Mode opératoire Simulink et Stateflow

Démarrer le logiciel Matlab puis cliquer sur Simulink :



L'objectif de ce tutoriel est de créer une lampe commandée par deux boutons à la manière d'un va et vient dans une maison.

Solution n°1 : la solution câblée

Pour importer un composant, il suffit d'écrire son nom sur le fond blanc et de sélectionner dans la liste qui apparaît. On relit ensuite les composants en « tirant » des fils entre chaque :

Remarque : le composant Ou exclusif se trouve en tapant XOR



Régler le temps de simulation sur inf (infini) puis cliquer sur RUN :



Observer le comportement des switch lorsque l'on fait un double clic dessus et observer également le résultat dans le display.

L'interface étant austère, ajouter les deux types d'éléments suivants : rocker switch et lamp

Rocker Switch				Constant:Valu On	
switch du dessus, un maillon de chaine bleu	1	-		Off Rocker Switch	
apparaît, cliquer sur	Consta		Connect		
constant value. Ils sont		•	Constant:\	/alue	
désormais reliés					Quevolucif

virtuellement : l'état de la constante changera en fonction de l'état du rocker switch.

Modifier le schéma comme suit :



Pour mémoire, les boutons permettant de faire une interface sont :

<i>fx</i> >>	g		
Callback Button	Check Box	Circular Gauge	Combo Box
MA	42	42 1	
Dashboard Scope	Display	Edit	Gauge
(\bigcirc	
Half Gauge	Horizontal Gauge	Knob	Lamp
		-lm-	R
Linear Gauge	MultiStateImage	Push Button	Quarter Gauge
0 0 0 0 0	θ	\bigotimes	
Radio Button	Rocker Switch	Rotary Switch	Slider
	0		
Slider Switch	loggle Switch	Vertical Gauge	

Enfin connecter la lampe au Ou exclusif :

 	-			Display
Ou exclu			Connect	
	۰	t	Ou exclusif:1	
	\bigcirc	t	Constant1:1	
	\bigcirc	t	Constant:1	
				Lamp

Faire le test pour s'assurer que le comportement est le même que précédemment.

Remarque : un double clic sur la lampe permet de changer la couleur en fonction de l'état de la porte Ou exclusif.

Solution n°2 : La solution programmée

Constant: Valu lo or Bouton A lo Constant: lo Constant: lo Display lo Di

On ne conserve que ce schéma :

On choisit (en faisant glisser) chart dans stateflow depuis library Browser :



On reconnecte les constant et le display sur le chart (voir ci-contre).

On reconnecte également la lampe sur le display (voir ci-dessous).





Il faut faire comprendre à Chart la nature de ce qui rentre en lui et ce qui en sort, on clique sur model explorer depuis l'onglet modeling :

SIMULATION DEBUG	MODELING	On développe ensuite le fichier en
O Find ▼ Model Image: Compare Advisor ▼ Image: Environment ▼ Image: Model Explorer Image: Compare	Model Data Editor	Cours puis on clique sur chart.
File Edit View Tools Add Help		
Model Hierarchy 🖉 🗟 🚼 Contents of: untitled/Chart (o	nly) Filter Contents	Chart: Chart
Column View: Stateflow	Show Details 3 object(s) F	General Fixed-point properties Documentation Name: Card Machine: (mathine) untitled Action Language: MATLAB State Machine Type: Cassic Update method: Inherited Sample Time: 1 Update method: Inherited Sample Time: 1 Update method: Inherited Sample Time: 1 User-specified state/transition execution order Eport chart level functions Execute (enter) chart at initialization Initialize outputs every time chart wakes up Initialize outputs every time chart wakes up Support variable-size arrays Support variable-size arrays Generate preprocessor conditionals Create output for monitoring: Child activity Apply Apply
Contents Sear	ch Results	

Changer maintenant le nom des variables d'entrée et de sortie comme suit :

	Name	Scope	Port	Resolve Signal	DataType	Size	Init
101	bouton_A	Input	1		boolean	-1	
101 010	bouton_B	Input	2		boolean	-1	
101 010	lampe	Output	1		double	-1	

Le type de variable est très important, ici on déclare que les entrées ne peuvent prendre que deux états (0 ou 1), on choisira pour la sortie une valeur numérique « double » par exemple).

Faire maintenant un double clic sur le chart pour rentrer dans la zone de rédaction du programme et visionner cette vidéo : <u>https://www.youtube.com/watch?v=XpQ2osUE4v4&ab_channel=MATLAB</u>

Faire ce schéma des états à l'issue de la vidéo :



Compléter ensuite les transitions comme suit :



La transition bouton_A==0&&bouton_B==1 se lit : on passe de l'étape 1 à l'étape 2 si l'état du bouton A est à 0 ET si l'état du bouton B est à 1

Pour mémoire :

!=	signifie	différent de	=	signifie	égal à uniquement dans les cases d'état
!	signifie	contraire de	&&	signifie	ET
==	signifie	égal à uniquement dans les transitions	11	signifie	OU
>, <, >= ou <=	signifie	la même chose qu'en mathématiques			