Examen TR57

I Noyau temps réel PICOS18

On désire gérer une application multitâche temps réel à l'aide du noyau temps réel PICOS18.

I.1 Rappeler quelles sont les principales propriétés d'un noyau pour qu'il soit temps réel d'une part et embarqué d'autre part.
I.2 Indiquer comment les différentes tâches sont décrites logiciellement sous PICOS18.
I.3 Quel est l'intérêt des interruptions matérielles dans un système temps réel.
I.4 Indiquer comment sont pris en compte les interruptions matérielles avec PICOS18.
I.5 On désire gérer les réceptions de la liaison série RS232 par interruption et buffer circulaire. Renseigner les fonctions nécessaires de PICOS18 et écrire la fonction de gestion du buffer circulaire (l'initialisation du port série et des interruptions n'est pas traitée). On donne les éléments suivants :

- RCREG : registre de réception
- PIE1bits.RCIE : bit masque de l'interruption de réception sur le port série
- PIR1bits.RCIF : bit indicateur de l'interruption de réception sur le port série (remis à 0 par écriture d'un 0).

L'application à mettre en œuvre comprend les tâches suivantes, citées par ordre de priorité croissante :

- tâche 0 : affichage de paramètres par l'intermédiaire du port RS232 (tâche de fond)
- tâche 1 : lecture et traitement du buffer circulaire associé au port RS232 (tâche de fond)
- tâche 2 : faire clignoter la LED RB4 à 1 Hz comme témoin de fonctionnement du noyau
- tâche 3 : émission d'une trame CAN toutes les 10 ms contenant 3 consignes de 16 bits

I.6 Ecrire les descripteurs de chaque tâche et déclarer les variables et constantes associées.
I.7 Les tâches 2 et 3 sont déclenchées périodiquement. Indiquer comment déclarer les alarmes correspondantes, comment les initialiser et comment exécuter chaque tâche de manière synchronisée à chaque alarme. (On nommera les évènements postés LED_EVENT et CAN_EVENT).
I.8 Déterminer le rapport en % entre la durée de la transmission CAN 2.0A standard à 1 Mbauds et la période de déclenchement de la tâche 3. Conclure.

II Capteur intelligent

On utilise le DSP contrôleur TMS320LF2407A (40 MHz) comme capteur intelligent. Son rôle est de mesurer la grandeur analogique provenant d'un capteur, d'assurer la fonction de filtre numérique configurable, et de transférer les mesures filtrées vers un système hôte via une liaison CAN.

Le signal analogique est échantillonné à une fréquence de 10 kHz. Nous n'étudierons que le filtre du second ordre de fonction de transfert : $F(p) = \dfrac{1}{1 + \dfrac{2 \cdot m}{\omega_0} \cdot p + \dfrac{p^2}{\omega_0^2}}$

II.1 Etablir la fonction de récurrence donnant la sortie du filtre en fonction du signal non filtré, en utilisant l'approximation du rectangle inférieur.
II.2 On considère que le timer3 déclenche les conversions analogiques numériques. Le module ADC déclenchent ensuite l'interruption INT1 sur fin de conversion. Ecrire en langage C la fonction d'interruption ACQ correspondant au calcul du filtre (sans donner de valeur aux paramètres).

Pour éviter la perte de données lorsque le bus CAN est occupé, les valeurs à transmettre sont stockées dans un buffer circulaire par la fonction d'interruption ACQ. La transmission s'effectue en tâche de fond dans le programme principal.

II.3 Ecrire la fonction INITCAN réalisant l'activation et l'initialisation du contrôleur CAN (trame standard 2.0A, 1 Mbauds, Id = 0x100, mailbox 2, données sur 2 octets).

II.4 Indiquer comment déclarer le buffer circulaire et initialiser les variables qui lui sont associées.

II.5 Compléter la fonction d'interruption pour qu'elle gère le remplissage du buffer circulaire.

II.6 Ecrire la partie du programme principal permettant de gérer la transmission des données stockées dans le buffer circulaire.

II.7 Quel doit être la taille du buffer circulaire pour permet une indisponibilité du bus CAN d'une durée maximale de 1 seconde.

II.8 Calculer la fréquence d'échantillonnage maximale possible si les données devaient être transmises à chaque période d'échantillonnage.

En fin de transfert ou sur demande du système hôte, le capteur intelligent doit transmettre une série de données intermédiaires contenues dans la table PARAM de 256 octets.

II.9 Ecrire la fonction INITCAN2 modifiant la configuration (Id = 0x101, données sur 8 octets).

II.10 Ecrire la fonction TRANSPARAM et la fonction d'interruption INTERCAN associée, réalisant la transmission de la table PARAM en utilisant les interruptions du contrôleur CAN.