

Systèmes Temps Réel

Travaux pratiques sur FreeRTOS

Matière	Systèmes Temps Réel
Support	Travaux pratiques sur FreeRTOS
Prérequis	Cours temps réel FreeRTOS – Yves Auffret
Année	M2/CIPA5
Durée	15 heures
Version	1.01– 2017-2018
Enseignant	Yves Auffret

TP
Mini projet

Durée 12 heures
(travail individuel)
+
Durée 3 heures
pour le portage sur la carte
STM32F4-Discovery
(travail individuel)

Acquisition, horodatage, démultiplexage,
traitement et enregistrement de données

L'objectif de ce TP est de réaliser un système d'acquisition, d'horodatage, de démultiplexage des données en fonction du type de message puis de traitement et d'enregistrement de ces données.

L'application se décompose en plusieurs tâches et utilise des mécanismes de file, de synchronisation entre les tâches, de gestion de priorité, de gestion de ressources partagées et de gestion de la tâche *idle*.

1) Tâche d'acquisition de données

Cette tâche permettra de simuler une acquisition de données en continu. Cette tâche devra pour l'ensemble des voies et de façon synchrone, générer à une fréquence précise de 1 Hz une trame de données (3 voies et un état d'acquisition). Chaque voie est composée de 4 octets dont le format est le suivant : numéro de voie de mesure sur un octet non signé, donnée aléatoire sur 24 bits signés correspondant au type de mesure. Ce mot de 24 bits est composé de trois octets dont le premier est l'octet de poids faible et le dernier l'octet de poids fort.

Le type de mesure devra être généré dans un ordre aléatoire parmi ces possibilités : 1=voie1, 2=voie2, 3=voie3, 4=l'état de l'acquisition des 3 voies.

Contrairement aux voies 1, 2 et 3, l'état de l'acquisition est suivi d'un seul octet. Pour simplifier, cet état sera toujours à zéro.

Cette tâche publiera au fur et à mesure les données dans une file. Il sera nécessaire de mettre en place un mécanisme pour signaler un éventuel débordement dans la file.

\$02	\$23(Pf)	\$34	\$32(PF)	\$01	\$03(Pf)	\$F3	\$F0(PF)	\$04	\$00	\$03	\$45(Pf)	\$FF	\$01(PF)
------	----------	------	----------	------	----------	------	----------	------	------	------	----------	------	----------

Exemple de données, dont l'ordre est aléatoire, qui doivent être générées à une fréquence précise de 1 Hz
Voie2 Voie1 Etat Voie3

2) Tâche d'horodatage, de démultiplexage et de traitement

Cette tâche devra lire les données de la file et les horodater (*ticks*) au moment de la lecture.

Le traitement consiste simplement à inverser les octets de données afin d'avoir en premier l'octet de poids fort, puis l'octet de poids moyen et en dernier l'octet de poids faible.

Chaque donnée horodatée et traitée sera publiée dans une file spécifique à chaque type de message : *file_1*, *file_2*, *file_3*. L'état de l'acquisition sera ignoré et ne sera pas publié dans une file.

Il sera nécessaire de mettre en place un mécanisme pour signaler un éventuel débordement dans chaque file spécifique.

3) Tâches d'enregistrement vers une ressource partagée

La sortie « debug (printf) » sera utilisée pour simuler un enregistrement utilisant une ressource partagée.

Pour mettre en pratique la gestion de cette ressource partagée, une tâche par file sera créée pour traiter chaque file spécifique : *file_1*, *file_2*, *file_3*.

Ces trois tâches devront écrire sur la sortie « debug (printf) » et dans l'ordre chronologique (donné par l'horodatage) les numéros de voie, les données converties en ASCII, les messages d'horodatage.

4) Calcul de la charge du système

La tâche *idle* sera utilisée pour calculer la charge du système afin de connaître la capacité à prendre en charge de nouveaux traitements. Le résultat sera affiché périodiquement en pourcentage sur la sortie « debug (printf) ».

5) Rédiger un compte rendu

La notation prendra en compte : l'architecture de votre programme, l'efficacité et la simplicité des algorithmes utilisés, la fiabilité de fonctionnement, les méthodes utilisées pour démontrer le bon fonctionnement du programme, les méthodes utilisées pour gérer les éventuels débordements dans les files, l'organisation de votre programme (fichiers, factorisation, procédures, fonctions, variables, constantes, alias, ...), les traces utilisées pour la mise au point, les règles de nommage, le nombre et la pertinence des commentaires, la clarté de présentation, la lisibilité de l'ensemble...

Inclure dans votre compte rendu un descriptif des algorithmes, les sources de vos programmes et une recopie des chronogrammes des différentes tâches.

Montrer à l'encadrant l'exécution de votre programme en fonctionnement normal et en cas de débordement dans la file.

Ajouter ce programme à votre compte rendu.

6) Portage de l'application sur la carte STM32F4-Discovery

En utilisant le package FreeRTOS préconfiguré pour la carte STM32F4-Discovery, vous porterez l'application sur la carte en utilisant les 4 LED centrales pour visualiser l'activité des tâches suivantes :

LED_Green = tâche acquisition de données
LED_Orange = tâche enregistrement file_1
LED_Red = tâche enregistrement file_2
LED_Blue = tâche enregistrement file_3

La sortie « debug (printf) » reste utilisable.

Inclure dans votre compte rendu les sources de vos programmes et une recopie des chronogrammes des différentes tâches.

Montrer à l'encadrant l'exécution de votre programme en fonctionnement normal.

Ajouter ce programme à votre compte rendu.

Déposer le compte rendu au format PDF sur l'ENT.
