

Exercices - PPI – LabVIEW Design Pattern

Exercice 1

En LabVIEW, faire une machine d'état qui simule le feu d'un passage pour piéton.

Votre machine aura 3 états :

1. attendre bouton,
2. temporisation rouge,
3. traverser

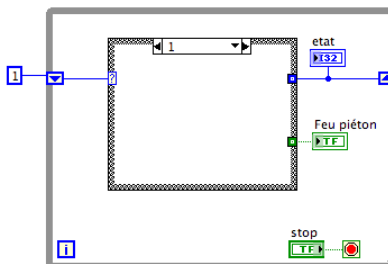


Au démarrage l'état par défaut est le 1 et le feu est rouge. La machine reste dans cet état tant que l'utilisateur ne presse pas sur le bouton « traverser ». Dès que le bouton « traverser » est appuyé, va dans l'état 2 et attend trois secondes, met le feu à vert, puis va dans l'état 3. Dans l'état 3, attend trois secondes, met le feu à rouge, puis va dans l'état 1.

Etapas

En vous inspirant du slide 22, esquissez votre machine d'états à l'aide de 3 boîtes, une par état et de flèches (transitions).

Transposez votre code en une machine d'états LabVIEW (slide 23). La structure de votre code devra comporter une boucle *while* avec à l'intérieur une *case structure* avec un *case* par état. L'état sera sauvegardé dans un *shift register*. Une ébauche de la structure devrait ressembler à :



A vous d'ajouter le code pour calculer le nouvel état et allumer/éteindre le feu.

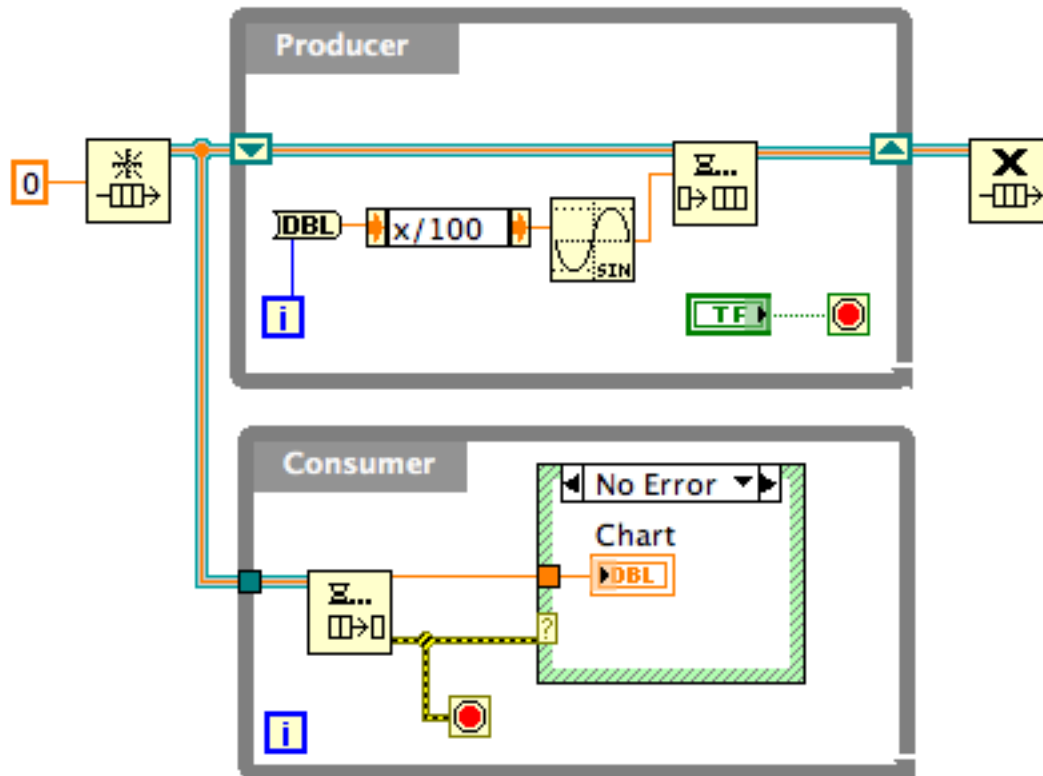
L'attente se fait à l'aide du VI **milliseconds to wait**

A l'aide du pinceau il est possible de définir les couleurs d'un bouton (boolean).

Testez et validez votre code.

Exercice 2

Câblez l'exemple du *slide 36* – producer/consumer.



Dans chaque boucles ajoutez une temporisation variable à l'aide du VI

milliseconds to wait 

Que se passe-t-il si la temporisation du *Producer* est plus grande que celle du *Consumer* ? Et inversement ?

Comment et pourquoi la boucle *while* du *Consumer* s'arrête ?

Ajouter une 3^e boucle *Producer* générant une sinusoïdale, qu'observez-vous ?

Exercice 3 (Q. examen 2016)

Que vaut **S** après l'exécution du *diagram* ci-dessous ?

