

Cours Temps Réel embarqué

Projet Temps Réel avec Linux/Xenomai

Objectif :

Le but de ce projet est de faire découvrir et apprécier les outils temps-réel offerts par Xenomai. Pour cela nous utiliserons une carte Raspberry Pi 4/4b ou 3/3b/3b+ fournie par l'ESME sur laquelle on installera un système préparé à base du noyau GNU-Linux modifié avec un I-Pipe et équipé d'un micro-noyau Xenomai Cobalt version 3.

Présentation :

Le projet est de réaliser une ou plusieurs tâches temps réel dans l'environnement Linux/Xenomai tournant sur la carte RaspberryPi pour faire des acquisitions de données et transmettre les valeurs à une application non-temps réel afin que celle-ci puisse les envoyer, via le réseau TCP/IP, vers un système plus complet de type PC. Sur ce système, une application simple affichera le graphe représentant la courbe de contrôle des valeurs acquises.

Installation :

Les documentations indiquant comment installer les images sur les RPis sont dans les documents suivants :

- **InstallXenoRPi4B.pdf pour les RPis 4/4b,**
- **InstallXenoRPi3bp.pdf pour les RPis 3/3b/3b+**

ATTENTION :

Pour accéder au système Linux/Xenomai de la RPi avec ssh, il est fortement conseillé d'utiliser un câble ethernet.

L'adresse IP de l'interface Ethernet, une fois le système lancé, est :

- **sur les RPis 3 : 192.168.77.81**
- **sur les RPis 4 : 192.168.77.83**

Pour utiliser l'interface Wifi, vous devez configurer une adresse « à vous » qui ne risque pas d'être identique à celles utilisées par votre entourage.

Cette configuration se fait de 2 manières :

- avec le shell script **/etc/setwifi2 (si présent)** pour un test provisoire
- avec le fichier de configuration **/etc/dhcpd.conf** pour une modification pérenne.

Sujet :

Le but de ce projet est de montrer qu'il est possible d'exécuter une tâche temps-réel, chargée de faire des acquisitions de données à une fréquence élevée, et de faire en sorte qu'on puisse récupérer, quand on en a besoin et sans perturber l'acquisition, les données acquises pour les sauvegarder dans un fichier, les envoyer via le réseau TCP/IP, tracer une courbe, etc.

N'ayant à notre disposition ni carte d'acquisition de données, ni capteur, nous simulerons cette acquisition en faisant des prélèvements sur l'horloge du système. L'idée est la suivante : si l'on prélève la valeur de l'horloge avec une précision suffisante sur une durée finie que l'on transforme cette valeur en un réel représentant les secondes, on peut vérifier rapidement avec un logiciel qui permet de tracer des courbes, comme *gnuplot*, que la courbe obtenue par l'ensemble des points (N_i , $\sin(V_i)$), N_i étant le numéro de l'acquisition et V_i le réel obtenu lors de cette acquisition, doit être une belle sinusoïde.

Tout retard dans le système nous donnera une valeur décalée de l'horloge et provoquera une cassure rapidement repérable sur le graphe.

Plan proposé :

1°) **Prise en main de l'outil Gnuplot** qui va nous permettre de visualiser les courbes des valeurs d'acquisitions. Pour cela on commence par installer le package (apt install gnuplot sous root) et tester l'outil avec les exemples fournis. Voir la petite documentation (**DocGnuPlot.pdf**) ainsi que les exemples donnés en cours (**TestsGnuplot.tar.gz**).

Cet outil proposé est facultatif. Vous pouvez en choisir un autre (xgraph, R+, tcltk, Qt, etc).

2°) **Etude du code d'un programme non temps-réel** qui demande au système d'être réveillé à intervalles réguliers (type setitimer) afin de récupérer la valeur de l'horloge (cf gettimeofday) et d'enregistrer les valeurs dans un fichier au format lu par *gnuplot*. On utilise de préférence les fonctions POSIX 1003.1. Tester ce programme en modifiant la charge du système (on peut prolonger la durée d'acquisition pour mieux observer les variations).

Voir le fichier **AcquisSintGP.tar.gz** qui contient toutes les sources du code présenté en cours.

ATTENTION : Ces exemples NE SONT PAS TEMPS-REEL !!

3°) Toujours avec les fonctions POSIX 1003.1, et en vous inspirant de l'exemple fournit (**acquis.c**) qui montre un échange basé sur l'utilisation astucieuse d'un double buffer, mettre au point un algorithme de type producteurs/consommateurs permettant l'échange entre une tâche temps-réel et une tâche non temps-réel. D'autres solutions, avec ou sans sémaphore, sont aussi possibles.

4°) En utilisant, sur la RPi, le code du programme Xenomaï **testacq.c** (cf l'archive **TestacquisXeno_v3.tar.gz**) déterminer la fréquence optimale qui permettra de faire le travail sans erreur (**sans overrun !**).

5°) A partir des exemples de communication TCP/IP vus en cours, choisir un protocole de base (**UDP ou TCP**) et définir votre propre protocole pour assurer le transfert des données entre la RPi et le PC. **L'expliquer en détail dans le rapport à rendre** (cf la suite).

6°) En choisissant une interface Xenomaï (Posix ou Native) pour la programmation de votre projet, réaliser le projet final qui sera composé :

- d'une application Xenomaï comprenant une ou plusieurs tâches (temp-réel ou non) qui s'exécutera sur la carte Raspberry Pi et fera des acquisitions de données sur l'horloge,
- d'une (ou plusieurs) tâche(s) non temps-réel pour envoyer les données de l'acquisition à un PC via un réseau TCP/IP,
- d'une application simple sur le PC, qui récupérera les données et affichera des courbes via un outil simple tel que *gnuplot*.

Remarques : Il est conseillé de bien lire les documentations des API Xenomaï Native et Posix, (disponibles entre autre sur le site de Getall), et de bien comprendre les notions de « primary mode » et « secondary mode », mis en évidence dans les exemples de code présentés en cours et en TP.

Travail à rendre :

Chaque équipe devra rendre sa production dans les délais fixés par le responsable pédagogique, Monsieur Debadier. Le rendu se composera des choses suivantes :

1. les sources accompagnées du (ou des) Makefile, afin que la compilation soit instantanée,
2. un jeu de test éventuel pour présenter une démonstration du bon fonctionnement. **Un système fonctionnant en mode non-stop sera apprécié.**
3. un rapport de quelques pages, au format PDF, expliquant la démarche, les algorithmes, la fréquence d'acquisition choisie, le protocole de transfert, les difficultés rencontrées, etc.

Ces éléments seront regroupés dans un fichier **ESME_3S_STR_Gnn.tar.gz**, nn le no du groupe.

Ce fichier sera envoyé par email à : **esme@seriane.fr**

La date limite sera fixée lors de l'audit qui aura lieu le **jeudi 15 février 2024 à 8h30**.

Le rapport seul au format PDF doit être envoyer au plus tard le 11 février 2024.

Pour toute question : esme@seriane.fr

Bon courage !

P. Foubet – le 30 janvier 2024.