

TME 7 - Robot NXT - esterel

Exercice 1 – Robot NXT et Esterel

Nous avons adapté la sortie du compilateur `esterel` pour pouvoir la faire tourner sur une brique Mindstorm NXT.

Une brique mindstorm NXT est un microcontrôleur (en fait 2 !) sur lequel nous avons installé une machine virtuelle java nommée `lejos`. Ce microcontrôleur dispose des entrées suivantes :

- Boutons `ENTER`, `ESCAPE`, `LEFT`, `RIGHT`
- Un capteur de contact `Touch`
- Un capteur Ultrason `Distance`
- Un capteur de Son `Sound`
- Un capteur de Lumière, interrogé avec `Depassement_Lumiere` pour le seuillage et `Light` pour sa valeur courante.

Comme sortie nous disposons des sorties suivantes :

- `A_Avance`, `B_Avance`, `C_Avance`
- `A_Recul`, `B_Recul`, `C_Recul`
- `A_Stop`, `B_Stop`, `C_Stop`
- `A_Vitesse`, `B_Vitesse`, `C_Vitesse`
- `Bip`
- `LCD`, `LCDint`, `LCDfloat` (affichage de string integer et float respectivement).

De part l'existence physique de ces différents objets, nous avons une contrainte quant à l'interface minimale requise :

```
input ENTER,ESCAPE,LEFT,RIGHT;
input A_isMoving,B_isMoving,C_isMoving ;
input Distance : integer,Depassement_Lumiere, Sound;
input Touch;

output A_Vitesse : integer, C_Vitesse : integer;
output A_Avance, C_Avance,tmpsig,LCD: string;
output LCDint : integer, LCDfloat : float;
output A_Recul, C_Recul;
output A_Stop, C_Stop;
```

Question 1

En quoi `Light` est-il un sensor au sens esterel du terme mais `Touch` un simple signal d'entrée ?

Question 2

Écrire un programme esterel attendant le signal `Touch` et émettant le signal `Bip`.

Une fois ce programme écrit, pour le transférer sur la brique NXT le protocole est le suivant :

- compiler le programme vers java à l'aide du script `doall`
- Compiler le java à l'aide du compilateur `nxjc`
- Effectuer l'édition de liens, commande `nxjlink`.
- Charger le binaire produit dans le robot avec la commande `nxjupload`.

Ces deux dernières étapes peuvent être fusionnées en une seule en utilisant la commande `nxj`.

Question 3

Ecrire un programme estereel avançant le robot pendant 30 instants puis reculant le robot pendant 30 instants puis re-avançant etc... Combien de secondes durent 30 instants ? Ici vous devrez enlever les piles du robot pour l'arrêter.

Question 4

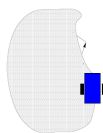
Modifier le programme précédant pour que l'appui sur `ESCAPE` mette fin au programme. A partir de maintenant il sera de bon ton de toujours implanter ce mecanisme dans vos programmes !

Question 5

Ecrire un programme permettant de calibrer le capteur de lumière pour qu'il sache distinguer une zone claire d'une zone foncée. On procédera comme suit : on placera le robot sur une zone claire on appuiera sur le bouton `ENTER` puis on le placera sur une zone foncée et on recommencera. On fera finalement afficher les deux valeurs du *sensor Light*.

Question 6

Ecrire un programme de suivi de forme : on souhaite que le robot utilise son capteur de lumière pour detecter à quel moment le robot chevauche une zone claire et une zone foncée. Lorsque le robot détecte un changement de surface, il se met à tourner sur place (bloque une roue) pour que le capteur se repalce à nouveau au dessus de l'ancienne surface, puis il bloque l'autre roue et se met à tourner dans l'autre sens pour replacer le capteur au dessus de la nouvelle surface etc.... ce qui nous donne le schéma suivant :



Question 7

Ecrire un programme qui met le robot à distance constante d'une cible mobile placée devant le robot.

Question 8

Modifier le programme précédant pour que le passage du robot sur une zone foncée provoque l'émission d'un Bip.

Question 9

Ecrire un programme avançant le robot en ligne droite et lorsque qu'un obstacle surgit sur sa route le contourne et reprend sa trajectoire initiale.

Annexe

Syntaxe de l'instruction run

```
run nommodule [ signal    extérieurs1/locals1, extérieurs2/locals2 ;  
                constant extérieurs1/localc1 ;  
                <et type, function, procedure, task> ]
```

Compilation d'un programme comportant des modules :

- 1) `esterel nom.str1` permet de contrôler syntaxe et erreurs grossières
- 2) pour compiler un programme comportant des modules, on doit donner la liste de tous les modules et le nom choisi pour le programme c (en général le nom du module principal racine des instanciations des autres modules)

```
esterel -simul nom1.str1 nom2.str1 -B nom3
```

Le programme C créé se traduit en exécutable comme tout autre programme Esterel :

```
gcc -m32 -o feux_tricolores4 feux_tricolores4.c -I \${ESTEREL}/include -L \${ESTEREL}/lib
```