5TC-BED, TP2 ez430-RF2500: OS et Protothread

Durée encadrée : 2 heures

But du TP

L'objectif est de pouvoir faire fonctionner une application radio sur la clé ez430 et de comprendre le principe de fonctionnement des *protothread* qui permettent de programmer une application sous la forme d'un certain nombre de tâches *concurrentes* tout en se passant d'un véritable OS.

1 Protothread

Les protothreads sont une forme de tâches coopératives inspirées des antique co-routines. Ils permettent l'exécution de tâches en temps partagé sans ordonnanceur. Nous utilisons l'implantation en C développée par Adam Dunkels. Les protothreads sont économes en mémoire car ils sont sans états: un thread ne possède pas de pile, il n'utilise que des variables globales, ils peuvent donc s'avérer peu performants en raison des accès à la mémoire globale, mais c'est surtout une contrainte de programmation assez forte.

Un protothread ne peut se bloquer (i.e. rendre la main) que sur une condition. cela simplifie l'écriture du code concurrent car les portions de code où le protothread peut rendre la main sont déterminées à l'avance, il n'y a donc pas de section critique à protéger, si ce n'est des interruptions. En contrepartie, il rend difficile, voire impossible l'écriture de code en temps réèl dur. Enfin, il est à la charge du développeur de vérifier qu'un protothread rendra effectivement la main à chaque fois qu'il sera exécuté sous peine de bloquer l'application dans ce protothread.

L'implantation que nous allons utiliser est très légère et basée sur des Macro en C.

- Placez vous dans le répertoire `$SRC/ez430-applications-students/example_protothreads`
- Cette application lance trois prototheads, l'un fait clignoter la led rouge, l'autre échantillone régulièrement la température et commande le troisième qui fait clignoter la led verte.
- Parcourez le code de l'application (fichier `main.c`), vérifiez son fonctionnement sur la clé.
- Que se passe-t'il si on enlève l'instruction ligne 95 (expliquez): `PT_WAIT_UNTIL(pt, led_green_flag);`
- Parcourez les fichiers `.h` du repertoire `protothreads` pour bien comprendre l'implémentation des macros utilisées.
- Comprenez comment fonctionne les protothreads de Dunkels en utilisant l'option `-E` de `gcc` qui passe uniquement le pré-processeur.
- Rajoutez un protothread qui ajoute un petit clignotement de la led rouge (10 ms) toutes les 3 secondes.

2 Communication radio

Les communications radio que nous allons étudier dans le TP sont simples, nous allons mettre en place un émetteur et un récepteur sans même utiliser de couche MAC. Les fonctions d'émission et de réception sont très proches de celles qui ont été vues en cours.

- Placez vous dans le répertoire `$SRC/ez430-drivers/examples/radio`

- Parcourez le code de l'application (fichier `main.c`), vérifiez son fonctionnement sur la clé. La compilation du programme donne deux exécutable, un pour l'émission et un pour la réception.
- Comprenez comment fonctionne les émissions / réceptions et essayez de bien mettre en évidence l'interface avec la radio.