



open wide
INGENIERIE

Formation Linux embarqué et
temps réel

Version: 1.1

Auteur: Pierre Ficheux

Formation Linux embarqué et temps réel

Pierre Ficheux (pierre.ficheux@openwide.fr)

Du 19 au 22 novembre 2012.



open wide
INGENIERIE

Formation Linux embarqué et
temps réel

Version: 1.1

Auteur: Pierre Ficheux

MODIFICATIONS

<i>VERSION</i>	<i>DATE</i>	<i>AUTEUR(S)</i>	<i>DESCRIPTION</i>
1.0	20/1/2011	P. Ficheux	Création
1.1	5/12/2011	P. Ficheux	Amélioration du formatage



Table des matières

1.Introduction.....	5
2.Pré-requis.....	6
3.Programme de la formation.....	6
3.1.Linux embarqué (2 jours).....	6
3.2.Linux temps réel (2 jours).....	7



Index des figures

Figure 1. Module SODIM2410.....5

1.Introduction

Le but de la formation est de présenter les principaux aspects de la mise en place et de l'utilisation de Linux embarqué et des extensions temps réel du noyau Linux. Les travaux pratiques seront réalisés sur une carte ARM9 (S3C2410) et sur plate-forme x86.

Les stagiaires pourront utiliser leur PC (Windows ou Linux) ou Mac (Mac OS X 10.5 ou 10.6) puisque l'environnement de développement sera intégré à une image VirtualBox contenant l'environnement de développement. L'image de développement est basée sur la distribution Fedora 14.

Le matériel nécessaire aux travaux pratiques sera fourni aux stagiaires (carte ARM9). La carte est basée sur le module décrit ci-dessous.

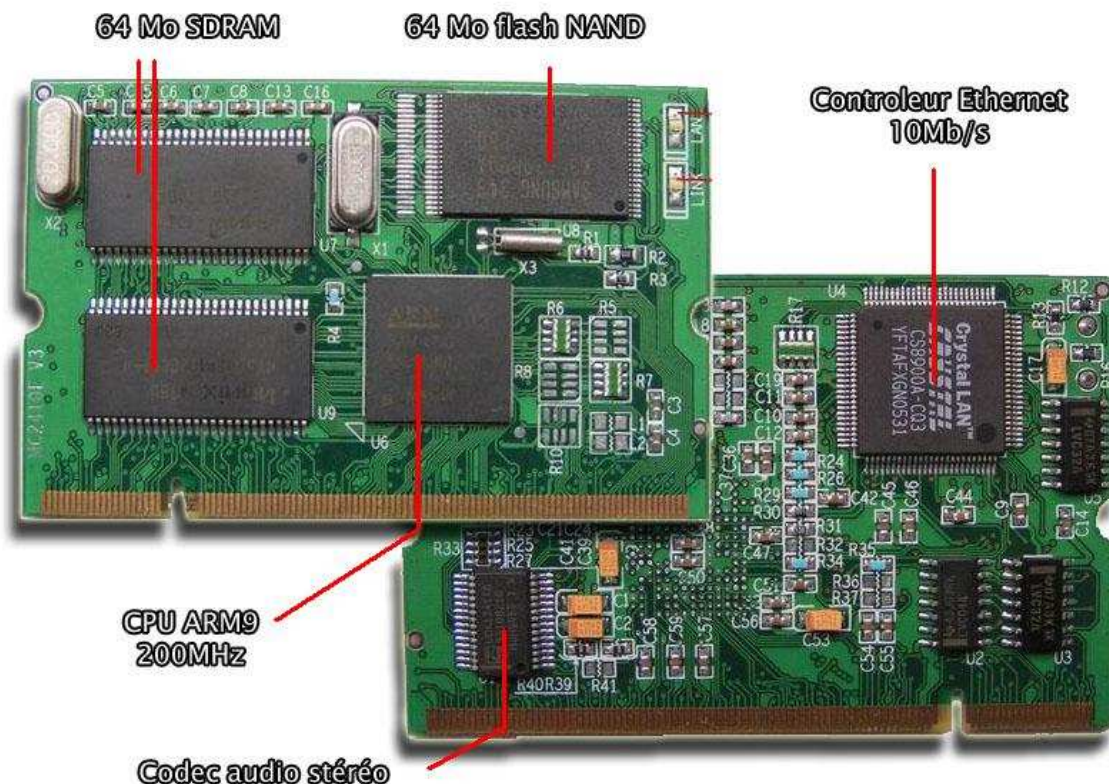


Figure 1. Module SODIMM2410

2.Pré-requis

Les connaissances nécessaires pour la participation au stage sont les suivantes:

- Connaissance utilisateur Linux ou à défaut UNIX
- Notions de langage C
- Notions de manipulation de la ligne de commande (shell)

Les pré-requis pour le matériel sont les suivants. Ces pré-requis sont **INDISPENSABLES** pour pouvoir suivre le stage :

- Un PC portable sous Windows ou Linux ou bien un MacBook sous Mac OS X 10.5 ou supérieur. La machine devra disposer **AU MINIMUM** de 2 Go de mémoire vive.
- Sur le PC, un port série RS-232 ou un câble d'adaptation série/USB afin d'exploiter la console de la carte ARM9. Ce câble ne fait pas partie du matériel fourni par Open Wide Ingénierie.

3. Programme de la formation

La formation est divisée en deux périodes de deux jours (14 heures) :

- Linux embarqué (2 j) [Linux_ARM9_SODIMM2410](#)
- Extensions temps réel pour Linux (2 j) [Prog_POSIX](#), [Xenomai_advanced](#)

3.1. Linux embarqué (2 jours)

- Quelques rappels sur GNU/Linux
- Licences GPL/LGPL
- Architecture de Linux : principaux fichiers et répertoires, procédure de démarrage
- Construction d'une distribution embarquée
 - Installation de la chaîne croisée Code Sourcery pour ARM
 - Notion de chargeur de démarrage (bootloader), focus sur U-Boot, compilation
 - Le noyau Linux: introduction, configuration, compilation
 - Busybox
 - Installation des bibliothèques
- Outils de construction de distribution
 - OpenEmbedded
 - Buildroot
- Utilisation de Buildroot, construction de la distribution ARM
- Test avec NFS-Root



- Introduction aux mémoires flash NOR et NAND, pilote MTD pour Linux
- Utilisation de la flash avec U-Boot
- Test de la distribution au format JFFS2, installation sur la flash de la carte
- Compilation d'une application minimale, installation sur la carte
- Mise au point croisée avec GDB/GDBSERVER
- Mise au point du noyau avec GDB/KGDB

3.2.Linux temps réel (2 jours)

- Programmation temps réel en POSIX sur noyau Linux standard
 - Configuration de l'ordonnanceur Linux (SCHED_OTHER, SCHED_FIFO, SCHED_RR)
 - Utilisation des threads et des compteurs POSIX.1 et POSIX.4
 - Démonstration des problèmes de latence avec un exemple simple (programme square)
- Solutions temps réel sous Linux
 - Rappels sur le temps réel
 - Linux standard
 - Les patch de préemption du noyau Linux (preempt-kernel, low-latency)
 - Le patch PREEMPT-RT
 - Technique du co-noyau (RTLinux, RTAI, Xenomai)
 - Tableau comparatif des solutions
 - Test des extensions temps réel de Linux sur la base d'une mini-distribution construite avec *Buildroot* et basée sur PREEMPT-RT, RTAI, Xenomai. Pour chaque extension, on mettra en évidence les performances en utilisant les outils `hackbench`, `latency`, `cylictest`, `stress`.
- Focus sur Xenomai
 - Introduction
 - Historique
 - Description fonctionnelle



- Architecture
 - Installation et mesure de performances avec les outils `latency` et `hackbench` (voir test PREEMPT-RT / RTAI)
 - Domaines d'exécution
 - Migration et conception d'application
 - API POSIX de Xenomai
 - Mise en œuvre de l'API POSIX sur l'exemple `xenomai_square` et comparaison avec l'application `square`
 - Présentation de l'API RTDM (Real Time Driver Model)
 - Modification de l'application `xenomai_square` pour utiliser le pilote RTDM
- Les tests des programmes temps réel seront effectués sur x86 (PC des stagiaires) et sur la carte ARM9.