

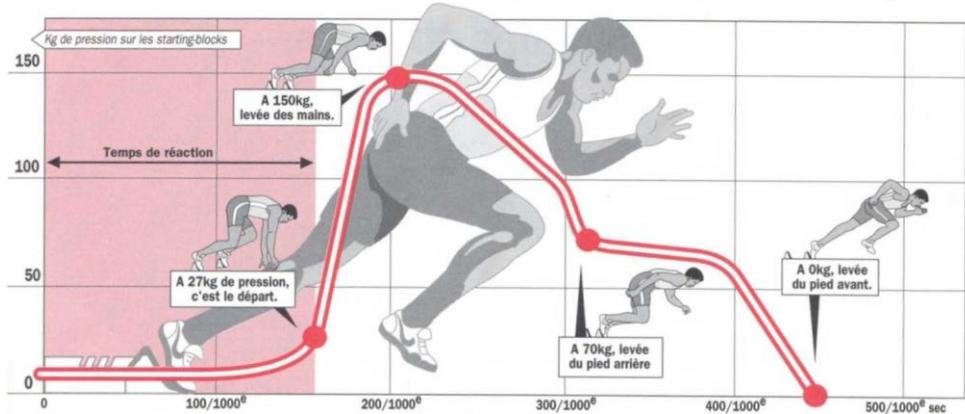
(durée 2 heures - aucun document autorisé)

INFORMATIQUE INDUSTRIELLE

L'étude porte sur l'analyse du départ de la course de 100m, où un ordinateur est chargé de surveiller les coureurs en détectant un éventuel faux départ :

- si un faux départ est détecté, le calculateur émet un signal sonore et indique le(s) couloir(s) en défaut ;
- si la course a pu démarrer, il affiche l'ordre de départ des coureurs ;
- dans tous les cas, il pilote des tables traçantes reproduisant l'évolution du départ de chaque coureur.

Le temps de réaction du sprinter (le temps qui s'écoule entre le signal du départ et la mise en action) est capital : les 3 premiers de la finale des JO de Rio 2016 se tenaient en 1 dixième de seconde, et du 2^e au 6^e en 7 centièmes ! Le temps de réaction moyen d'un être humain est de 1,5 dixième de seconde. Pour un sprinter, on accorde un minimum de 1 dixième.



En deçà, on considère que l'athlète a anticipé le coup de pistolet du starter. Le contrôle du départ s'effectue via les starting-blocks qui détectent le déclenchement de l'effort en fonction de la poussée exercée par le sprinter. A 27 kg de poussée, on considère que l'effort est produit et l'on enregistre le temps de réaction (si ce dernier est inférieur à 1/10 s, le starter est averti acoustiquement et annonce le faux départ). La poussée atteint un maximum de 150 kg quand les mains du coureur quittent le sol, elle tombe à environ 70 kg quand le pied arrière quitte son starting-block et à 0 quand le pied avant se soulève et que l'athlète a définitivement quitté l'emplacement de départ.

Le calculateur est relié aux capteurs et actionneurs du système de surveillance des couloirs de 6 coureurs :

capteurs	«A vos marques !» (Marques)	binaire, valeur de ce signal,
	«Prêts ?» (Prets)	binaire, valeur de ce signal,
	«Partez !» (Partez)	binaire, valeur de ce signal,
actionneurs	poussée (P1 à P6)	1 par couloir, 15 kg/V, de 0 à 180 kg, avec 0V pour 0 kg.
	table traçante (T1 à T6)	1 par couloir, axe Y gradué en 20 kg/1.5 cm pour 3 cm/V (0kg ≡ 0cm ≡ 0V), axe X gradué en 50 ms/cm pour 2 cm/V (0s ≡ 0cm ≡ 0V)
	alarme «Faux départ» (FD) couloirs défaillants (C1 à C6)	binaire, vrai pour signaler le faux départ, binaires, vrais pour indiquer les couloirs des coureurs en faux départ.
entrées/sorties calculateur	6 CAN	CAN flash 12 bits, de 0 à 12V : liés aux capteurs de poussée (P1 à P6)
	6 CNA	CNA à réseau pondéré, 8 bits, de 0 à 5V : liés aux tables traçantes (T1 à T6)
	port d'entrée numérique	3 bits utilisés : bit 0 : Marques bit 1 : Prets bit 2 : Partez
	port de sortie numérique	7 bits utilisés : bit 0 : FD bit 1 : C1 à bit 6 : C6

Etude technique

- 1) Tracer le schéma du système étudié, en indiquant les unités des signaux utilisés.
- 2) Etablir l'expression mathématique de la loi de conversion $Poussée_Mesuree = f(Valeur_Lue_Sur_CAN)$.
A quelle poussée mesurée correspond la valeur 2401 lue sur le CAN ?
Que renvoie le CAN pour une poussée de 27 kg ?
- 3) Etablir l'expression mathématique de la loi de conversion $Valeur_A_Ecrire_Sur_CNA = g(Poussée_A_Tracer)$.
Quelle Poussée sera tracée sur la table traçante en appliquant la valeur 205 au CNA ?
- 4) Les données $Poussée_Mesuree$ et $Poussée_A_Tracer$ sont représentées par des réels.
En expliquant le raisonnement retenu, donner la valeur des bits pour un codage en virgule fixe avec partie entière signée sur 4 bits et partie décimale sur 8 bits pour $Poussée_Mesuree = 26,625$ kg
Expliquer comment est constitué un nombre en virgule flottante simple précision (IEEE 754).
En détaillant la démarche adoptée, donner la valeur des bits de ce codage pour $Poussée_Mesuree = 26,625$ kg

Etude algorithmique

Veiller à : glossaires : préciser les données manipulées (nature, gamme de valeurs, signification, ...),
algorithmes : décrire clairement les traitements envisagés.

- 5) La loi de conversion $Poussée_Mesurée = f(Valeur_Lue_Sur_CAN)$ est décrite par une fonction.
Donner l'algorithme de ce traitement.
Retenir le prototype : fonction F (Valeur_Lue_Sur_CAN entier) réel
- 6) La conversion $Valeur_A_Ecrire_Sur_CNA = g(Poussée_A_Tracer)$ est décrite par une fonction.
Donner l'algorithme de ce traitement.
Retenir le prototype : fonction G (Poussee_A_Tracer réel) entier
- 7) Les capteurs binaires (*Marques, Prêts, Partez*) sont accessibles grâce à un port d'entrée numérique de 8 bits. En considérant qu'on dispose de fonction Port_Entree entier, renvoyant la valeur lue sur le port d'entrée numérique, donner les descriptions algorithmiques des fonctions suivantes :
- fonction f_Marques entier renvoie 1 si le capteur *Marques* est vrai, 0 sinon
fonction f_Prets entier renvoie 1 si le capteur *Prêts* est vrai, 0 sinon
fonction f_Partiez entier renvoie 1 si le capteur *Partez* est vrai, 0 sinon
- 8) Les actionneurs binaires (*FD* et *CI* à *C6*) sont associés à un port de sortie numérique de 8 bits. En considérant l'existence de fonction Port_Sortie (Valeur entier) qui applique en écriture *Valeur* sur le port de sortie numérique, donner les descriptions algorithmiques des fonctions suivantes :
- fonction FauxDepart_ON applique 1 au signal *FD* pour faire marcher l'alarme
fonction FauxDepart_OFF applique 0 au signal *FD*
fonction C_ON (x entier) applique 1 au signal *Cx* pour signaler le coureur x en faux départ
fonction C_OFF (x entier) applique 0 au signal *Cx* (x dans [1,6])
- Tenir compte dans chaque description que les autres signaux du port de sortie doivent être préservés et donc non modifiés : utiliser une variable globale représentant la valeur du port de sortie (ex : ValeurSortie).*
- 9) Algorithme général du calculateur de surveillance (abordé au *recto*).
- le programme attend la séquence des ordres « A vos marques », « Prêts », « Partez » puis il aborde la phase de départ pendant laquelle il surveille les coureurs : en comptant le temps, il considère qu'un coureur est fautif si la poussée de 27 kg est constatée avant la fin du 1^{er} dixième de seconde. Le signal « Faux départ » doit être émis, ainsi que le signal du couloir du fautif (il faut noter que plusieurs coureurs peuvent être concernés).
 - durant la phase de départ, qui est limitée à 1 seconde, le programme assure le tracé des différentes courbes de poussée. Il faut 2 informations par table traçante : la poussée et le temps courant. La 2nde information, commune à toutes les tables traçantes, est transmise en utilisant la fonction CNA_Temps () décrite ci-dessous.
 - les signaux binaires émis (*FD* et *CI* à *C6*) sont maintenus à leur valeur à la fin du traitement.
- Donner l'algorithme général du fonctionnement, en s'appuyant sur l'utilisation des fonctions F, G, f_Marques, f_Prets, FauxDepart_ON, ..., C_ON, C_OFF, ... CAN, ... etc.
- 10) Affichage de l'ordre de départ des coureurs, s'il n'y a pas eu de faux départ.
- un coureur est parti lorsque la poussée mesurée est égale à 0. Il faut donc constater l'ordre dans lequel les différentes poussées arrivent à 0 (elles sont >0 au départ). Pour cela, il est possible d'utiliser 2 vecteurs d'entiers : Classement[6] des coureurs du 1^{er} au 6^{ème}, et Presence[6] des coureurs sur la ligne de départ.
- Compléter l'algorithme décrit en 9) pour affichage éventuel (en fin de traitement) du classement du départ.

NOTAS :

-N1- pour la description des algorithmes, on dispose aussi des fonctions suivantes :

fonction CAN (Num entier) entier renvoie comme entier la valeur lue sur le CAN Num
fonction CNA (Num entier, Donnee entier) applique au CNA numéro Num la valeur entière Donnee
fonction CNA_Temps (Temps entier) applique au CNA de base de temps des tables traçantes la valeur entière Temps en ms.
fonction DELAI (Duree entier) retient le calculateur durant Duree ms

-N2- penser à l'initialisation des données.

-N3- la question 5 dépend de la question 2, la 6 de la 3, la 10 complète la 9 ; les autres sont indépendantes.