

Portail automatique

Le portail automatique que vous avez devant vous peut fonctionner suivant différents modes selon la position des interrupteurs TCA, IBL et FCH.

Voir le dossier technique pour la mise en route.

Objectifs du T.P.

Le but de ce T.P. est :

- de découvrir la modélisation du comportement d'un système par diagramme d'états.
- d'améliorer la modélisation proposée.
- d'implanter un modèle valide sur le microcontrôleur d'un prototype.

Comportement du système

Vous allez observer le comportement réel du portail dans deux cas d'utilisation

1. faire fonctionner pas à pas.

Positionner correctement les interrupteurs TCA, IBL et FCH pour que le portail soit en mode "pas à pas" (voir dossier technique).

Faire fonctionner le portail et compléter, à partir de vos observations sur le système réel, le chronogramme du document réponse.

2. ouvrir fermer automatiquement.

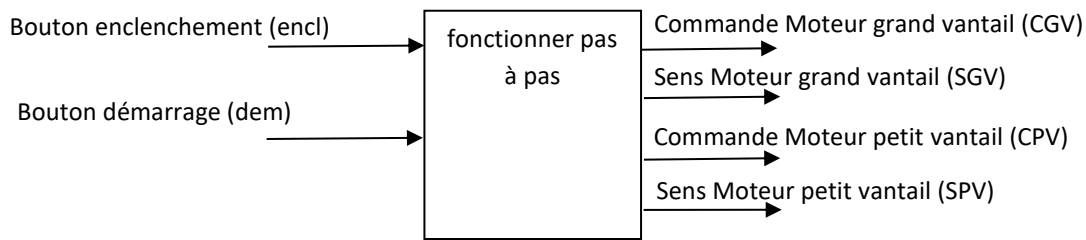
Positionner correctement les interrupteurs TCA, IBL et FCH pour que le portail soit en mode "automatique" (voir dossier technique).

Faire fonctionner le portail et compléter, à partir de vos observations sur le système réel, le chronogramme du document réponse.

Fonctionnement "pas à pas"

Vous allez simuler le comportement d'une partie du système à l'aide d'un diagramme d'état.

1. Modélisation.



On vous propose deux modélisations du comportement du système pour le cas d'utilisation "fonctionner pas à pas" dans les fichiers (portail1.slx) et (portail1bis.slx).

Quelles différences observez-vous entre ces deux modèles, notamment sur la nature des entrées « encl » et « dem » ? (voir dossier ressources)

A l'aide du dossier ressources, lancer une simulation et observer l'évolution du diagramme d'état pour chaque modèle.

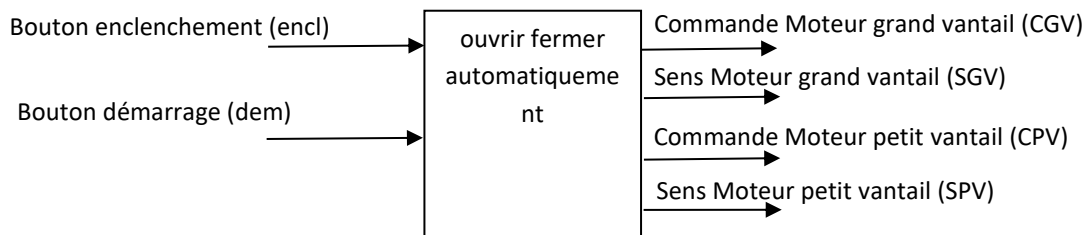
2. Analyse des résultats.

Comparer le chronogramme obtenu par simulation avec le chronogramme observé sur le système réel. Expliquer pourquoi l'un des deux modèles ne donne pas satisfaction.

Fonctionnement "automatique"

Vous allez modéliser le comportement d'une partie du système par un diagramme d'état.

1. Modélisation.



Identifier les entrées et les sorties intervenants dans ce cas d'utilisation et définir leur nature (événement ou condition).

Identifier les différents états du système pour ce cas d'utilisation..

Identifier les transitions permettant le passage d'un état à un autre.

A l'aide du dossier ressources, compléter le diagramme d'état fourni (portail2.slx) modélisant le comportement du système pour le cas d'utilisation "ouvrir fermer automatiquement".

Lancer une simulation.

2. Analyse des résultats.

Comparer le chronogramme obtenu par simulation avec le chronogramme observé sur le système réel. Valider ou non votre modèle.

3. Amélioration du modèle.

Le modèle précédent ne prend pas en compte le fait que le petit vantail doit se fermer après le grand. Pour cela, la fermeture du petit vantail démarrera 2 secondes après celle du grand vantail.

Compléter en conséquence le diagramme d'état fourni (portail3.slx). (Voir dossier ressources sur les états composites)

Lancer une simulation.

4. Analyse des résultats.

Comparer le chronogramme obtenu par simulation avec le chronogramme observé sur le système réel. Valider votre modèle.

Prototype

Vous allez implanter votre modèle dans le microcontrôleur d'un prototype, et valider ainsi le prototype.

On propose de réaliser un prototype du portail sur la base d'une carte microcontrôleur Arduino UNO. Les connexions sur la carte sont déjà réalisées de la manière suivante :

Entrées		Sorties	
Bouton Poussoir enclenchement (encl)	PIN 5	Commande Moteur GV (CGV)	PIN 3
Bouton Poussoir démarrage (dem)	PIN 6	Sens Moteur GV (SGV)	PIN 12
		Commande Moteur PV (CPV)	PIN 11
		Sens Moteur PV (SPV)	PIN 13

A l'aide du dossier ressources, implanter le modèle pour le cas d'utilisation "ouvrir fermer automatiquement". Vérifier le bon fonctionnement du prototype.

Sécurité

Vous allez améliorer le modèle précédent en tenant compte des exigences de sécurité.

1. Modélisation.

A partir du diagramme de séquence fourni pour le cas d'utilisation " Tenir compte d'obstacle", identifier les entrées et les sorties intervenants pour ce cas d'utilisation.

A l'aide du dossier ressources, compléter votre modèle précédent (portail3.slx) en rajoutant les entrées et les sorties nécessaires.

2. diagramme d'états.

Compléter le diagramme d'états précédent, en tenant compte des exigences de sécurité décrites dans le diagramme de séquences.

Simuler son fonctionnement, afin de valider vos choix.

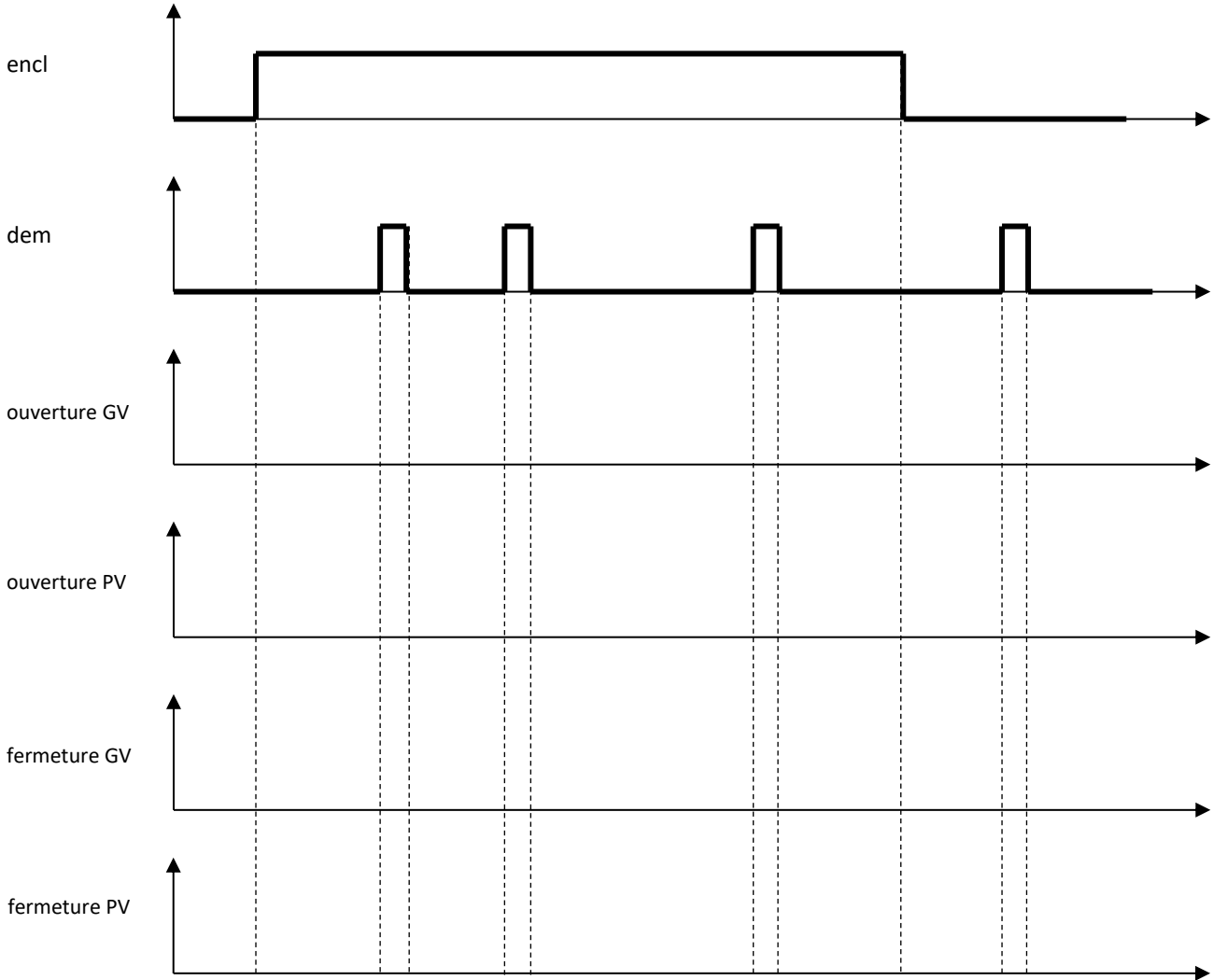
3. Prototype.

A l'aide du dossier ressources, implanter votre modèle sur la carte microcontrôleur Arduino UNO.

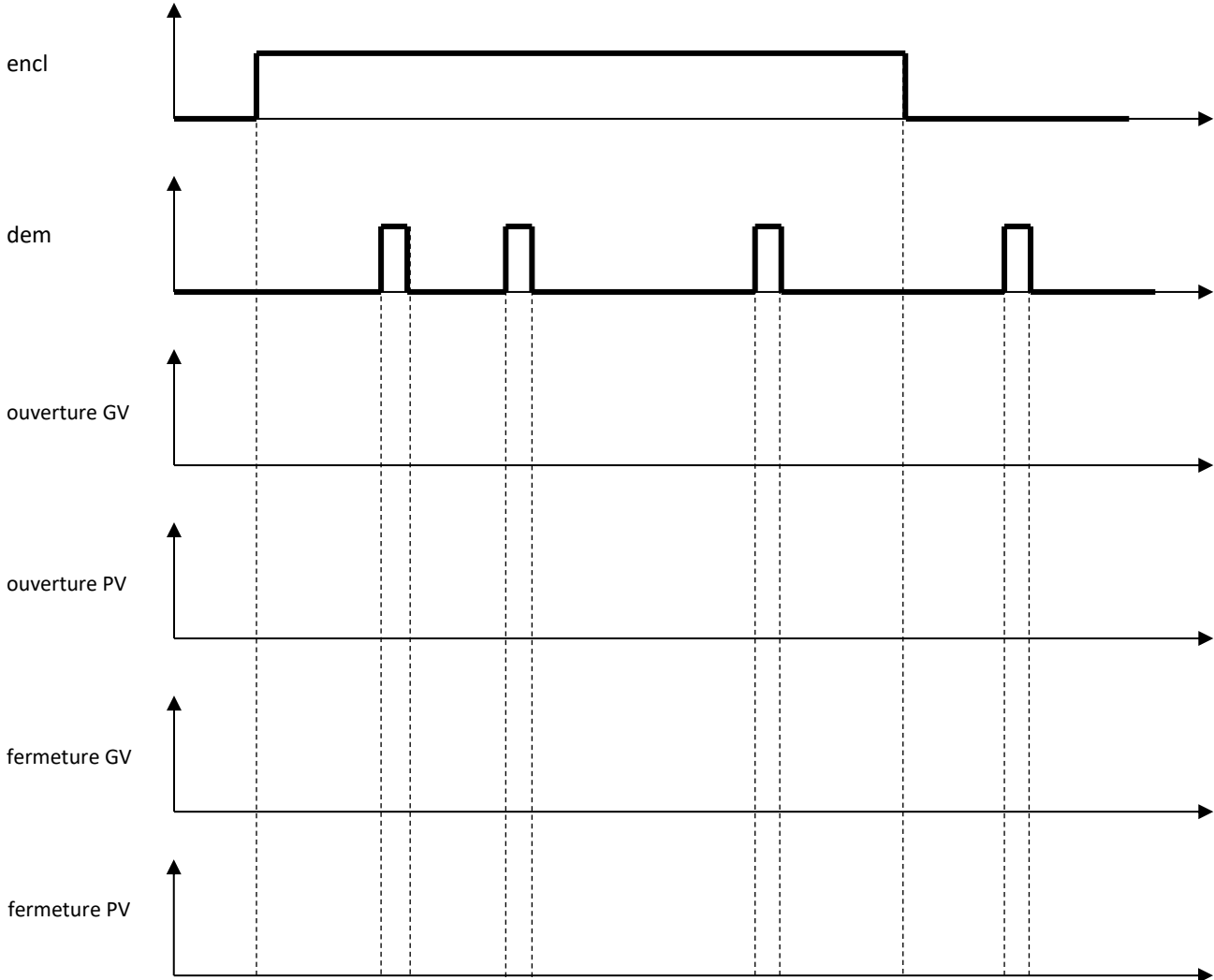
Vérifier le bon fonctionnement du prototype.

Entrées		Sorties	
Bouton Poussoir enclenchement (encl)	PIN 5	Commande Moteur GV (CGV)	PIN 3
Bouton Poussoir démarrage (dem)	PIN 6	Sens Moteur GV (SGV)	PIN 12
Cellule photo électrique (phot)	PIN	Commande Moteur PV (CPV)	PIN 11
		Sens Moteur PV (SPV)	PIN 13

Mode de fonctionnement "pas à pas" observé :



Mode de fonctionnement "automatique" observé :

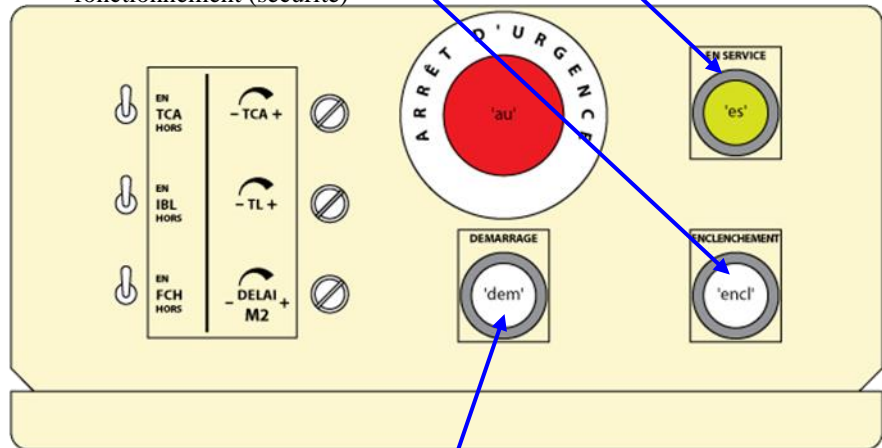


DOSSIER TECHNIQUE

FONCTIONNEMENT DU PORTAIL

Mode pas à pas : IBL, TCA et FCH en position "hors".

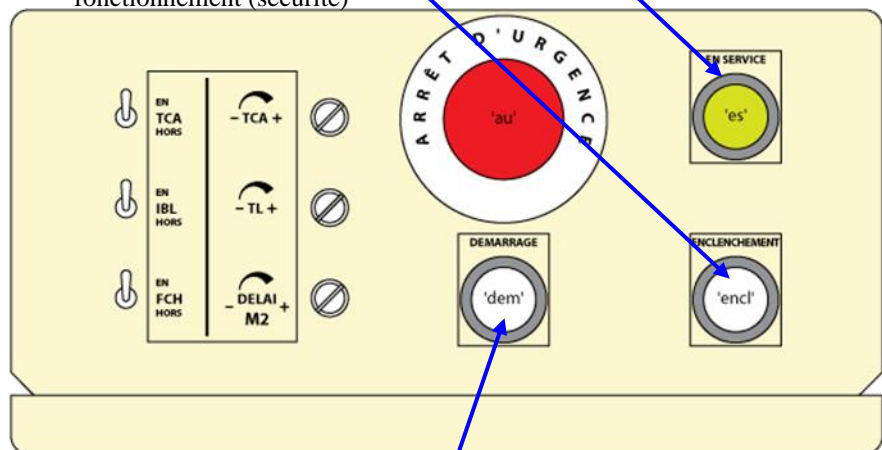
- 1/ mettre en service
- 2/ garder le bouton "enclenchement" enfoncé durant tout le fonctionnement (sécurité)



- 3/ à la première impulsion sur le bouton "démarrage", le portail s'ouvre ; à la deuxième, il s'arrête ; à la troisième il se ferme...

Mode automatique : IBL en position "en", TCA et FCH en position "hors".

- 1/ mettre en service
- 2/ garder le bouton "enclenchement" enfoncé durant tout le fonctionnement (sécurité)



- 3/ à la première impulsion sur le bouton "démarrage", le portail s'ouvre, puis s'arrête et se referme.

Diagramme des cas d'utilisation :

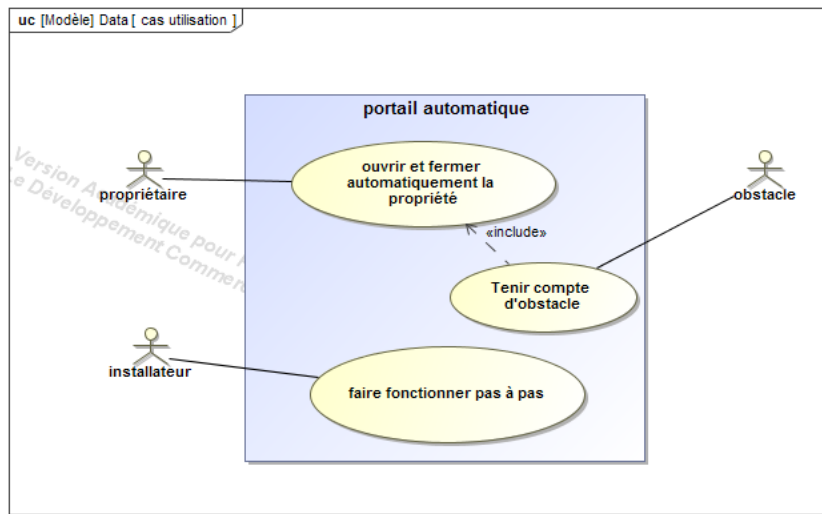
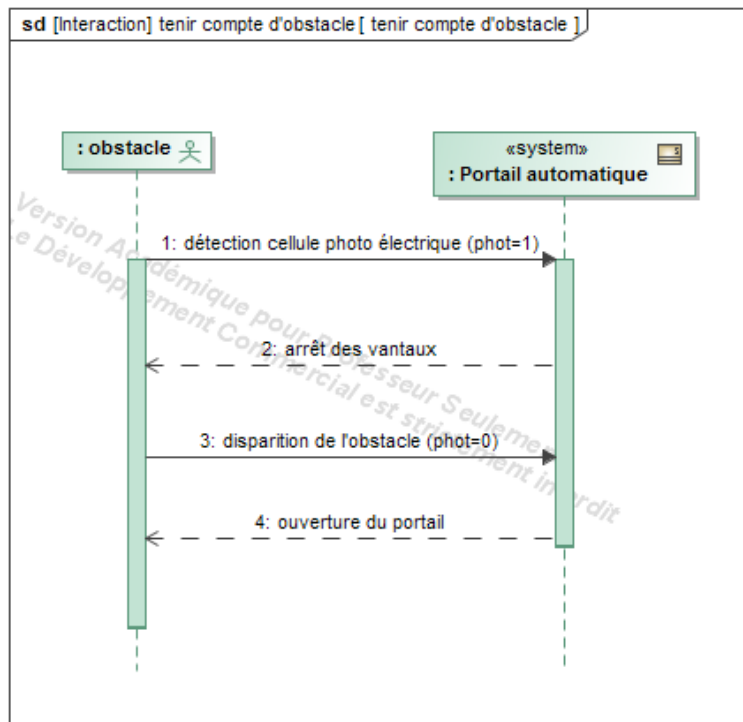


Diagramme de séquence pour le cas d'utilisation "Tenir compte d'obstacle" :

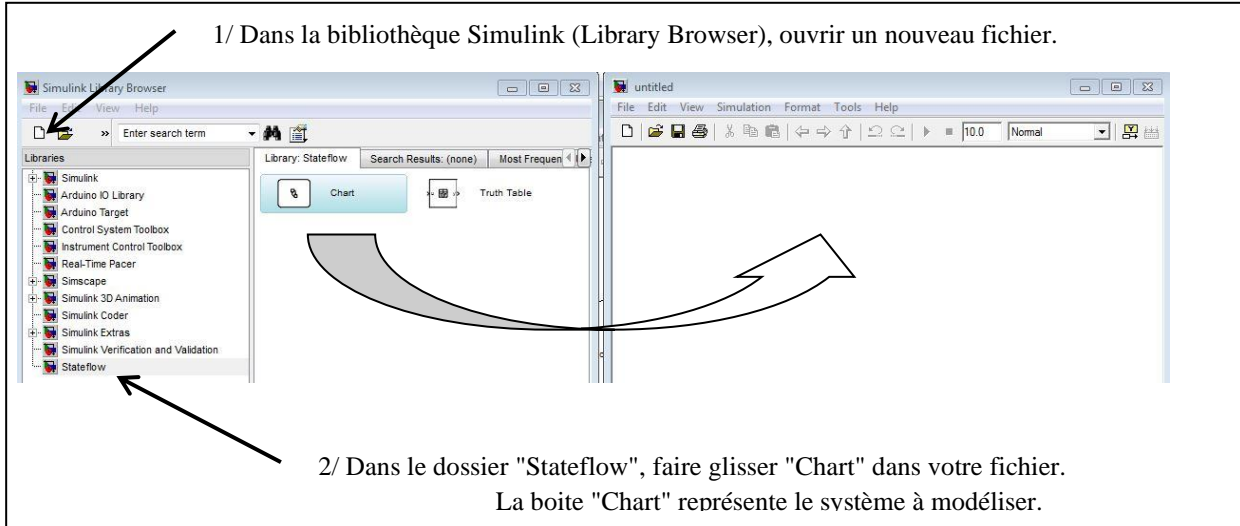


DOSSIER RESSOURCES

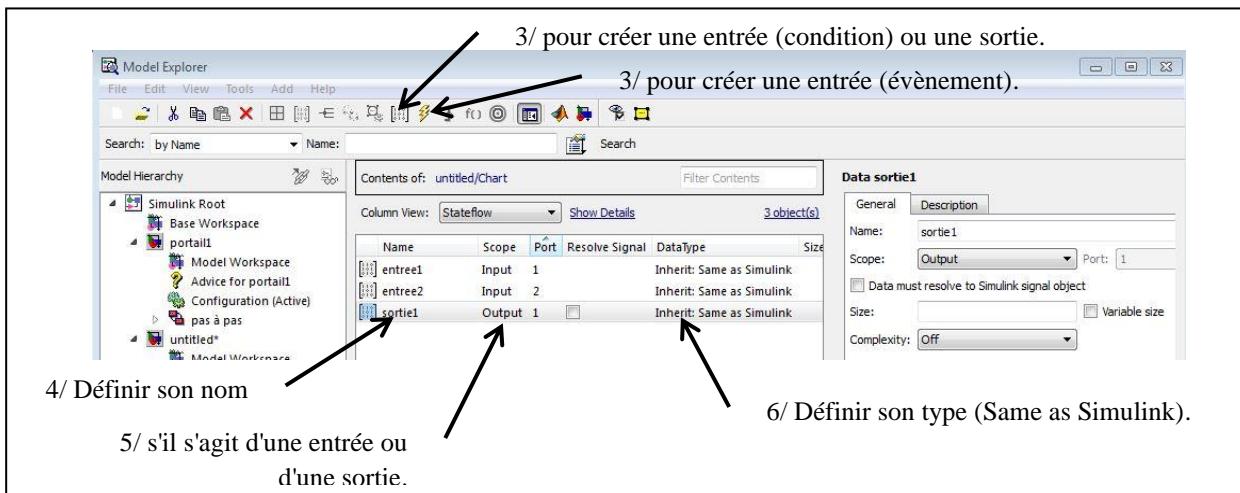
UTILISATION DE MATLAB STATEFLOW

Stateflow permet, dans l'environnement de Simulink de simuler le comportement d'un système grâce à son diagramme d'états.

POUR DEFINIR LES ENTREES/SORTIES :

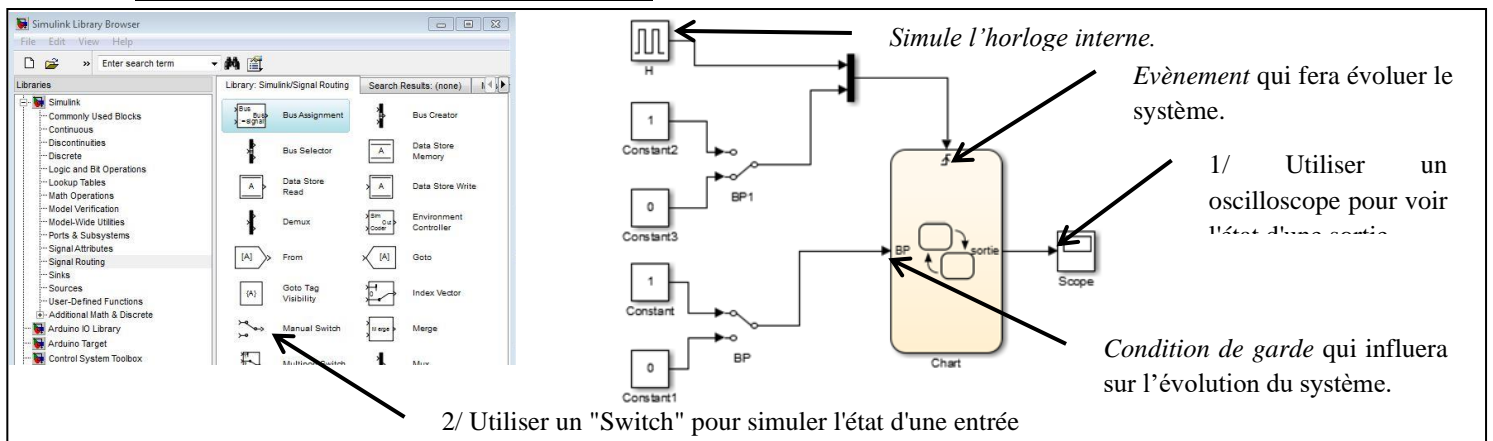


Dans le menu contextuel (clic droit sur la boîte "Chart"), choisir "Explore", pour définir les entrées (condition ou évènement) et les sorties :



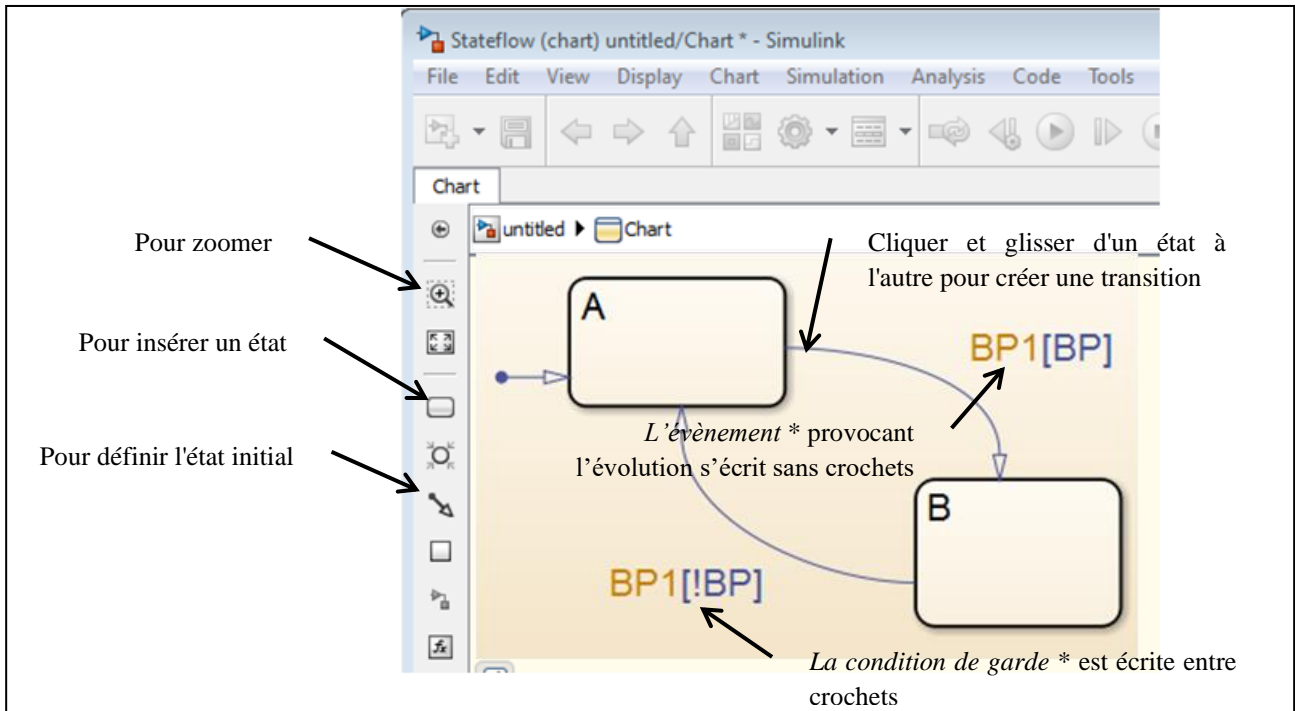
Les entrées/sorties ainsi définies apparaissent sur la boîte "Chart".

POUR SIMULER LES ENTREES/SORTIES :



POUR DEFINIR LE DIAGRAMME D'ETATS :

Double cliquer sur la boîte "Chart" pour ouvrir Stateflow et définir le diagramme d'états.



* : La transition est franchie lorsque l'évènement apparaît, si la condition de garde est vraie.

Les fonctions logiques s'écrivent :

NON	\bar{a}	!a
ET	$a \cdot b$	a&b
OU	$a + b$	a b

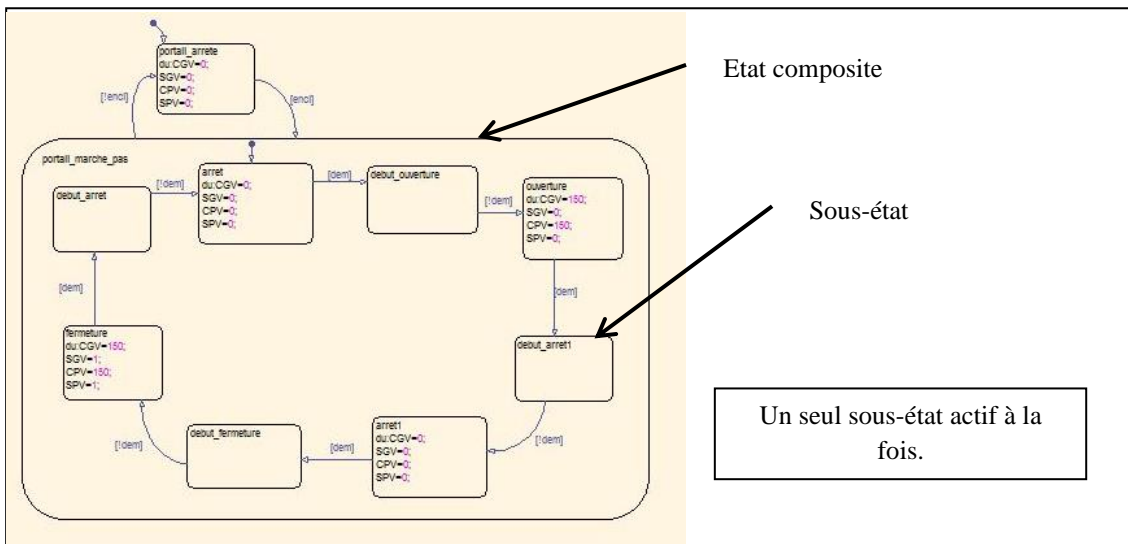
Les fonctions temporelles s'écrivent :

Après	after (12,sec)
Avant	before (10,sec)
A la date	At(11,tick)

Tick : temps d'échantillonnage du solveur.

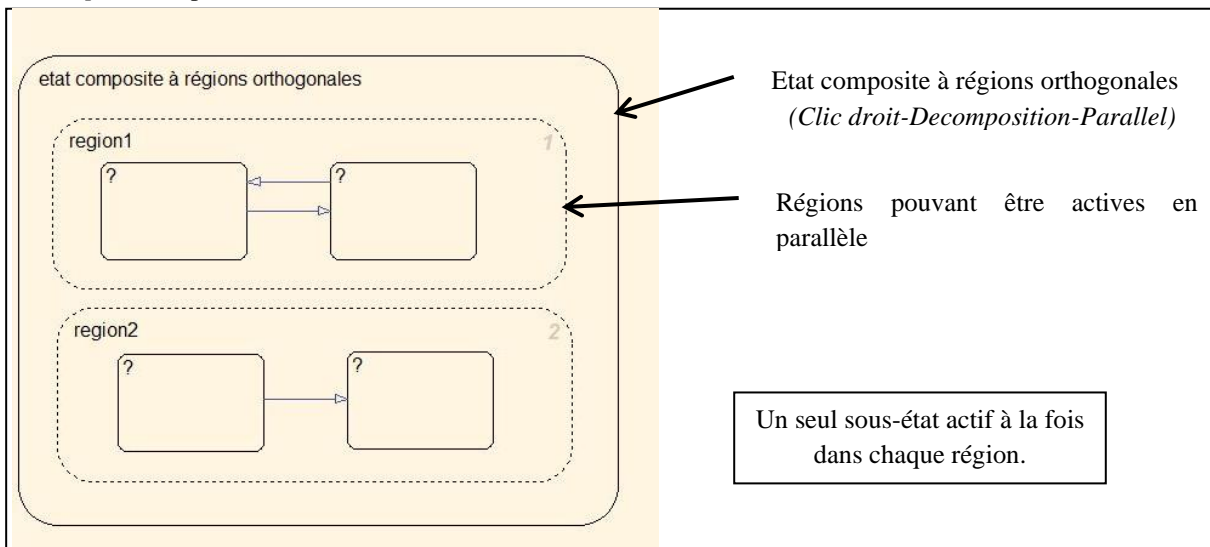
Etat composite :

On peut créer un état composite, en glissant à l'intérieur d'un état d'autres états : L'état composite doit être actif pour que l'un des sous-états le soit.

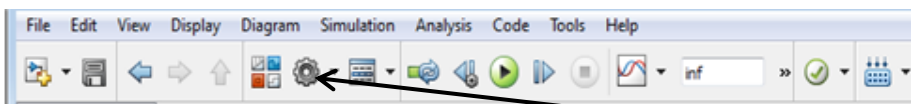


Etat composite avec régions orthogonales:

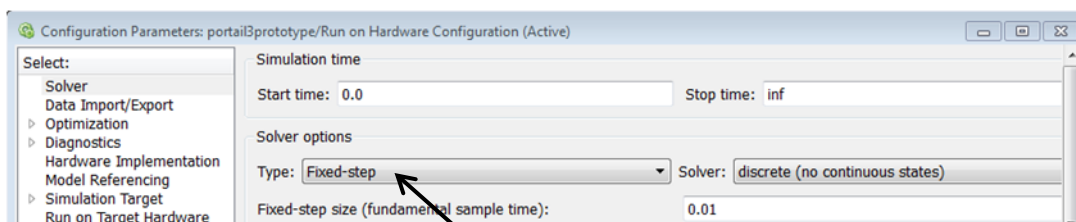
On peut créer un état composite avec régions orthogonales, en choisissant dans le menu contextuel (clic droit sur l'état) "Decomposition" puis "Parallele(AND)".



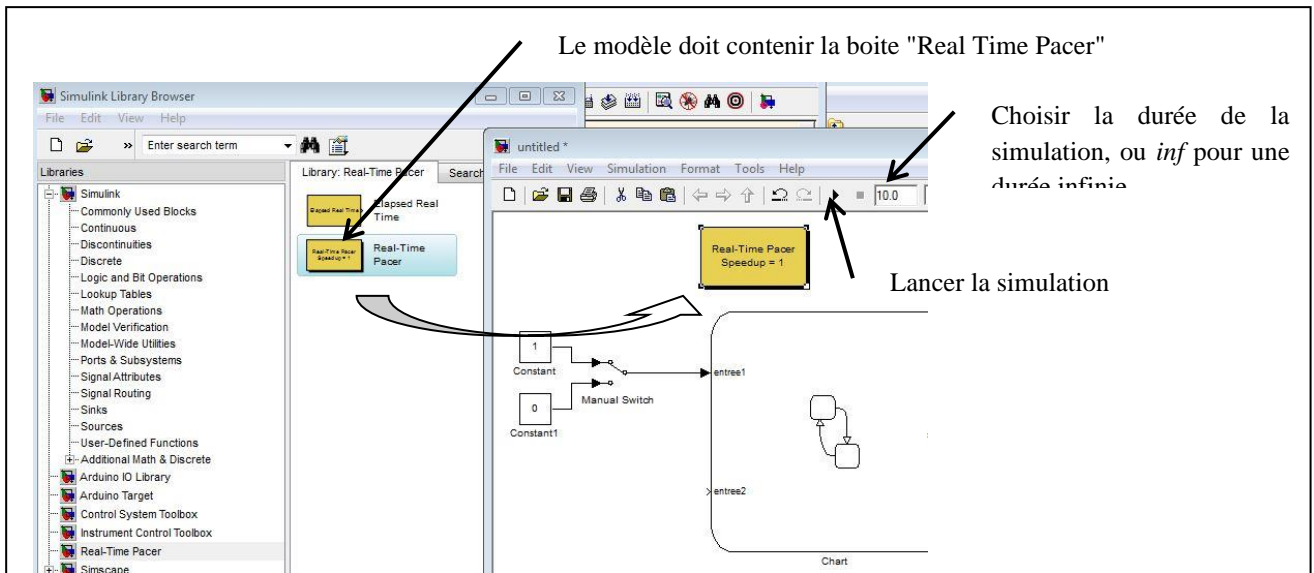
POUR LANCER UNE SIMULATION :



Sélectionner un solveur adapté à la simulation



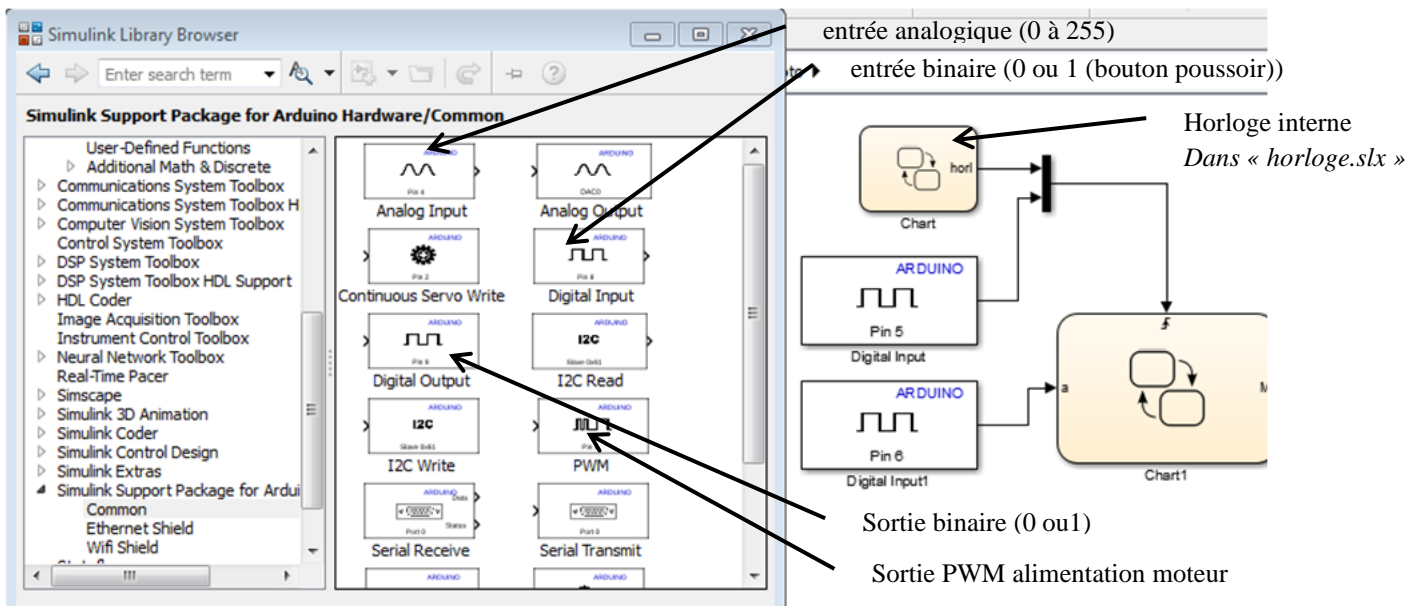
Choisir un solveur non continu à pas fixe



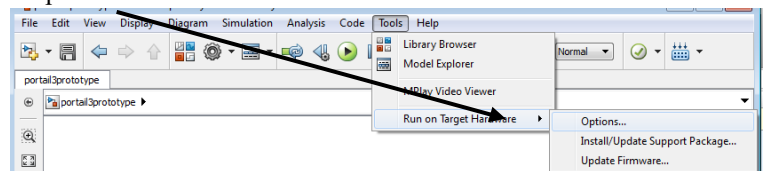
POUR IMPLANTER LE DIAGRAMME DANS LE MICROCONTROLEUR :

Il suffit de remplacer les entrées/sorties par les boîtes correspondantes du dossier "Simulink support package for Arduino Hardware/common".

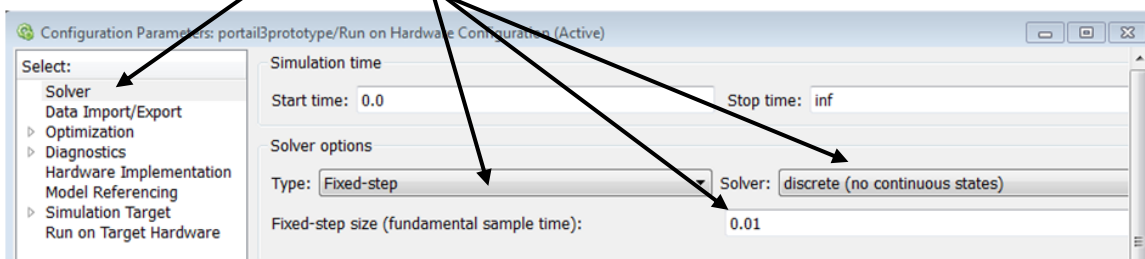
Remplacer également l'horloge interne de la simulation par la boîte « chart » du fichier « horloge.slx » fourni.



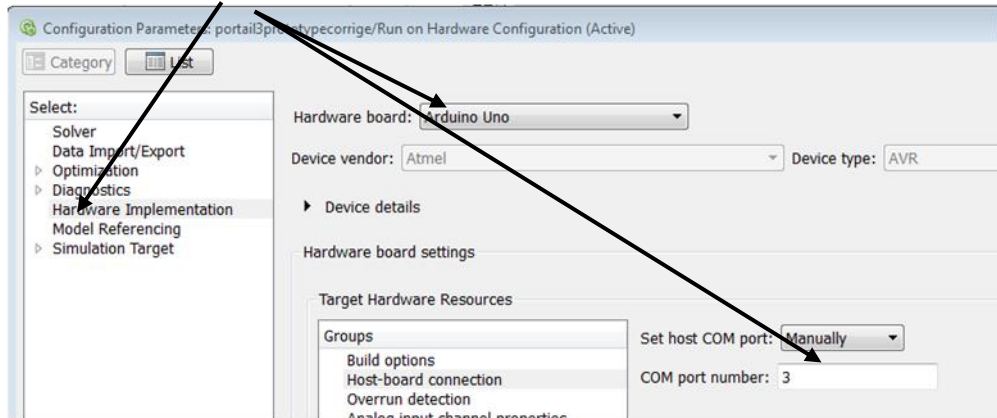
Dans le menu « Tools - Run on Target Hardware – Options... »



Vérifier les paramètres du solveur : (non continu à pas fixe)



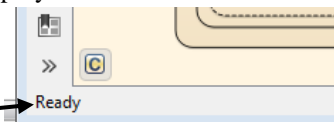
Définir la cible et le port de connexion :



Pour suivre l'évolution du diagramme d'états pendant le fonctionnement, choisir le mode « External »



Pour compiler et implanter le programme dans la cible, cliquer sur « Deploy to Hardware »



Bien attendre la fin de cette opération avant de passer à la suite...

Pour lancer le programme et observer l'évolution



Bien attendre la fin de cette opération avant de faire fonctionner le prototype...



On peut faire fonctionner le prototype et voir en même temps l'évolution du diagramme d'états.

