

Programmation Synchrone — M2 Informatique et EIDD

Examen du 19 décembre 2017

Université Paris Diderot

Barème indicatif. Documents interdits. Toute la programmation se fait en SCADE
Durée : 3h (de 15h30 à 18h30). Halle aux Farines — salle 418C.

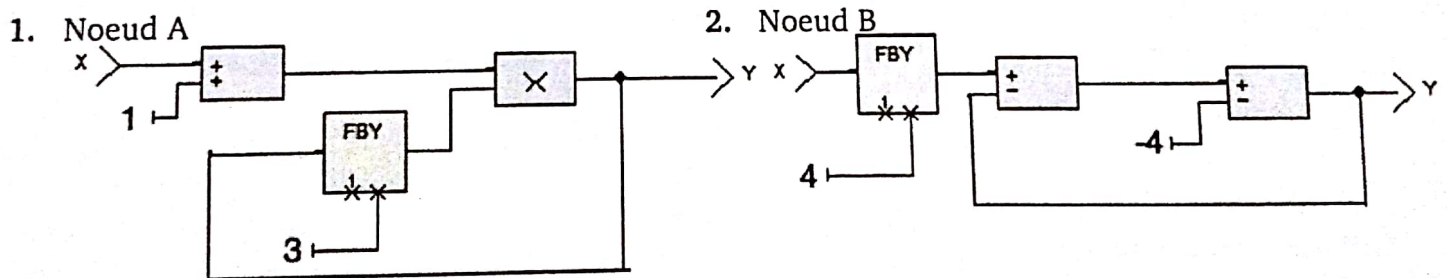
Exercice 1 (3 pts) – Terminologie

Expliquez en 1-3 lignes chacune des notions suivantes (dans le contexte du cours) *Barème avec bonus-malus : réponse fausse -0.25 ; pas de réponse ou réponse hors contexte 0 ; réponse incomplète ou imprécise 0,25 ; bonne réponse 0,5.*

- | | |
|----------------------|-------------------------|
| 1. Langage synchrone | 4. Logiciel critique |
| 2. Déterminisme | 5. Sécurité de logiciel |
| 3. Certification | 6. Erreur de causalité |

Exercice 2 (2pts) – Modèle synchrone

Est-ce que les opérateurs Scade ci-dessous sont corrects ? Si c'est le cas quelle est leur sortie Y pour l'entrée $X = 2, 1, 0, 1, 2, 0 \dots$ Sinon expliquez pourquoi.



Exercice 3 (2 pts) – Régulateur

La chaudière d'une machine à café est un procédé avec une entrée U (la tension électrique en Volts qu'on applique à la résistance de chauffage) et une sortie P (la pression de la vapeur dans la cuve en Bars). Concevez un régulateur PID qui maintiendra la pression très proche à 9 Bars.

Faites un schéma général, n'entrez pas dans les détails du calcul d'un intégrateur ou autre dérivée. Inutile donc de faire le programme SCADE complet.

Exercice 4 (3 pts) – Ordre supérieur et qualité de l'eau

Le système de monitoring de la qualité de l'eau mesure le niveau de 10 polluants toutes les 15 minutes (ce qui correspond à un tick). Il s'agit de la radioactivité, contenu en hydrocarbures, métaux lourds, dioxines, nitrates, bactéries *E. Coli* etc. Le niveau de polluant numéro i est $pol[i]$. La concentration maximale admissible de polluant numéro i est $cma[i]$. L'eau est considérée polluée si pour un i quelconque $pol[i] > cma[i]$.

1. Codez un opérateur SCADE qui reçoit les entrées $pol:real^{10}$ et $cma:real^{10}$ et renvoie la sortie $sale:bool$ qui est vraie si l'eau est polluée selon la définition ci-dessus. Est-ce un noeud ou une fonction ?
2. Codez un opérateur SCADE qui compte combien de fois depuis le lancement du système la variable $sale$ a pris la valeur true. Est-ce un noeud ou une fonction ?

Exercice 5 (6pts) – Climatiseur

Dans cet exercice vous devez programmer un climatiseur. La notation tiendra compte de la clarté et modularité de votre programme. Un tick d'horloge a lieu chaque minute.

1. Faites un opérateur `clim_base` avec
 - trois entrées entières : la température mesurée `temp`, et deux consignes `temp_on` et `temp_off` ;
 - une entrée booléenne `slow` ;
 - une sortie entière `clim` (puissance de climatisation : 0, 1 ou 2).La logique de fonctionnement est comme ceci :
 - au départ le climatiseur est à l'arrêt (`clim=0`) ;
 - si le climatiseur est à l'arrêt pendant au moins 5 minutes, et $temp > temp_on$, il se met en marche (`clim` devient 2 si `slow=faux` ou 1 si `slow=true`) ;
 - si le climatiseur est en marche pendant au moins 5 minutes, et $temp < temp_off$, il s'arrête.
2. Ajoutez un contrôle central avec une entrée entière `heure` qui commute le système entre deux régimes :
 - tous les jours (de 8 à 22h) on a $temp_on=27$, $temp_off=23$, `slow=false` et un LED vert est allumé. ;
 - toutes les nuits (de 22h à 8h) on a $temp_on=25$, $temp_off=20$, `slow=true` et un LED bleu est allumé.(décidez vous-même comment le contrôle central interagit avec `clim_base`, quelles sont ses sorties etc.).
3. Faites un opérateur `timer` avec entrée booléenne `minuit` qui est vraie chaque minuit (on reçoit un signal radio) et une sortie entière `heure`. Cet opérateur mesure le temps et renvoie à tout moment l'heure qu'il est (les heures sans minutes, entre 0 et 23), il se met à l'heure exacte tous les minuits.
4. Si ce n'est pas encore fait, dessinez le schéma global de votre climatiseur (inutile de recopier les opérateurs déjà faits, dessinez juste des boîtes avec leurs noms).

Exercice 6 (4 pts) – Algorithmique

1. Programmez la fonction SCADE `fact` avec une entrée entière n et sortie entière f telle que
 - pour $0 \leq n < 20$ on a que $f = n!$
 - pour les autres valeurs de n on a que $f = -1$.Indication : vous pouvez par exemple multiplier $1 * 2 * 3 * 4 * 5 * 1 * 1 * 1 \dots * 1$ pour calculer $5!$
2. Programmez la fonction `exp` (exponentielle) qui pour une entrée $x:real$ donnée calcule une approximation de $y \approx e^x$ selon la formule :

$$y = \sum_{n=0}^{10} \frac{x^n}{n!}.$$