

AG43

Examen Final Automne 2015

Durée 1h30

Documents interdits

Commande de portail coulissant - Partie 2

Le sujet est la suite de l'exercice Portail Coulissant étudié en Travaux Dirigés. Il propose des questions complémentaires à celles déjà étudiées.

Récapitulatif :

Liste des capteurs (Ils sont vus comme des variables globales de type octet en lecture seule. Leur valeur est 0 ou 1)

- OUVERT
- FERME
- SIGNAL
- CELLULE

Liste des variables globales :

- LARGEUR : entier, largeur maximale d'ouverture de la barrière exprimée en nombre de pas moteur
- POSITION : entier, compteur qui indique la position en pas moteur de la barrière, valeur comprise entre 0 pour barrière fermée et LARGEUR pour barrière entièrement ouverte.
- MARGE : entier, donne la distance en pas moteur sur laquelle le moteur doit passer en vitesse lente avant ouverture ou fermeture complète
- TOLERANCE : Erreur de comptage autorisée
- VITESSE entier non signé. Durée de la temporisation entre deux pas moteurs
- TEMPO entier non signé. Compteur de temporisation utilisé par l'interruption `timer()`

Constantes (elles sont à considérer comme des valeurs numériques que l'on n'a pas à déclarer) :

- VLENTE : Valeur de la temporisation nécessaire entre deux pas moteur pour la vitesse lente.
- VRAPIDE : Valeur de la temporisation nécessaire entre deux pas moteur pour la vitesse rapide.
- OUVERTURE : Constante égale à 1
- FERMETURE : Constante égale à -1

On dispose des fonctions et procédures suivantes :

barriere(sens) : fonction de pilotage du déplacement de la barrière. Elle admet comme argument l'entier `sens` qui prend la valeur `OUVERTURE` ou `FERMETURE` (valeurs prédéfinies) et retourne un entier qui indique la raison de l'arrêt moteur. Elle accède aux variables globales `POSITION`, `LARGEUR`, `MARGE`, `OUVERT`, `FERME`, `SIGNAL`, `CELLULE` et `VITESSE`. Cette fonction retourne les valeurs suivantes :

- `sens` si la barrière s'est normalement ouverte ou fermée

- 2, si durant le mouvement d'ouverture ou fermeture, elle a reçu un signal de télécommande
- 3, si durant le mouvement de fermeture, la cellule détecte un obstacle

timer() : procédure de type interruption qui décrémente toutes les **10 ms** la variable globale entière TEMPO lorsqu'elle n'est pas nulle.

moteur(sens) : procédure provoquant le mouvement d'un pas du moteur, dans le sens indiqué.

Exercice 1 : Ecrire la procédure `tempo(duree)` qui effectue une attente **donnée en secondes** par l'argument entier `duree`. Si, durant l'attente elle détecte un signal de télécommande, elle met aussitôt fin à la boucle d'attente. Elle utilisera la variable globale TEMPO, mise à jour par l'interruption `timer()`. En tenant compte du type des variables utilisées (entiers signés), quelle sera la durée maximale en secondes de la procédure `tempo` ? On ne fera pas de test de cohérence sur la valeur de l'argument.

Exercice 2 : Modification de la fonction `barriere(sens)`

Dans la fonction `barriere(sens)` étudiée en TD, on a volontairement négligé les incidents suivants :

- moteur bloqué ou contact fin de course défectueux
- présence de neige durcie ou de glace activant intempestivement les contacts fin de course OUVERT ou FERME avant ouverture ou fermeture théorique complète

On demande de modifier la fonction `barriere(sens)` de façon à ce qu'elle prenne également en compte ces incidents de la manière suivante :

- La fonction retourne 4, si durant le mouvement d'ouverture ou fermeture, le contact OUVERT ou FERME est passé prématurément à 1 (présence de neige ou de glace sur la barrière)
- La fonction retourne 5, si le contact OUVERT ou FERME n'est jamais activé (moteur bloqué, ou contact défectueux).

Ce type d'incident est détecté en tenant compte de la position théorique du moteur (variable POSITION) et de la largeur d'ouverture (variable LARGEUR). On accepte une erreur de comptage qui doit être inférieure en valeur absolue à la valeur contenue dans la variable TOLERANCE. Ainsi, si dans le cas de l'ouverture, la valeur du compteur POSITION dépasse la valeur LARGEUR+TOLERANCE, ou, si dans le cas de la fermeture, POSITION devient inférieur à -TOLERANCE, on considère que l'on est en présence d'un problème de moteur ou de contact fin de course.

La fonction initiale `barriere` est :

```
Fonction barriere(sens)
Retourne un entier
```

```
Argument :
sens : entier, sens de marche souhaité
```

```
Variables Globales :
FERME, OUVERT, SIGNAL : octets
```

```

VITESSE : entier
TEMPO : entier
LARGEUR : entier
POSITION : entier

Variable locale :
Status : entier

VITESSE = VRAPIDE
Si1 sens = FERMETURE alors1
    Tant que FERME = 0 et SIGNAL = 0 et CELLULE = 0
        Si POSITION < MARGE alors
            VITESSE = VLENTE
        Fin Si
        TEMPO = VITESSE
        Moteur(sens)
        POSITION = POSITION +sens
        Tant que TEMPO <> 0
        Fin tant que
    Fin tant que
Sinon1
    Tant que OUVERT = 0 et SIGNAL = 0
        Si POSITION > LARGEUR - MARGE alors
            VITESSE = VLENTE
        Fin Si
        TEMPO = VITESSE
        Moteur(sens)
        POSITION = POSITION + sens
        Tant que TEMPO <> 0
        Fin tant que
    Fin tant que
Fin Si1

Moteur(0)

Si OUVERT = 1 ou FERME = 1 Alors
    Status = sens
Sinon
    Si SIGNAL = 1 Alors
        Status = 2
    Sinon
        Status = 3
    Fin Si
Fin si
Retourner status
Fin fonction

```

Exercice 3 : Fonction approche (sens)

On désire écrire une fonction de déplacement en butée à vitesse lente, limité à au maximum $LARGEUR + TOLERANCE$ pas pour palier à un éventuel défaut moteur ou capteur. Cette fonction déplace la barrière dans la direction `sens` et surveille le capteur correspondant au sens du mouvement (`FERME` pour `sens = FERMETURE`, `OUVERT` pour `sens = OUVERTURE`). Elle est destinée à

être utilisée pour la calibration de la largeur d'ouverture de la barrière. On dispose des variables globales TEMPO, OUVERT, FERME, LARGEUR, TOLERANCE. Si le déplacement se déroule correctement, la fonction retourne le nombre de pas moteurs effectués jusqu'au changement d'état du fin de course. Si l'on a dépassé le maximum de pas autorisés, on retourne -1.

Exercice 4 : Calibration de la barrière

La fonction calibration permet d'attribuer à la variable LARGEUR le nombre de pas nécessaires à l'ouverture totale de la barrière. Initialement, cette variable contient une valeur correspondant à une barrière de largeur record, afin de toujours arrêter le moteur au bout d'un certain nombre de pas. Cette variable permet, en fonctionnement normal, de détecter des défauts comme le blocage moteur, ou la défaillance d'un fin de course. La procédure de calibrage consiste à mesurer le nombre de pas nécessaires pour l'ouverture complète de la barrière, en partant de la position fermée. La procédure de calibration se déroule comme suit :

1. La barrière est amenée en position fermée (variable FERME = 1). LARGEUR est initialisé à VALEUR_DEFAULT.
2. On ouvre la barrière en comptant le nombre de pas nécessaires. Ce nombre de pas sera placé dans la variable LARGEUR. Si le capteur OUVERT ne passe pas à 1 (Le nombre de pas moteur est supérieur ou égal à VALEUR_DEFAULT), on retourne la valeur -1.
3. On referme la barrière en vérifiant le nombre de pas. La différence ne doit pas excéder la valeur contenue dans TOLERANCE
4. Si le comptage reste cohérent, on conserve la valeur de LARGEUR, on met la variable globale POSITION à 0 et on retourne 1, sinon, on se contente de retourner -1 et on initialise LARGEUR à VALEUR_DEFAULT.

On utilise la vitesse lente du moteur durant toute la procédure de calibration.

Ecrire la fonction `calibration()` qui recherche la valeur à placer dans LARGEUR, conformément au descriptif ci-dessus. On utilisera la fonction `approche(sens)` présentée question précédente.