

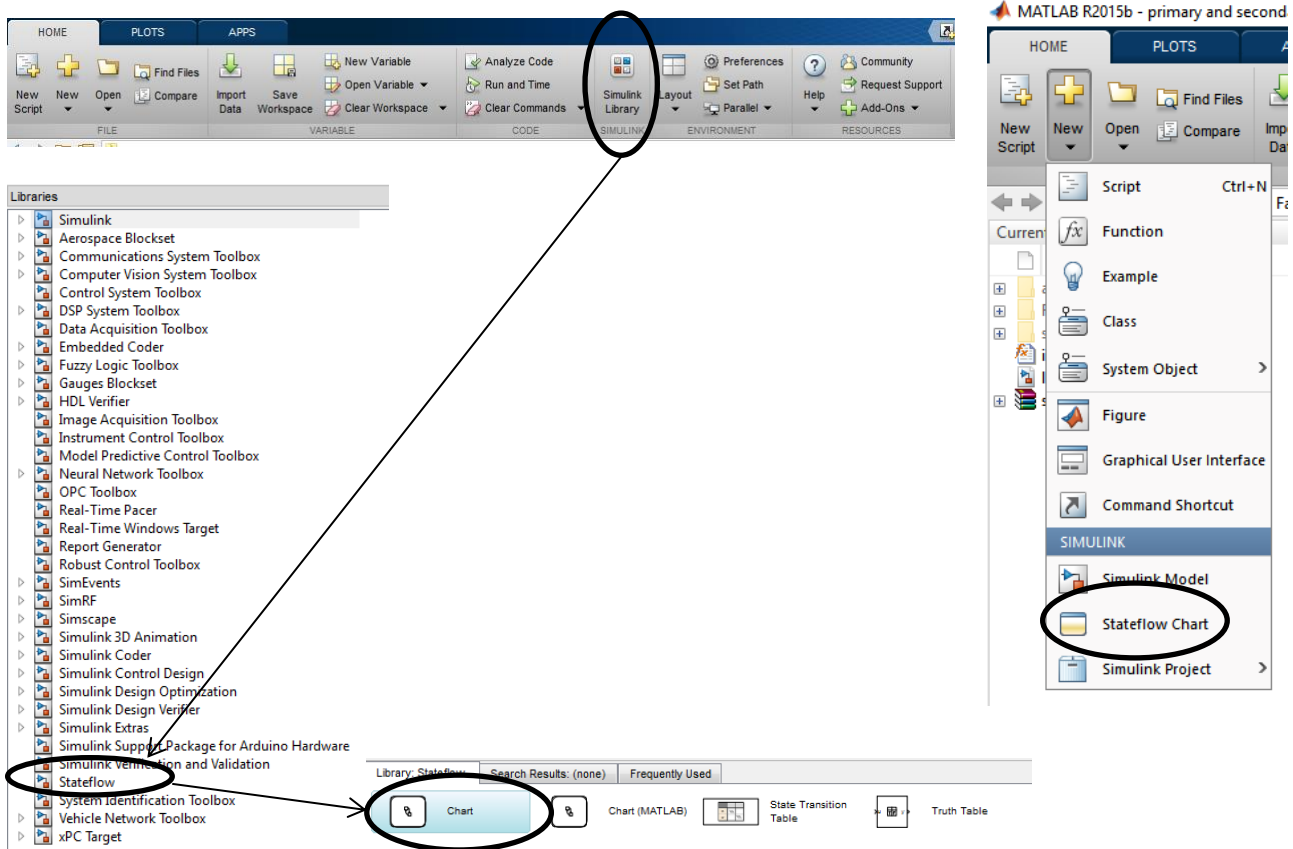
Document largement inspiré du travail de Jean-Christophe MICHEL disponible sur [www.gecif.net](http://www.gecif.net)

**Stateflow** est un outil graphique interactif intégré à **Simulink** pour modéliser et simuler des machines d'état fini, systèmes qui réagissent à des événements, dits systèmes réactifs. Ces systèmes passent d'un état à un autre en réponse à des événements et des conditions.

Une machine à états finis est une machine qui ne fonctionne que dans un nombre finis d'états, ou modes opératoires et permet de modéliser des processus dynamiques.

La description comportementale de tels systèmes est définie dans un diagramme d'état, faisant apparaître les différents états du système ainsi que les transitions permettant de passer d'un état à l'autre. Le logiciel **Stateflow** permet de dessiner ces diagrammes d'état.

### ouverture



## 1 LE DIAGRAMME D'ETAT

Dans l'arborescence des bibliothèques de **Simulink** le diagramme d'état est disponible à l'emplacement suivant :

**Stateflow > Chart**

Vous devez d'abord ouvrir un Stateflow Chart puis le compléter avec la bibliothèque disponible dans Simulink.

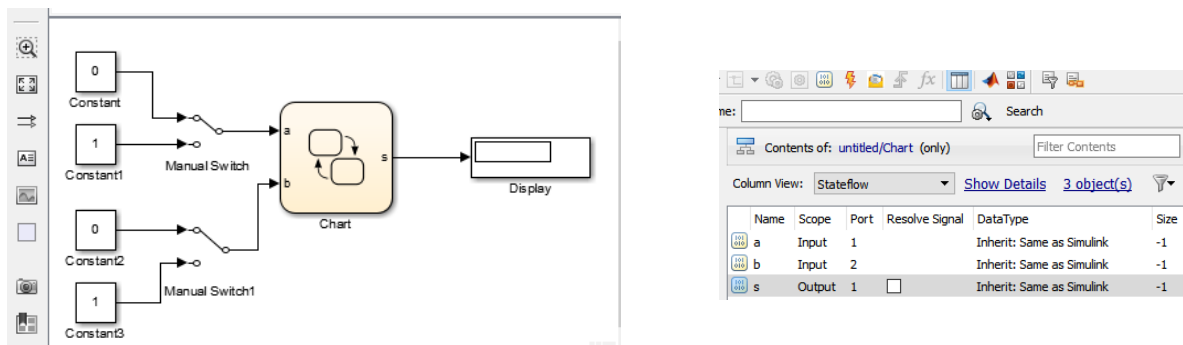


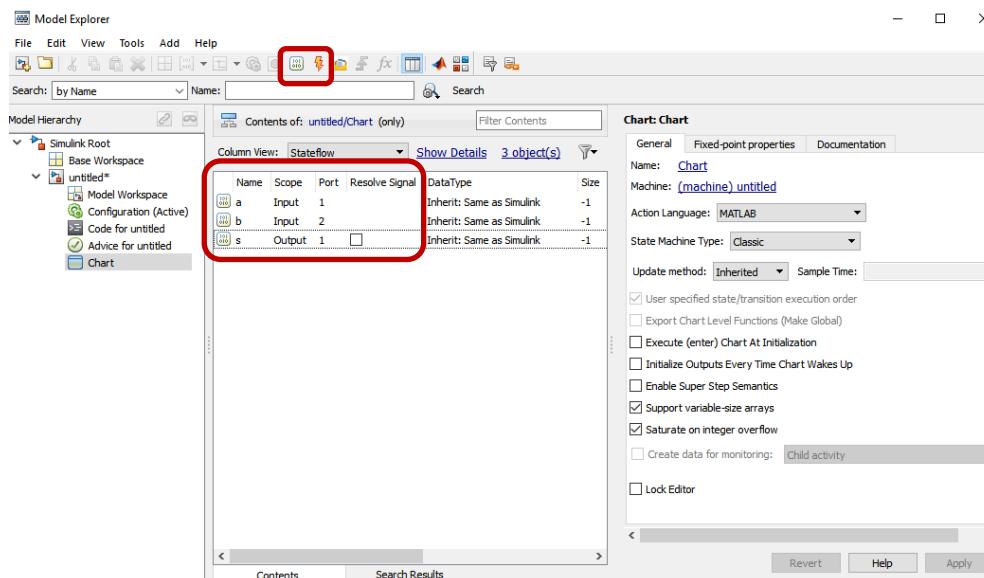
Schéma bloc dans Simulink

Configuration du diagramme d'état :

- cliquer droit sur le bloc **Chart** puis cliquer sur **Explore**
- ajouter une donnée (**Ctrl-D** ou menu **Add+Data**) pour chacune des entrées, les renommer, et les configurer en entrée (**Input** dans la colonne **Scope**)
- ajouter une donnée (**Ctrl-D** ou menu **Add+Data**) pour la sortie, la renommer et la configurer en sortie (**Output** dans la colonne **Scope**)
- pour que toutes les entrées/sortie du diagramme d'état soient booléennes, il faut cocher la case **Enable C-bit operations** (si le système n'est pas uniquement booléen, notamment dans l'utilisation de simulink, ce choix n'est pas toujours possible).



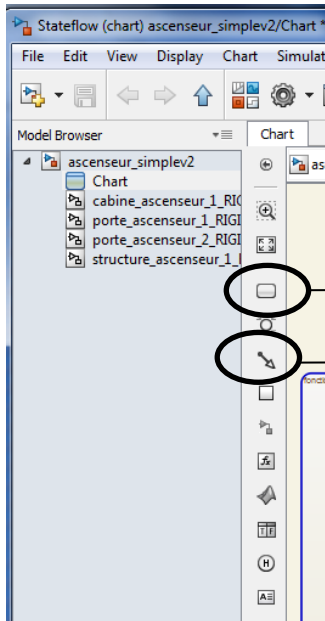
- permet d'ajouter des données (icône de gauche) ou des événements (icône de droite)



Configuration du diagramme d'état dans l'explorateur de modèle

## 2 LES ETATS STATE ET TRANSITIONS (ET/OU EVENEMENTS)

Pour ajouter un état dans **Stateflow**, il faut ouvrir le chart en cliquant droit dessus et sélectionner « open in a new window ».



Un état. Renseigner son nom ou étiquette

Transition par défaut, permet l'entrée dans le diagramme

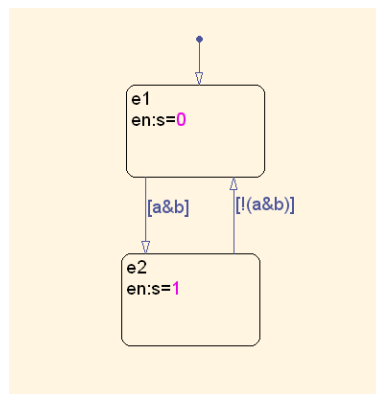


Diagramme d'état d'un porte logique ET dans Stateflow

L'état porte un nom ou une étiquette « state label », en haut à gauche. Les actions associées aux états peuvent avoir lieu :

- A l'activation de l'état : « entry : action ; »
- A la désactivation de l'état : « exit : action ; »
- Pendant l'activité de l'état : « during : action ; »
- « on événement : action ; »

Si plusieurs actions, les séparer par un point-virgule.

Les commentaires suivent le signe %.

Pour ajouter une transition entre deux états dans **Stateflow** il suffit de cliquer sur des bords de chacun des rectangles d'état.

Les transitions sont caractérisées par une étiquette qui décrit les circonstances ou les conditions de passage d'un état à un autre.

Les transitions sont constituées de conditions entre [], de temporisations after(**10,sec**)

La transition réflexive « **self loop transition** » part d'un état ou pseudo état et y revient.

Une transition peut contenir l'état d'un état (très utile) :

- **[in(etat1.etat11)]** signifie que la condition est vraie quand l'état11 appartenant à l'état1 est actif (état1 actif aussi).

Les opérateurs logiques utilisables dans les transitions du diagramme d'état tracé dans **Stateflow** sont les suivants :

opérateur logique	symbole dans Stateflow
<b>ET</b>	<b>&amp;&amp;</b>
<b>OU</b>	<b>  </b>
<b>OU-Exclusif</b>	<b>^</b>
<b>NON</b>	<b>!</b>

Pour le OU logique, la commande clavier est « Alt0124 » ou Alt Gr 6

Pour le front montant « Alt24 » et le front descendant « Alt25 » (non reconnu par matlab). Il faut utiliser **rising**, **falling** ou **either**

### 3 EXEMPLES DE DIAGRAMMES D'ETAT

#### 3.1 Commande de feux tricolores

Le diagramme d'état contient 3 états (nommés ici **e1** **e2** et **e3**) allumant chacun 1 feu et éteignant les 2 autres. Pour passer automatiquement d'un état à l'autre au bout d'un certain temps on utilise la transition **after** :

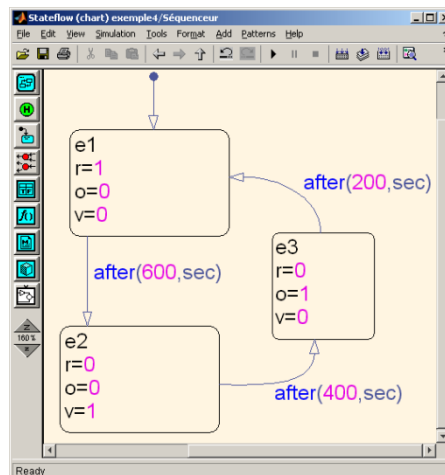


Diagramme d'état séquentiel des feux tricolores dans Stateflow

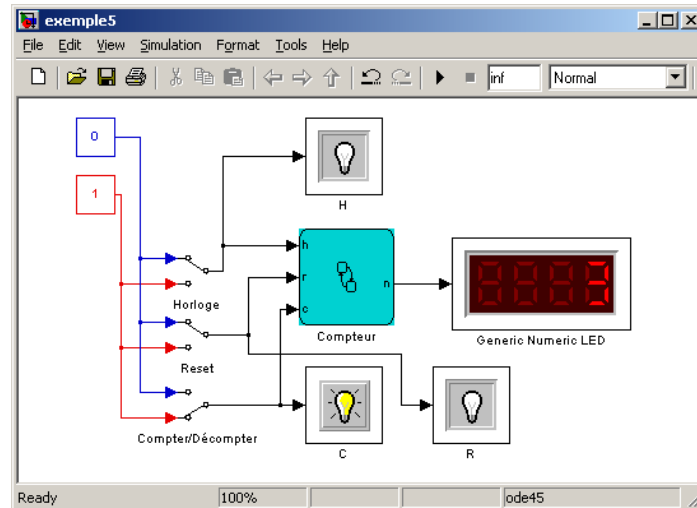
#### 3.2 Réalisation d'un compteur

Grâce au diagramme d'état il est possible de programmer n'importe quel système séquentiel. Voyons un nouvel exemple : un compteur/décompteur.

Notre compteur doit avoir 3 entrées :

- une entrée d'horloge **H** active sur front montant
- une entrée de remise à zéro **R** active au niveau haut
- et une entrée **C** permettant de choisir le sens de comptage (si **C=1** le compteur **compte**, et si **C=0** le compteur **décompte**)

Le schéma bloc contient le diagramme d'état avec ces 3 entrées et 3 interrupteurs permettant d'agir sur ces entrées :



*Schéma bloc du compteur/décompteur dans Simulink*

l'afficheur 7 segments permet de visualiser la valeur numérique **n** en sortie du compteur.

Les ampoules permettent de visualiser facilement l'état logique des entrées.

Dans la configuration du modèle du diagramme d'état, 4 données ont été ajoutées, toutes de type double. Trois d'entre elles sont des entrées (**h**, **r** et **c**), et la quatrième est une sortie (**n**, la valeur numérique en sortie du compteur) :

Name	Scope	Port	Resolve signal	Data type
h	Input	1	<input type="checkbox"/>	double
n	Out...	1	<input type="checkbox"/>	double
r	Input	2	<input type="checkbox"/>	double
c	Input	3	<input type="checkbox"/>	double

Et voici le diagramme d'état décrivant le comportement du compteur en fonction de l'état logique appliqué sur les entrées :

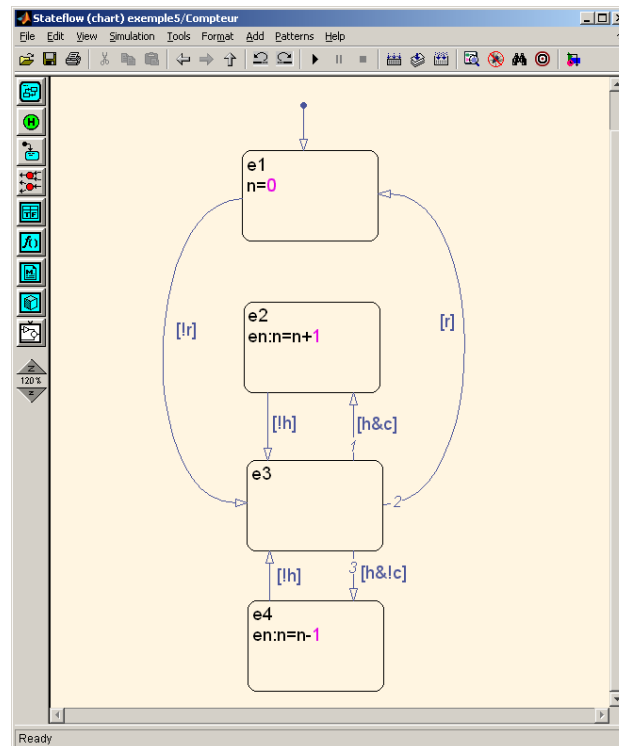


Diagramme d'état du compteur/décompteur dans Stateflow

Explications et commentaires de ce diagramme d'état du compteur :

- le diagramme contient 4 états nommés **e1**, **e2**, **e3** et **e4**
- au démarrage le programme rentre dans le diagramme par l'état **e1** qui remet le compteur à zéro
- dans l'état **e1** : si **r=0** (transition **!r**) on passe dans l'état **e3** qui est l'état d'attente ou rien ne se passe
- dans l'état **e3** le compteur ne fait qu'attendre une action sur ses entrées :
  - si **h=1** et **c=1** (transition **h&c**) on passe dans l'état **e2** pour incrémenter le compteur (action **n=n+1**)
  - si **h=1** et **c=0** (transition **h&!c**) on passe dans l'état **e4** pour décrémenter le compteur (action **n=n-1**)
  - si **r=1** (transition **r**) on passe dans l'état **e1** pour remettre le compteur à zéro
- dans les états **e2** et **e4** :
  - on ne fait évoluer le compteur qu'une seule fois à l'entrée dans l'état (grâce à **en:** qui précède l'action) et non en continu
  - dès que l'horloge **h** repasse à zéro (transition **!h**) on retourne dans l'état d'attente **e3**