismes 9.1*, Pos Industry, 2009.