

Durée : 1h15

Tous documents autorisés

Une attention toute particulière sera apportée à l'évaluation de la clarté et de la simplicité des implémentations proposées ainsi qu'au bon respect des règles de codage (cf. ANNEXE photocopié)

PROJET – 3pts – 5mn

- Se rendre sous moodle, télécharger l'archive présente dans *download > année courante > eval > eval_nom.zip*, l'extraire sur le *bureau* puis renommer le répertoire racine avec votre propre nom. Créer un projet MPLABX du même nom dans le répertoire *eval_nom/apps/pjct* et y ajouter les sources applicatif présents dans *eval_nom/apps/src* ainsi que les sources de FreeRTOS. S'assurer de la bonne compilation du projet et du chargement de l'exécutable sur la cible (ou de l'exécution sur simulateur).

SQUELETTE – 9pts – 35mn

- Développer, compiler et exécuter le squelette complet mais vide de l'application. Ce squelette devra implémenter les points suivants :
 - Respecter les nommages des interfaces logicielles (tâches, outils système, etc)
 - Création et exécution des tâches
 - Création voire gestion des outils systèmes (queues de messages, mutex, sémaphores, etc)
 - Implémenter sous forme de commentaires les étages de traitement de chaque tâche. Ne pas implémenter le code réalisant chaque tâche (des commentaires succincts suffiront)
 - *Création voire gestion de tout outil système complémentaire potentiellement utilisé pour durcir l'application (mutex, sémaphore, section critique, etc)*

APPLICATION – 5pts – 30mn

- Développer, tester et valider l'application présenté dans le cahier des charges. Respecter l'exemple de sortie proposée sur terminal série.

A MERITER – 3pts – ∞

- Exponentiellement évolutif :
 - *Driver de réception UART par sémaphore à compteur*
 - *Contrôle de flux logiciel*
 - *Etc*

APPLICATION - CAHIER DES CHARGES

Cahier des charges

L'application consiste à développer un programme réalisant l'écho (simple renvoie vers l'ordinateur) des caractères réceptionnés avec une tâche complémentaire de supervision et d'analyse des données reçues. Si le caractère X majuscule (code ASCII 0x58) est reçu, alors il devra être retiré du buffer de réception (queue de message entre écrivain, lecteur et superviseur) et être remplacé sur la console par un message d'erreur (travail de la tâche de supervision et de gestion d'erreur). Dans tous les autres cas de figure, les caractères seront laissés inchangés (utiliser la fonction xQueuePeek de FreeRTOS). Vous trouverez un exemple de retour de console ci-dessous :

```
Start eval demo application
eval# a
eval # b
eval # d
eval # D
eval # *** X is a forbidden character ***
eval # a
eval #
```

Architecture logicielle et outils

Outils et versions à respecter : IDE MPLABX v5.15, toolchain C XC32 vs2.15 et framework Harmony v2,06. Implémentation de l'applicatif sur exécutif multi-tâches temps réel FreeRTOS v9.0.0. La modélisation de l'applicatif est donnée ci-dessous.

