

TP N°3

Commande d'une montre digitale en temps réel

I. But de la manipulation

Il s'agit d'effectuer la commande d'une montre digitale à l'aide d'un noyau temps réel. L'objectif de la commande est d'assurer les différentes fonctions d'une montre digitale (affichage, réglage ...).

Cette manipulation a pour but d'illustrer le cours de « système de commande logique » sur la commande d'un système à événement discret. La modélisation du fonctionnement de la montre est réalisée par Statecharts et développée sous Matlab/Simulink avec la boîte à outils Stateflow.

II. Présentation de la maquette à commander



Figure 1: montre digitale

La montre considérée dans ce TP est semblable à celle donnée sur la Figure 1.

Cette montre comporte plusieurs **modes de fonctionnement** (heure, date, alarme, chronomètre) accessibles par les quatre boutons (Bouton_1, Bouton_2, Bouton_3 et Bouton_4).

Le fonctionnement de chaque bouton est le suivant :

- Bouton_4 : il permet d'éclairer l'écran. Il peut être actionné en même temps que n'importe quel autre bouton.
- Bouton_3 : Un appui simple sur ce bouton permet de passer de mode en mode de manière cyclique. Le mode initial est le mode *heure*, le mode suivant est le mode *date*, puis *alarme*, et enfin *chronomètre*.
- Bouton_2 : Ce bouton permet, pour les modes *heure*, *date* et *alarme* de rentrer dans le mode *réglages*. Une fois le mode réglage sélectionné, un appui sur le bouton 2 permet de passer à l'élément suivant à régler. Si le mode *heure* est actif, un appui sur le bouton 2 permet de rentrer dans le mode *réglages*. Il faut alors régler d'abord les secondes. Ensuite, un nouvel appui sur le bouton 2 permettra de régler les minutes, puis les heures, etc de manière cyclique. Si le mode date est actif, le bouton 2 permettra de régler le jour, le mois puis le jour de la semaine.

- Bouton_1 : Lorsque le mode *réglages* est actif, ce bouton permet en général d'incrémenter de 1 l'élément à régler (sauf pour les secondes qui sont réinitialisées sur appui du bouton 1).

L'affichage des informations (voir figure 2) se fait de la manière suivante : les digits principaux (au centre visu 1 à 3) affichent l'élément correspondant au mode de fonctionnement actif. Les digits du haut (visu 6) affichent toujours les deux premières lettres du jour (en anglais). Les digits des secondes en mode *heure* n'affichent rien dans les autres modes, sauf en mode *chronomètre* où ils permettent d'afficher les dixièmes et les centièmes de secondes. Par exemple, si le mode *alarme* est actif, l'heure et les minutes de l'alarme seront affichées sur les digits centraux.

Initialement, le jour de la semaine est MONday, la date 01.01 et l'heure 00.00.00.

Pour la gestion de l'alarme plusieurs choix sont disponibles. Il existe en effet 3 modes d'alarme. *Carillon* : un bip est émis par la montre à chaque passage d'une heure ; *Alarme* : la montre sonne et le voyant buzzer s'allume à l'heure réglée par l'utilisateur. Le dernier mode combine *Carillon et Alarme*. Pour régler ces modes, il faut appuyer simultanément sur les boutons 4 et 3 (en commençant par le bouton 4). L'accès à ces trois modes se fait de manière cyclique : *Bip horaire*, *Bip horaire et Alarme*, puis *Alarme*, puis aucun des deux modes. Un appui sur un bouton quelconque permet d'arrêter la sonnerie de l'alarme.

Deux voyants permettent d'indiquer le mode d'alarme actif.

La montre dispose également d'une fonction **minuteur**. Dans un premier temps nous considérerons un minuteur simplifié. Quand le mode *minuteur* est activé, l'affichage est par défaut fixé à 1 minute. L'appui sur le bouton 2 permet de démarrer et d'arrêter le décompte du minuteur. Quand le minuteur arrive à 0, alors l'alarme sonne. La remise à zéro s'effectue avec le bouton 1.

La maquette de la montre commandée en TP est donnée figure 2.

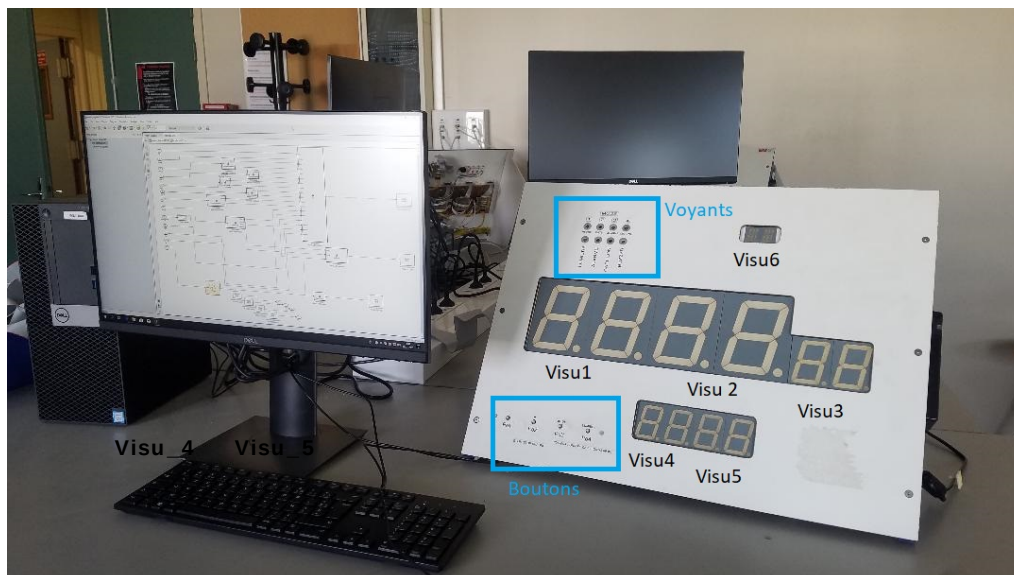


Figure 2. La maquette de la montre

La face avant de la maquette de la montre (voir figure 2) comporte cinq zones de visualisation composées chacune de deux afficheurs sept segments et une sixième zone d'affichage composée elle de deux afficheurs alpha-numériques. La maquette comprend également quatre boutons correspondant à Bouton_1, Bouton_2, Bouton_3 et Bouton_4.

Huit voyants lumineux permettent de visualiser l'activation de la lumière de la montre, de l'alarme, du carillon, du buzzer ainsi que l'activation du mode de fonctionnement.

Les grandeurs **Visu_x** correspondent aux nombres qui vont être affichés sur les afficheurs de la maquette et qui varient selon le mode de fonctionnement actif. La grandeur à affichée sur chaque afficheur est traitée en amont de la maquette pour être décomposée en chiffre des dizaines et chiffres des unités, puis convertie pour l'affichage sept segments.

III. Présentation de l'environnement de développement

La commande de la montre est réalisée à l'aide d'un PC sous l'environnement Simulink Desktop Real Time.

Il s'agit d'un système de prototypage rapide dans Matlab. Le prototypage rapide permet de passer de la phase essai/simulation à la phase mise en œuvre/expérimentation très rapidement et très facilement en s'affranchissant de l'étape de codage. L'ordinateur qui sert au développement sert également à exécuter un code en temps réel : il a deux rôles bien distincts.

- **En tant que PC hôte**, il sert d'unité de développement. A partir de Matlab/Simulink, il est possible de générer directement un code de commande (commande de la montre dans notre cas).
- **En tant que PC cible**, il est orienté temps réel. Il permet d'exécuter le code engendré par l'unité de développement. Il est équipé de cartes (dans notre cas une carte National Instrument PCI_6503 24 bits avec convertisseurs analogique/numérique et convertisseurs numérique/analogique) qui lui permettent de commander le processus.

La Figure 3 présente les entités manipulées en TP : le PC de développement avec Matlab/Simulink et la boîte à outils Stateflow, le PC cible et le système à contrôler i.e. la maquette de la montre.

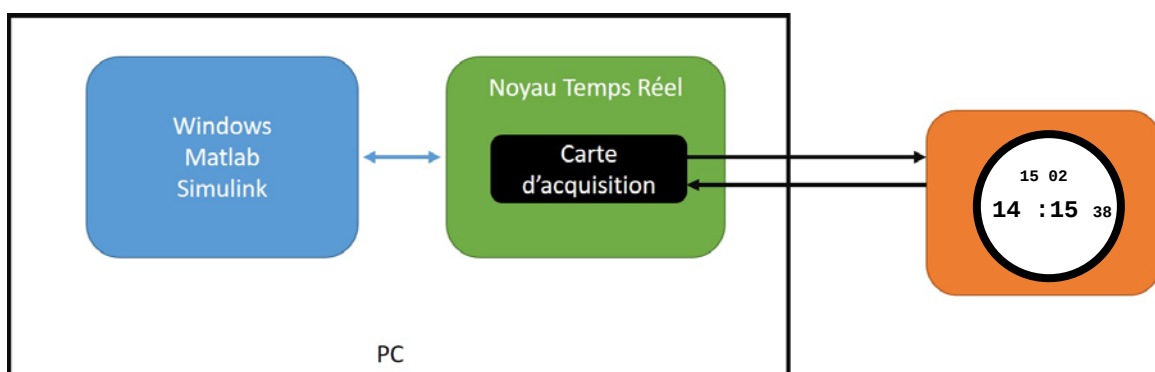


Figure 3. Commande déportée sous Simulink De

Le tableau ci-dessous résume les différentes entrées/ sorties de la carte d'acquisitions.

Les noms indiqués sont ceux des signaux utilisés dans le source Matlab.

Nom	Type	Signification
Bouton_1	entrée	Bouton de la montre
Bouton_2	entrée	Bouton de la montre
Bouton_3	entrée	Bouton de la montre
Bouton_4	entrée	Bouton de la montre
visu_1	sortie	Voir Figure 2
visu_2	sortie	Voir Figure 2
visu_3	sortie	Voir Figure 2
visu_4	sortie	Voir Figure 2
visu_5	sortie	Voir Figure 2
visu_6	sortie	Voir Figure 2
light	sortie	Lumière
al	sortie	Alarme
hor	sortie	Carillon
Bip	sortie	Buzzer
mode	sortie	Mode de fonctionnement

IV. Manipulation

Il s'agit dans un premier temps de tester certaines fonctionnalités de base de la montre déjà modélisées. Dans un deuxième temps il s'agit de rajouter d'autres fonctionnalités.

Vous trouverez sur moodle le statechart à compléter qui vous aidera pour la **préparation**.

L'annexe intitulée « Stateflow et Matlab » vous aidera pour la syntaxe Stateflow.

Attention : tout le chemin dans lequel vous téléchargez le dossier de travail ne doit pas contenir d'espace

Partie 1 :

Vous allez tout d'abord tester le fonctionnement d'une montre simplifiée.

Cette montre simplifiée ne dispose ni d'éclairage, ni de fonction permettant la gestion de l'alarme, ni de mode *chronomètre*.

Lancez Matlab par le menu **Démarrer**. En cliquant sur le bouton « Browse for folder » indiqué sur la Figure 2, positionnez-vous dans le dossier où vous avez enregistré les fichiers fournis.

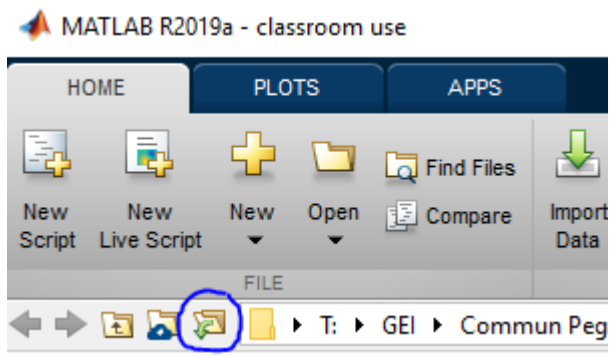


Figure 2 : Browse for folder

Sur le côté gauche de la fenêtre, on retrouve le fichier **montre_digitale.slx** : modèle Simulink contenant le statechart à compléter.

Attention : Il ne faut pas renommer ce fichier ou les blocs.

Ouvrez le fichier *montre_digitale.slx* en double-cliquant dessus.

Ce fichier (voir Figure 3) est un modèle Simulink de la commande de la montre digitale dans lequel apparaissent

- la carte d'acquisition Entrées / Sorties PCI_6503 permettant la connexion avec la maquette (bloc Interface).
- un bloc Statechart correspondant au modèle de la commande de la montre.
- différents blocs Simulink permettant d'assurer les conversions nécessaires à l'affichage des données numériques représentant les heures, jours etc.... vers les afficheurs 7 segments, les afficheurs alpha-numériques et les voyants lumineux.

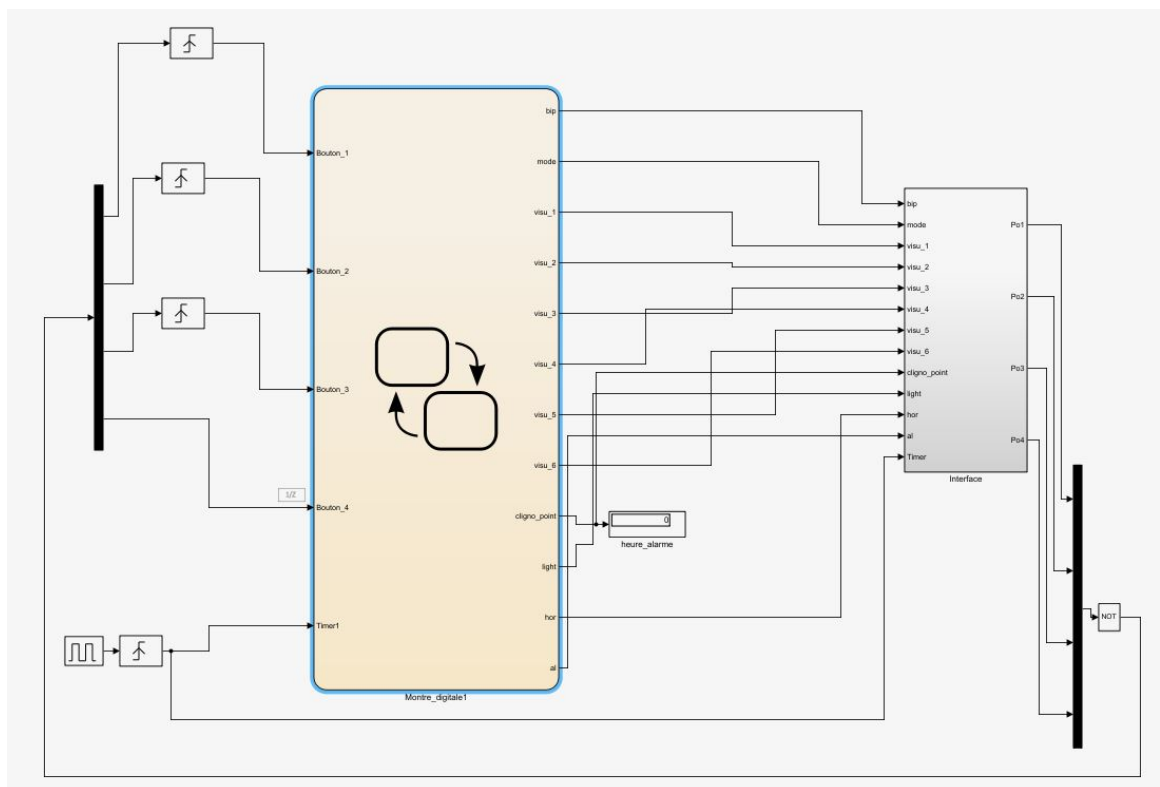


Figure 3 : Simulink principal la montre digitale

Vous allez travailler au niveau du bloc Statechart. Ouvrez ce bloc par un double clic souris.

Ce bloc ouvre une fenêtre Stateflow contenant 8 macro-états ET :

- Gest_horloge : gère le fonctionnement de l'horloge, c'est-à-dire l'incrémentation des centièmes, secondes, minutes et heures.
- Gest alarme : gère l'activation du buzzer de l'alarme.
- Gest_date : gère le fonctionnement de la date, c'est-à-dire l'incrémentation du jour, mois et année (sans prise en compte des années bissextiles et de mois de 30 jours).
- Mode_alarme : gère l'alarme c'est-à-dire le passage entre les trois modes d'alarme.
- Gest_bip_passe_heure : gère l'activation du carillon de l'alarme.
- MODES1 : ce macro état permet d'effectuer les différents réglages de la montre et contient 2 sous états OU : TIME qui permet le réglage de l'heure et DATE qui permet le réglage de la date.
- GestionAffichage : qui gère l'affectation des données à afficher sur les différents afficheurs de la maquette en fonction du mode dans lequel se trouve la montre.

Lancement d'une première exécution

Dans l'onglet Desktop Real Time, cliquer sur Run in Real Time. (Figure 4) Régler le temps d'exécution à une valeur infinie (inf). Pour lancer l'exécution, cliquer sur la flèche verte du bandeau de commande.

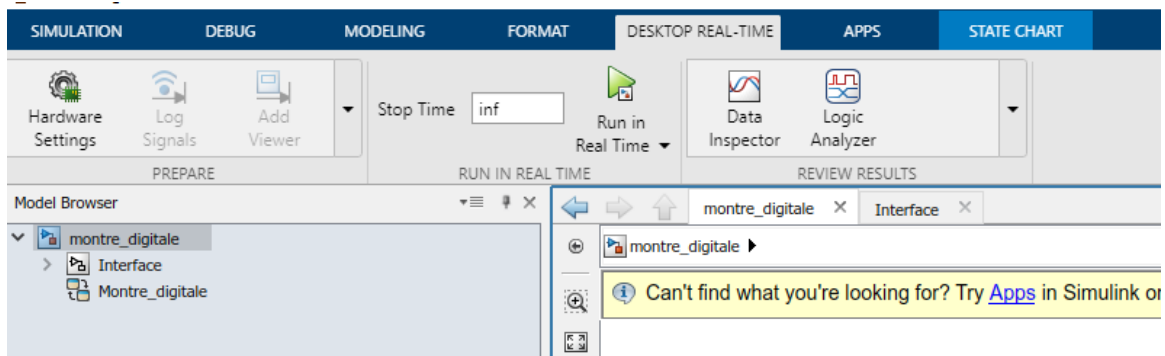


Figure 4 : bandeau de commande

Le lancement de l'exécution peut prendre plusieurs dizaines de secondes.

Testez le programme en place en utilisant les quatre boutons poussoirs et observer l'évolution du système. De manière générale, si un dysfonctionnement est constaté, il est possible d'en trouver la cause en observant l'évolution du Stateflow au cours de l'exécution du programme. Dans la fenêtre Stateflow, les états actifs sont indiqués en bleu. A cause de la communication entre le système et le PC, un certain délai entre un appui sur un bouton et le changement d'état sur le statechart peut être observé. De plus, il est conseillé de zoomer sur la partie intéressante du statechart pour le débogage de votre programme de commande.

Arrêtez l'exécution en appuyant sur le bouton « carré » du bandeau de commande.

Partie 2 : suppose un travail de modélisation à faire en préparation

Mise en place de l'éclairage :

Modifier le fichier précédent contenant le modèle Stateflow de la montre simplifiée de manière à rajouter la fonction d'éclairage. Cette nouvelle fonction sera mise en place via un nouveau bloc Statechart inséré dans le fichier précédent et connecté si besoin à certains blocs déjà existant. Les entrées/sorties du fichier devront si besoin être mises à jour.

La fonction d'éclairage est activée par l'appui sur le bouton 4. Le mode éclairage est un macro-état OU à deux sous-états. Un sous-état « lumière éteinte » et un sous-état « lumière allumée ». Le passage d'un état à l'autre se fait par appui sur le bouton 4. La lumière reste activée tant que le bouton 4 est appuyé. Au départ la lumière est éteinte.

Tester le fonctionnement de votre commande.

Mise en place du réglage de l'alarme :

Modifier le fichier précédent contenant le modèle Stateflow de la montre simplifiée de manière à rajouter la fonction de gestion de l'alarme. Cette nouvelle fonction sera mise en place via un nouveau bloc Statechart inséré dans le fichier précédent et connecté si besoin à certains blocs déjà existant.

Cette fonction gère le réglage des minutes et des heures de l'alarme. Les secondes ne seront pas réglées. L'activation du mode de réglage de l'alarme ainsi que le réglage en lui-même se font selon le principe général décrit partie II.

Tester le fonctionnement de votre commande en réglant une heure pour l'alarme et en vérifiant que la montre sonne.

Mise en place du minuteur version 1 :

Modifier le fichier précédent contenant le modèle Stateflow de la montre simplifiée de manière à rajouter la fonction minuteur (on utilisera les variables du chronomètre). Cette nouvelle fonction sera mise en place via un nouveau bloc Statechart inséré dans le fichier précédent et connecté si besoin à certains blocs déjà existant.

L'activation du mode minuteur ainsi que le fonctionnement du minuteur version_1 se font selon le principe général décrit partie II. C'est un minuteur qui décompte à la seconde.

La mise en place du minuteur peut être réalisée en s'inspirant du bloc *Gest_horloge* qui assure la gestion de l'heure.

Testez la fonction minuteur.

Mise en place du minuteur version 2 :

Modifier le fichier précédent pour améliorer la fonction minuteur de telle sorte qu'il soit possible de régler le nombre de minutes et de secondes à décompter dans un premier temps.

Dans un deuxième temps, on veut pouvoir sortir de la fonction minuteur sans pour autant que le minuteur s'arrête.

Mise en place d'un chronomètre version 1 :

Modifier le fichier précédent contenant le modèle Stateflow de la montre simplifiée de manière à rajouter une fonction chronomètre. Cette nouvelle fonction sera mise en place via un nouveau bloc Statechart inséré dans le fichier précédent et connecté si besoin à certains blocs déjà existant.

L'activation du mode chronomètre ainsi que le fonctionnement du chronomètre version_1 se font selon le principe suivant. Une fois le mode chronomètre activé, l'appui sur le bouton 2 permet de démarrer et de stopper le chronomètre. La remise à zéro s'effectue avec le bouton 1. C'est un chronomètre qui compte au centième.

La mise en place du chronomètre peut être réalisée en s'inspirant du bloc *Gest_horloge* qui assure la gestion de l'heure.

Testez la fonction chronomètre.

Mise en place du chronomètre version 2 :

Modifier le fichier précédent pour améliorer la fonction chronomètre de telle sorte que l'arrêt du chronomètre par le Bouton 2 fige l'affichage sans pour autant stopper le comptage.

IV. Références

- *"Statecharts : a visual formalism for complex systems " D. Harel, 1987, Journal of Science of Computer Programming, vol.8 n°1 pp 231-274*
- *" On visual Formalism " D. Harel, 1985, Communication of the ACM, vol. 31 n° 51*
- *Stateflow and Stateflow Coder User's Guide*

V. Préparation attendue

Statecharts :

- De l'éclairage
- Réglage de l'alarme
- minuteur
- chronomètres

Annexe 6

Stateflow et Matlab



Documentation Matlab disponible ici :

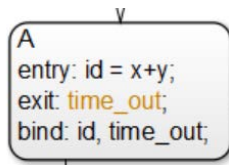
<https://fr.mathworks.com/help/stateflow/index.html>

Stateflow est un outil graphique interactif intégré à **Simulink** pour modéliser et simuler des systèmes avec un formalisme semblable à des Statecharts.

Syntaxe des états

<https://fr.mathworks.com/help/stateflow/syntax-for-states-and-transitions.html>

Les états ont le schéma général suivant :



name

entry:entry action1;action2,action3

during:during actions

exit:exit actions

Si on ne spécifie pas le type d'action explicitement, le traitement par défaut combine les types entry et during.

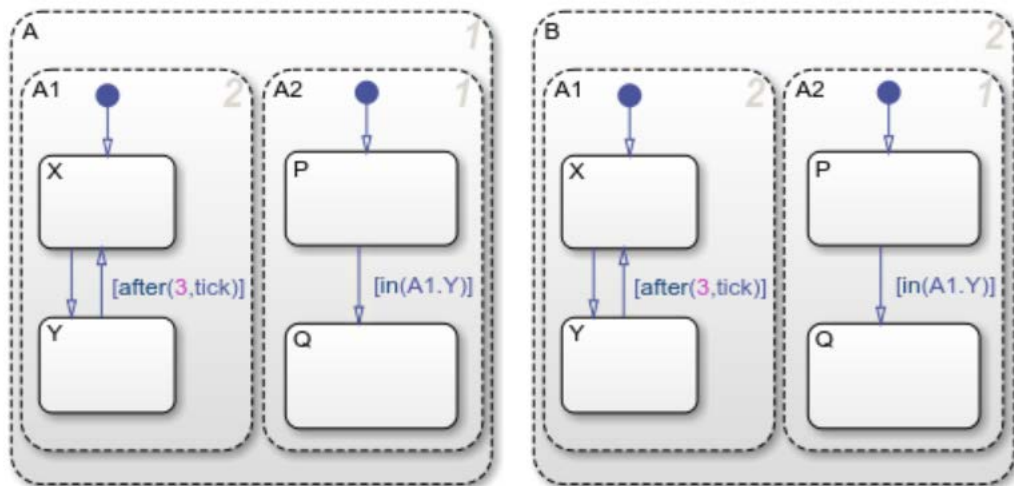
types	description
entry	Exécute l'action quand l'état devient actif
during	Exécute l'action quand l'état est actif et qu'un événement survient
exit	Exécute l'action quand l'état est actif et qu'une transition de sortie de l'état survient

Syntaxe des transitions :

Les opérateurs logiques utilisables dans les transitions sont les suivants :

Opérateur logique	Symbole dans Stateflow
ET	&
OU	
NON	!

Pour passer automatiquement d'un état à un autre au bout d'un certain temps, on utilise le mot clef **after**. Pour tester le fait d'être dans un état, on utilise le mot clef **in**.



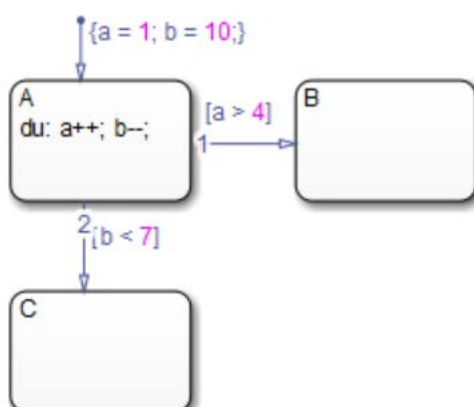
- In the state A, the condition `in(A1.Y)` checks the activity of state A.A1.Y.
- In the state B, the condition `in(A1.Y)` checks the activity of state B.A1.Y.

Si on veut tester le fait d'être dans un état précis du statechart, on doit utiliser toute sa hiérarchie pour le décrire. Exemple B.A1.Y pour décrire l'état Y du sous-état A1 de l'état B.

Spécifier une décomposition dans un état :

On peut définir un macro-état ET (AND decomposition) pour spécifier que les sous-états seront en parallèle, ou bien un macro-état OU (OR decomposition) pour spécifier que les sous-états seront en séquentiel. Pour changer la décomposition d'un état, sélectionner l'état, clic-droit pour faire apparaître le menu **Decomposition**, et sélectionner **OR (exclusive)** ou **AND (Parallel)** dans le menu. Les sous états en OU exclusif auront les bords en trait plein, et les sous-états en parallèle seront en trait pointillé.

Gérer les transitions en conflit



Deux transitions potentiellement en conflit seront numérotées et prises en compte dans l'ordre. Toutes les transitions partant d'un même état source sont automatiquement numérotées dans l'ordre de création. Pour changer cet ordre, clic droit sur la transition, placer le curseur sur **Execution order** et sélectionner l'ordre dans lequel vous voulez exécuter la transition.