

Linux embarqué : le projet μ Clinux



Patrice KADIONIK,
Maître de Conférence à l'ENSEIRB
kadionik@enseirb.fr

Mots clés : système embarqué, Linux embarqué, μ Clinux, Temps Réel

Cet article présente le projet μ Clinux, une distribution Linux dédiée aux systèmes embarqués. Après avoir précisé l'intérêt d'utiliser Linux dans le domaine de l'embarqué jusqu'alors réservé à des plateformes propriétaires, le projet μ Clinux et sa mise en œuvre seront présentés. Un exemple d'utilisation de μ Clinux sera ensuite donné pour un contrôle par le Web d'un équipement électronique.

1. SYSTEMES EMBARQUES, SYSTEMES TEMPS REEL. QUELQUES RAPPELS.

Un système embarqué peut être défini comme un système électronique et informatique autonome ne possédant pas des entrées/sorties standards comme un clavier ou un écran d'ordinateur. Le système matériel et l'application sont intimement liés et noyés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type PC [1] [2].

On peut citer comme exemples de systèmes embarqués :

- Un four à micro ondes.
- Une télécommande de TV.
- Une fusée.
- Un missile.
- ...

Généralement, un système embarqué doit respecter des contraintes temporelles fortes (*Hard Real Time*) et l'on y trouve enfoui un système d'exploitation ou un noyau Temps Réel (*Real Time Operating System, RTOS*).

Le Temps Réel est un concept un peu vague et chacun y va de sa définition dont l'auteur ☺. On pourrait le définir comme : "*Un système est dit Temps Réel lorsque l'information après acquisition et traitement reste encore pertinente*".

Cela veut dire que dans le cas d' une information arrivant de façon périodique (sous forme d'une interruption périodique du système), les temps d' acquisition et de traitement doivent rester inférieur à la période de rafraîchissement de cette information. Pour cela, il faut que le noyau ou le système Temps Réel soit *déterministe* et *préemptif* pour toujours donner la main durant le prochain *tick* à la tâche de plus forte priorité prête.

Une confusion classique est de mélanger Temps Réel et rapidité de calcul du système donc puissance du processeur (microprocesseur, microcontrôleur, DSP). On entend souvent : « *Etre temps Réel, c'est avoir beaucoup de puissance : des MIPS, des MFLOPS...* ».

Ce n'est pas toujours vrai. En fait, être Temps Réel dans l'exemple donné précédemment, c'est être capable d'acquitter l'interruption périodique (moyennant un temps de latence de traitement d'interruption imposé par le matériel), traiter l'information et le signaler au niveau utilisateur (réveil d'une tâche, libération d'un sémaphore...) dans un temps inférieur au temps entre deux interruptions périodiques consécutives. On est donc lié à la contrainte durée entre deux interruptions. Si cette durée est de l'ordre de la seconde (pour le contrôle d'une réaction chimique par exemple), il ne sert à rien d'avoir un système à base de Pentium III ! Un simple processeur 8 bits du type microcontrôleur Motorola 68HC11 ou Microchip PIC ou même un processeur 4 bits fera amplement l'affaire ; ce qui permettra de minimiser les coûts sur des forts volumes de production.

Il faut toujours garder à l'esprit cette donnée spécifique de l'embarqué où économiser une simple résistance à 0,20 Francs (0,03 Euros) permet d'économiser 200000 Francs (30 490 Euros) sur un volume d'un million de pièces !

Si ce temps est maintenant de quelques dizaines de microsecondes (pour le traitement des données issues de l'observation d'une réaction nucléaire par exemple), il convient de choisir un processeur nettement plus performant comme un processeur 32 bits Motorola 68K ou ColdFire.

Il est à noter que contrairement à l'environnement grand public PC où les processeurs Intel et AMD sont rois, ce sont plutôt les processeurs Motorola 68K, ColdFire, PowerPC... qui sont utilisés dans l'embarqué sous forme de cartes au standard VME (*Versa Module Eurocard*).

Ces processeurs sont réputés pour leur rapidité de traitement des interruptions et pour la prise en charge d'interruptions avec un niveau de demande d'interruption...

L'exemple donné est malheureusement idyllique (quoique fréquent dans le domaine des télécommunications et réseaux) puisque notre monde interagit plutôt avec un système électronique de façon aperiodique. Il convient donc avant de concevoir ledit système de connaître la durée minimale entre 2 interruptions ; ce qui est assez difficile à estimer voire même impossible. C'est pour cela que l'on a tendance à concevoir dans ce cas des systèmes performants (entre terme de puissance de calcul CPU et rapidité de traitement d'une interruption) et souvent surdimensionnés pour respecter des contraintes Temps Réel mal cernées à priori. Ceci induit en cas de surdimensionnement un surcoût non négligeable...

Outre les contraintes Temps Réel que l'on retrouve souvent dans un système embarqué, il existe d'autres contraintes importantes à prendre en compte :

- L'encombrement.
- L'environnement extérieur.
- L'aspect mécanique.

- La consommation.
- La tolérance aux fautes.
- ...

2. LINUX ET LE MONDE DE L'EMBARQUE

Linux depuis presque 3 ans est en train de conquérir un domaine où on ne l'attendait pas vraiment : l'univers des systèmes embarqués.

Pourquoi retrouve-t-on Linux dans l'embarqué ? Tout d'abord pour ses qualités qu'on lui reconnaît maintenant dans l'environnement plus standard du PC grand public [3] [4] :

- Libre, disponible gratuitement au niveau source : pas de royalties à reverser.
- Ouvert.
- Différentes distributions proposées pour coller au mieux à un type d'application.
- Stable et efficace.
- Aide rapide en cas de problèmes par la communauté Internet des développeurs Linux.
- Nombre de plus en plus important de logiciels disponibles.
- Connectivité Internet en standard.

Linux a aussi d'autres atouts très importants dans le domaine plus feutré des systèmes embarqués :

- Portage sur processeurs autres que x86 : PowerPC, ARM, MIPS, 68K, ColdFire...
- Taille du noyau modeste compatible avec les tailles de mémoires utilisées dans un système embarqué (500 Ko pour Hard Hat Linux de MontaVista).
- Différentes distributions proposées suivant le domaine : routeur IP, PDA, téléphone...
- Support du chargement dynamique de modules qui permet d'optimiser la taille du noyau.
- Migration rapide et en douceur pour un spécialiste Linux à Linux embarqué ; ce qui réduit les temps de formation (et les coûts...).

On retrouve généralement aussi un certain nombre de suppressions de fonctionnalités dans les distributions Linux embarqué :

- Pas de gestion de la MMU (*Memory Management Unit*) pour ne pas pénaliser les performances globales du système.
- Utilisation de systèmes de fichiers en mémoire RAM (*RAM Disk*) ou en mémoire FLASH (*FLASH Disk*) [2].

On a en fait entendu parler pour la première fois officiellement de Linux embarqué à une exposition *Linux World* en 1999 où les sociétés Motorola, Force et Ziatech ont présenté un système CompactPCI fonctionnant sous Linux.

En 2000 a été créé le consortium Linux embarqué (*Embedded Linux Consortium*) dont le but est de centraliser et de promouvoir les développements de solutions Linux embarqué. Ce consortium regroupe des éditeurs de distribution Linux, des éditeurs de systèmes Temps Réel propriétaires (comme WindRiver pour VxWorks) et des fabricants de composants. Il compte actuellement plus de 100 membres.

Les distributions Linux embarqué se sont rapidement imposées face à des distributions propriétaires généralement Temps Réel comme VxWorks, pSOS, QNX... où l'on est d'abord obligé de payer pour accéder à la plateforme de développement puis de payer des royalties

pour chaque système (ou cible) que l'on commercialise ensuite. Il est à noter que l'on observe une évolution de ce système à péage de certains face à la « menace » Linux.

Linux embarqué supporte aussi différentes extensions Temps Réel qui mettent en place une couche d'abstraction logique entre matériel, interruptions et Linux. Linux et l'ensemble des processus sont généralement considérés comme la tâche de fond exécutée quand il y a rien de Temps Réel à faire...

Généralement, on peut développer des threads Temps Réel conformes à la norme POSIX ; ce qui facilite aussi la migration d'un spécialiste Linux à Linux Temps Réel.

On peut citer comme extensions Temps Réel :

- La distribution RTLinux et sa distribution Mini RTLinux pour l'embarqué.
- La distribution RTAI.

Le tableau suivant établi par la société Lineo, leader dans ce secteur, donne une image du marché des systèmes embarqués où Linux se positionne.

Besoin	Miniature	Petit	Moyen	Haut de gamme	PC embarqué	Embarqué haute disponibilité
Taille RAM	<0,1 Mo	0,1-4 Mo	2-8 Mo	8-32 Mo	16-64 Mo	> x Mo
Taille ROM/FLASH	0,1-0,5 Mo	0,5-2 Mo	2-4 Mo FLASH	4-16 Mo FLASH	Xx Mo	Go-To
Processeurs	DragonBall 68K Mcore ColdFire ARM	MIPS Hitachi SH x86 PowerPC				Pentium PowerPC
Caractéristiques matérielles	MMU optionnelle	Ardoise Internet Carte unité centrale System on Chip (SoC)				CompactPCI
Exemples d'applications	Caméra numérique PDA Téléphone	Routeur Décodeur Stockage en réseau Imprimante en réseau				Commutateur téléphonique Routeur haute performance Serveur central

Tableau 1 : Linux dans le marché de l'embarqué

3. LE PROJET μ CLINUX

Parmi l'ensemble des distributions Linux embarqué [5], nous avons choisi de présenter μ Clinux qui est de plus en plus utilisé aujourd'hui.

μ Clinux (prononcer « you see Linux ») est l'acronyme de *Micro Controller Linux*.

Le projet μ Clinux lancé en janvier 1998 est un portage de Linux version 2.0.x originellement sur des processeurs ne possédant d'unité de gestion mémoire MMU (*Memory Management Unit*).

C'est en fait à la base un portage de Linux sur microcontrôleur Motorola 68328 et dérivés que l'on trouvait par exemple dans les ordinateurs de poche (*Handheld*) PalmPilot.

Outre la famille Motorola 683xx, il existe maintenant des portages μ Clinux pour processeurs Motorola ColdFire, i960 d'Intel, ARM7TDMI et depuis peu NIOS d'Altera (voir bibliographie).

μ Clinux basé sur le noyau Linux 2.4.x est maintenant opérationnel.

L'absence de MMU impose quelques limitations d'usage par rapport à l'environnement Linux :

- La mémoire virtuelle n'existe pas.
- L'appel système *fork()* n'est pas supporté. Il faudra utiliser alors une implémentation de l'appel système *vfork()* d'UNIX BSD (le processus parent est suspendu jusqu'à ce que le processus fils appelle *exec()* ou *exit()*).
- L'appel système *exec()* ne peut pas charger actuellement une image binaire supérieure à 256 Ko.
- La taille de la pile est fixe pour chaque processus

Outre le noyau μ Clinux, on retrouve dans la distribution les outils de développement (compilateur C, debugger...) GNU ainsi que le portage d'applications sous licence GPL (shells, serveurs Web...).

Les caractéristiques de μ Clinux données pour le portage Motorola 683xx (sur cible μ CSimm) sont :

- API compatible avec Linux.
- Taille du noyau inférieure à 512 Ko.
- Taille du noyau et commandes Linux inférieures à 900 Ko.

Un point intéressant déjà signalé est que μ Clinux inclut la pile Linux TCP/IP complète ; ce qui permet à l'équipement électronique de bénéficier d'une connectivité Internet (sur liaison série ou sur une interface Ethernet).

RTLinux a aussi été porté sur μ Clinux pour avoir finalement un système embarqué Temps Réel (voir bibliographie).

4. LES PLATEFORMES MATERIELLES POUR μ CLINUX

4.1. La plateforme originelle : le kit μ Csimm

Les responsables du projet μ Clinux ont proposé originellement une plateforme matérielle à des fins de tests : le kit μ Csimm [6].

Ce kit comprend un module qui se présente sous la forme d'une barrette SIMM à 30 broches. Le module μ Csimm se compose de :

- Un microcontrôleur DragonBall EZ (68EZ328) basé sur un cœur Motorola 68000 à 16 MHz. Il intègre un contrôleur de DRAM, un port série (interface RS.232), une interface SPI, un contrôleur LCD (résolution QVGA), une sortie Timer et une sortie PWM.
- Jusqu'à 18 E/S parallèles.
- Une horloge Temps Réel – calendrier.
- De la mémoire dynamique DRAM de 8 Mo.
- De la mémoire FLASH de 2 Mo.

- une interface Ethernet IEEE 802.3 10BaseT.

Le module μ Csimm est commercialisé par RtControl / Lineo sous forme d'un kit de développement comprenant le module μ Csimm, un carte d'assemblage / développement (kit μ Cgardener), le CD de développement, l'alimentation au prix de 575 USD.



Figure 1 : le module μ Csimm

Le kit μ Cgardener est une carte d'assemblage qui sert de support au module μ Csimm. Il comprend :

- Un support SIMM 30 broches.
- Un régulateur 3,3V.
- Un connecteur DB9 pour la liaison série avec le PC hôte.
- Un connecteur RJ45 pour l'interface Ethernet.
- Une zone à wrapper pour une éventuelle extension matérielle.

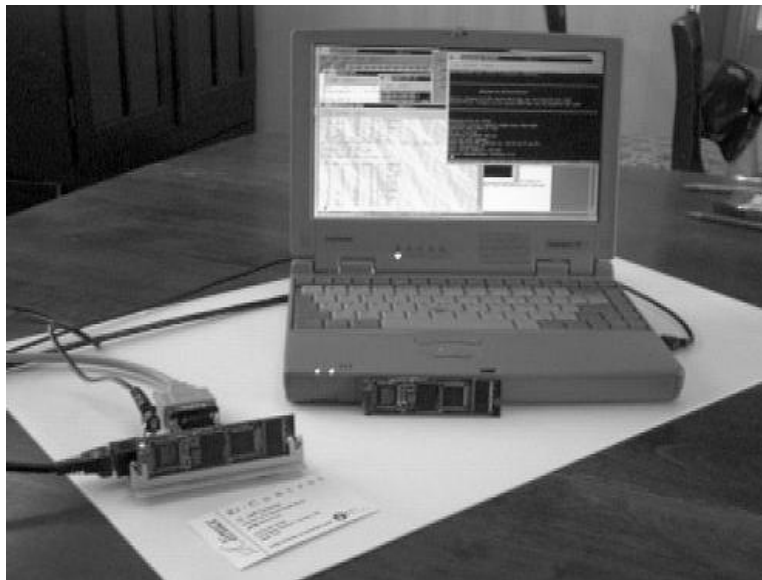


Figure 2 : Mise en œuvre du kit μ Csimm

La mémoire FLASH contient un moniteur (*bootloader*) qui permet de télécharger une image via la liaison série en mémoire DRAM et éventuellement la recopier en mémoire FLASH. L'image contient par défaut le noyau μ Clinux et son systèmes de fichiers avec un petit nombre d' utilitaires (client NFS, miniserveur Web...).

Le manuel fourni avec le kit donne toutes les informations pour une installation pas à pas et donne un exemple d'application avec son intégration dans l'image.

Il existe aussi un kit de développement μ Clinux à faible coût : le kit Geek. Il est composé seulement des modules μ Csimm / μ Cgardener à assembler soi-même. Il est proposé par Lineo à 345 USD.

4.2. Le kit μ Cdimm

Le kit μ Cdimm est une version plus performante du kit originel μ Csimm.

Le module μ Cdimm se présente sous la forme d' une barrette DIMM à 144 broches (1.7''x 2.7'').

Le module μ Csimm se compose de :

- Un microcontrôleur DragonBall VZ (68VZ328) à 33 MHz. Il intègre un contrôleur de DRAM, deux ports série, deux interfaces SPI, un contrôleur LCD (résolution jusqu'à 640x512), une sortie Timer et une sortie PWM.
- Jusqu'à 22 E/S parallèles.
- Une horloge Temps Réel – calendrier.
- De la mémoire dynamique DRAM de 8 Mo.
- De la mémoire FLASH de 2 Mo.
- Une interface Ethernet IEEE 802.3 10BaseT.

Les autres éléments du kit sont les même que ceux du kit μ Csimm. Le kit μ Cdimm est proposé par Lineo à 995 USD.

4.3. Le kit ARM7TDMI

Le kit ARM7TDMI proposé par Lineo ressemble fort aux précédents. Il est idéal pour développer des solutions embarquées à base de processeur ARM offrant plus de puissance CPU que les deux kits précédents.

Le kit est composé de la carte d'évaluation ATMEL EB40-LS à base du processeur ATMEL AT91 ARM7TDMI avec une carte fille (carte Lineo μ Cnet) pour la connectivité Internet.

L'ensemble forme une plateforme matérielle avec :

- Un processeur ARM7TDMI avec 128Ko de SRAM interne.
- De la mémoire statique SRAM de 2 Mo.
- De la mémoire FLASH de 128 Ko.
- Deux ports série.
- Un port JTAG.
- Une interface Ethernet 10BaseT IEEE 802.3.

On retrouve là aussi toute la documentation et les logiciels de développement nécessaires. Le kit ARM7TDMI est proposé par Lineo à 1495 USD.

4.4. La plateforme GPL : le projet OpenHardware

Le projet OpenHardware consiste à proposer des solutions libres de design de cartes électroniques comme il existe des logiciels libres.

Les schémas électroniques, les nomenclatures, les fichiers GERBER pour la fabrication du circuit imprimé et la notice de montage sont en libre accès. Il est à noter que OpenHardware n'a pas la prétention de proposer des cartes au design optimisé pour une production de masse mais propose plutôt un design pleinement testé comme point de départ à une reprise de CAO...

OpenHardware propose ainsi un design à base du microcontrôleur DragonBall EZ (68EZ328) supportant μ Clinux. Il se compose :

- D'une carte mère avec 4 sockets SIMM 72 broches dont 2 sont libres d'utilisation avec un connecteur RJ45 pour l'accès Ethernet IEEE 802.3 10BaseT, une interface pour écran LCD et une interface « *Touch Panel* ».
- D'un module SIMM 72 broches comportant :
 - Le processeur DragonBall EZ.
 - 8 Mo de DRAM.
 - 2, 4 ou 8 Mo de mémoire FLASH.
 - Horloge Temps Réel avec sauvegarde par pile bouton.
 - 9 E/S parallèles.
- D'un module SIMM 72 broches comportant le contrôleur Ethernet.

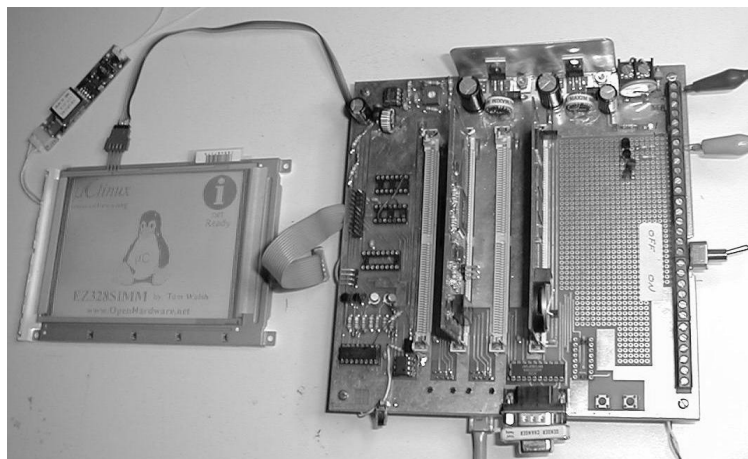


Figure 3 : Carte mère OpenHardware avec ses 2 modules SIMM processeur DragonBall EZ et contrôleur Ethernet

4.5. La plateforme ColdFire : la carte d'évaluation Motorola M5407C3

La plateforme matérielle choisie pour cet article est la carte d'évaluation Motorola M5407C3 à base de processeur ColdFire.

Les cartes d'évaluation proposées par les fabricants de composants sont à faible coût et sont un moyen rapide de tester un composant bien précis : convertisseur A/N, amplificateur opérationnel, processeur...

La distribution μ Clinux pour ColdFire supporte un ensemble de cartes d'évaluation du commerce comme :

- Les cartes Arnewsh SBC5206 et SBC5307.
- Les cartes Motorola M5206eLITE, M5206C3, M5307C3, M5272C3.
- Les cartes Lineo eLIA 5307, NETtel et SecureEdge.
- La carte Netburner CFV2-40.

La carte d'évaluation M5407C3 met en œuvre le processeur ColdFire MCF5407 et coûte environ 4000 Francs.

Le processeur ColdFire est le digne successeur du célèbre processeur Motorola 68000 dont la production s'est arrêtée avec le microprocesseur 68060 il y a quelques années. Il reprend le jeu d'instructions du microprocesseur 68020 où l'on a supprimé les instructions qui ne servaient guère ainsi que certains modes d'adressage. Suivant les versions de processeur ColdFire, Motorola a rajouté des instructions de type MAC (*Multiply and ACCumulate*) que l'on retrouve dans les processeurs de traitement du signal (*Digital Signal Processing*).

Il a aussi intégré des périphériques ; ce qui confère au processeur ColdFire les propriétés d'un microcontrôleur / DSP. Le marché visé par Motorola est clairement celui des télécommunications où l'on retrouve ce genre de besoins.

La carte d'évaluation M5407C3 possède :

- Un processeur MCF5407 à 150 MHz.
- 32 Mo de mémoire SDRAM.
- 2 Mo de mémoire FLASH dont 0,3 Mo occupé par le moniteur de la carte.
- 2 ports série.
- Une liaison Ethernet IEEE 802.3 10BaseT.
- Un port de debug BDM (*Background Debugger Module*).
- Un slot PCI.

5. MISE EN ŒUVRE DE μ CLINUX



La mise en œuvre de μ Clinux sera présentée pour le processeur ColdFire en utilisant comme plateforme matérielle la carte d'évaluation Motorola M5407C3. Le démarche à suivre sera bien sûr similaire pour μ Clinux porté sur d'autres processeurs...

La distribution μ Clinux/ColdFire est basée initialement sur un noyau Linux 2.0.38 mais propose aussi une version basée le noyau Linux 2.4.10.

L'ensemble est stable et offre une large palettes de services comme :

- Les systèmes de fichiers NFS et SMB côté client.
- Un client DHCP.
- L'IP masquerading.
- Des serveurs Web (boa et thttpd)
- ...

La mise en œuvre de μ Clinux/ColdFire pour la génération de l'image à télécharger dans la carte d'évaluation ressemble fort à une génération d'un noyau sous Linux.

Il faut d'abord récupérer les archives sur le site μ Clinux/ColdFire :

- de la chaîne de compilation croisée ColdFire (fichier m68k-elf-tools-xxxxxxx.tar.gz).
- de la distribution de μ Clinux/ColdFire (fichier uClinux-dist-xxxxxxx.tar.gz).

La première chose est d'installer sur le PC hôte (sous Linux) pour le développement croisé (*cross development*) la chaîne de compilation croisée ColdFire.

En étant root :

```
sh-2.04# cd /
sh-2.04# tar xvfz m68k-elf-tools-20010716.tar.gz
sh-2.04# exit
```

La deuxième chose est d'installer la distribution de μ Clinux/ColdFire dans son répertoire de travail en étant simple utilisateur :

```
sh-2.04$ tar xvfz uClinux-dist-20011112.tar.gz
sh-2.04$ cd uClinux-dist/
```

On retrouve ensuite l'enchaînement classique de compilation d'un noyau sous Linux :

```
sh-2.04$ make xconfig
```

Une série de fenêtres apparaît pour configurer le noyau μ Clinux/ColdFire et choisir ses applications :

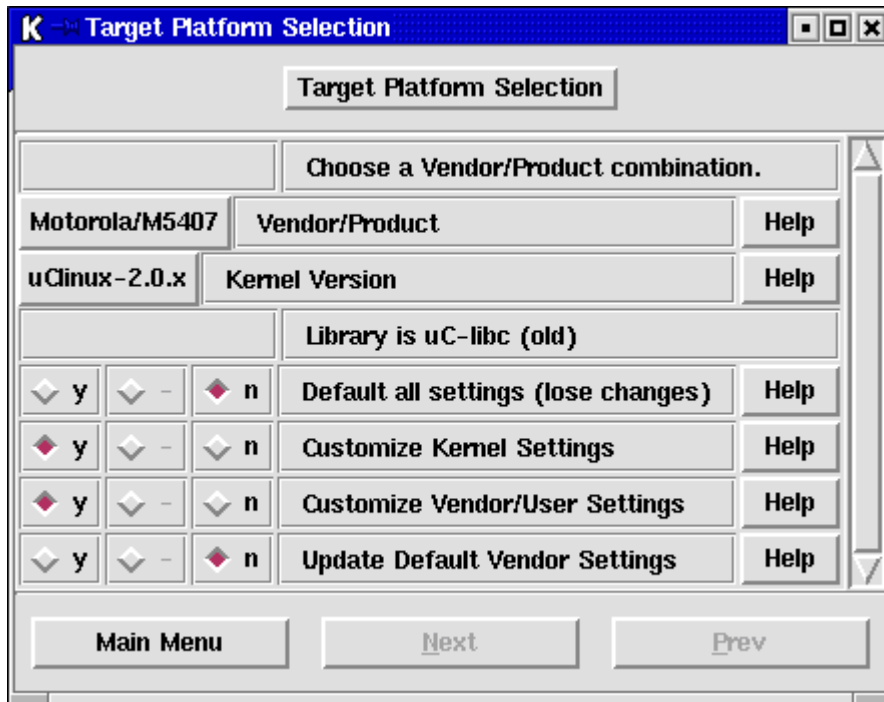


Figure 4 : Configuration du noyau μ Clinux/ColdFire

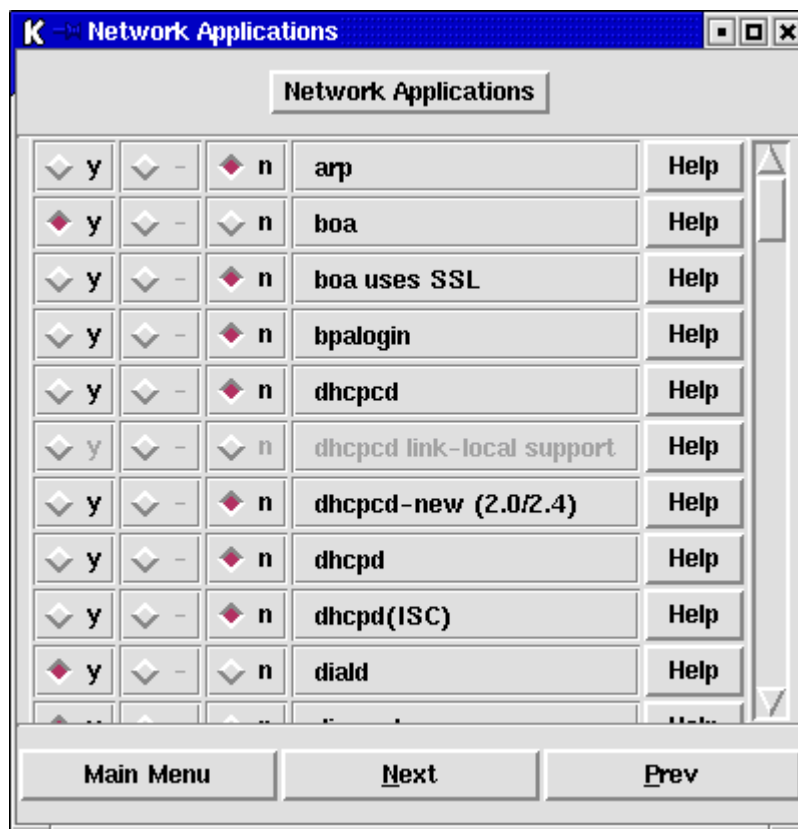


Figure 5 : Choix des applications (ici réseaux) μ Clinux/ColdFire

Puis, on retrouve ensuite la phase de compilation (croisée) :

```
sh-2.04$ make dep
sh-2.04$ make
```

Une image binaire du système μ Clinux/ColdFire *image.bin* est placée sous /tftpboot pour pouvoir être téléchargée en RAM de la carte d'évaluation via le réseau Ethernet avec le protocole TFTP.

Pour cela, on se connecte à la carte par le port série en utilisant l'application kermit ou minicom par exemple :

```
sh-2.04$ kermit
Connecting to /dev/ttyS1, speed 19200.
The escape character is Ctrl-\ (ASCII 28, FS)
Type the escape character followed by C to get back,
or followed by ? to see other options.
-----

Hard Reset
DRAM Size: 32M

Copyright 1995-1999 Motorola, Inc. All Rights Reserved.
ColdFire MCF5407 EVS Firmware v2e.1a.1a (Build 1 on Jul 27 2000 16:35:59)
Enter 'help' for help.

dBUG> dn image.bin
Eth Mac Addr is 00:CF:54:07:C3:01
Downloading Image 'image.bin' from 147.200.10.209
.....
1269540 bytes read via TFTP
```

Puis on lance le noyau μ Clinux/ColdFire :

```
dBUG> go 20000
uClinux/COLDFIRE(m5407)
COLDFIRE port done by Greg Ungerer, gerg@lineo.com
Flat model support (C) 1998,1999 Kenneth Albanowski, D. Jeff Dionne
Calibrating delay loop.. ok - 149.50 BogomIPS
Memory available: 30892k/32768k RAM, 0k/0k ROM (359k kernel code, 140k
data)
. . .
Execution Finished, Exiting
Sash command shell (version 1.1.1)
/>
```

On remarquera sur les traces précédentes la taille du noyau inférieure à 500 Ko !

On a alors maintenant un système Linux embarqué opérationnel et l'on peut lancer des applications comme par exemple le serveur Web embarqué boa :

```
/> boa&
[20]
/>
```

6. EXEMPLE D'APPLICATION μ CLINUX : CONNECTIVITE INTERNET PAR WWW

Un exemple d'application sous μ Clinux/ColdFire est la mise en place d'une connectivité Internet d'un équipement électronique (ici la carte d'évaluation) par WWW en utilisant l'interface CGI (*Common Gateway Interface*).

On utilise le serveur boa qui supporte l'interface CGI.

Le principe est d'écrire un programme source en langage C (fichier enseirb.c) que l'on placera dans la distribution μ Clinux/ColdFire sous `uClinux-dist/vendors/Generic/cgi/`. Il sera ensuite (cross)compilé pour générer l'exécutable CGI (fichier enseirb.cgi) et placé dans le système de fichiers de l'image μ Clinux sous `/home/httpd/`.

Le serveur boa a été configuré pour autoriser l'exécution des CGI.

Ce programme CGI allume ou éteint une led de la carte d'évaluation ColdFire en fonction de l'argument (0 ou 1) que l'on passe.

Le contrôle de la led sera fait en utilisant un navigateur Web avec une URL du style <http://adresse-ip-carte/enseirb.cgi?0> (led éteinte) ou <http://adresse-ip-carte/enseirb.cgi?1> (led allumée).

Le contrôle du matériel par WWW est limité ici au contrôle d'une led mais le principe reste le même pour le contrôle de toute autre matériel...

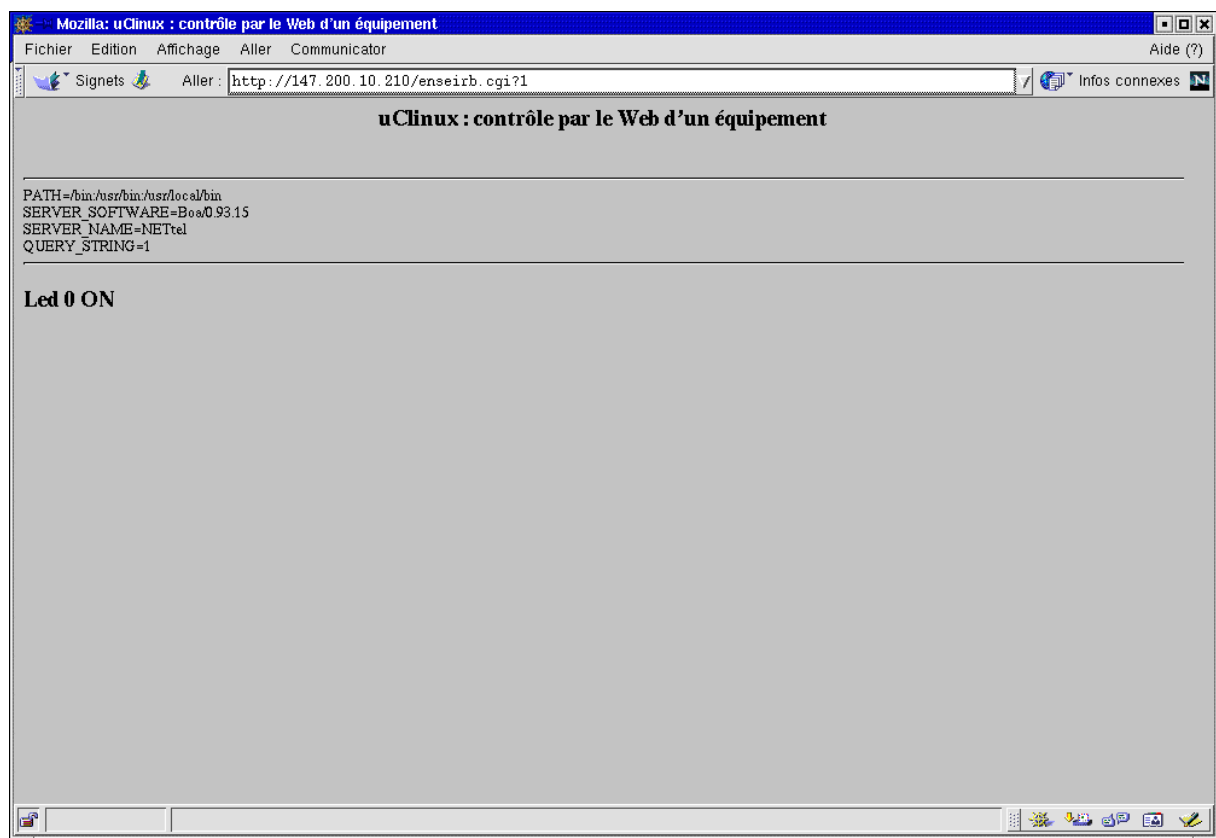


Figure 6 : Pilotage par le Web d'un équipement sous Linux embarqué μ Clinux

7. CONCLUSION. PERSPECTIVES

On a pu voir à travers cet article que Linux fait vraiment bon ménage avec les systèmes embarqués.

Le lecteur a pu découvrir la mise en œuvre de la distribution Linux embarqué μ Clinux/ColdFire sur les plans matériels et logiciels. Cette mise en œuvre est relativement simple pour un spécialiste Linux.

L'avenir de Linux embarqué est prometteur. On peut signaler particulièrement deux points très intéressants :

- Le mariage de Linux embarqué et du Système sur Silicium Programmable (*System On Programmable Chip*) qui permet de mettre en place une véritable offre de *codesign*. C'est chose faite avec l'offre processeur NIOS d'Altera sur composant programmable et le portage de μ Clinux sur ce processeur par la société Microtronix (voir bibliographie).
- L'utilisation de Linux embarqué sur les ordinateurs de poche (*handheld*). Ce domaine est très actif avec des offres Linux sur ordinateurs de poche Compaq iPAQ par exemple.

8. REFERENCES BIBLIOGRAPHIQUES

8.1. Articles

- [1] Linux embarqué. Linux Magazine, pp 16-18. Mai 2000.
- [2] Linux embarqué. Michel Stempin. Linux Magazine, pp 28-34. Novembre 2001.
- [3] Embedding Linux in a Commercial Product. Joel Williams. Linux Journal. Août 1999. <http://www.linuxjournal.com/lj-issues/issue66/3587.html>
- [4] Les distributions Linux industrielles. François Gauthier. Mesures, No 732, pp 74-78. Février 2001.
- [5] Les systèmes Linux embarqués. C. Gross. Electronique, No 116, pp 92-94. Juillet 2001.
- [6] Hardware : Le kit μ Csimm. Frédéric Navarro. Linux Focus. Mars 2000. <http://www.linuxfocus.org/Francais/March2000/article146.shtml>

8.2. Web

Linux et les systèmes embarqués :

- <http://www.linuxembedded.org>
- Page de l'auteur : <http://www.enseirb.fr/~kadionik/embedded/embedded.html>

Linux et le Temps Réel :

- RTLinux : <http://www.rtlinux.org>
- Mini RTLinux : <http://www.rtlinux.org/minirtl.html>
- RTAI : <http://www.rtai.com>
- Portail sur Linux et le Temps Réel : <http://www.realtimelinux.org>

Le projet μ Clinux :

- Le site officiel de μ Clinux : <http://www.uclinux.org/>
- Le site commercial de μ Clinux pour l'achat en ligne des kits (μ Csimm, μ Cdimm...) : <http://www.uclinux.com/>
- Le site OpenHardware : schémas électroniques en libre d'une carte MC68EZ328 sous μ Clinux : <http://www.openhardware.net/index.html>
- Site collectant tout sur μ Clinux : <http://home.at/uclinux/>
- Logiciels et matériels pour μ Clinux (MP3) : <http://www.uclinux.net/>
- μ Clinux Temps Réel : <http://www.uclinux.org/ucsimm/realtime.html>

- Serveur boa : <http://www.boa.org>

Portage μ Clinux sur 68K :

- Linux version 2.0.x : <http://www.uclinux.org/pub/uClinux/uClinux-2.0.x/>
- Linux version 2.4.x : <http://www.uclinux.org/pub/uClinux/uClinux-2.4.x/>
- Portage sur 68360 QUICC : <http://www.uclinux.org/pub/uClinux/ports/mc68360/>
- Portage sur Atari 68K : <http://www.esat.kuleuven.ac.be/~pcoene/atari.html>

Portage μ Clinux sur ColdFire :

- <http://www.uclinux.org/ports/coldfire/>
- Ressources sur le ColdFire : <http://fiddes.net/coldfire/>

Portage μ Clinux sur i960 :

- <http://www.uclinux.org/pub/uClinux/ports/i960/>

Portage μ Clinux sur ARM7TDMI :

- <http://www.uclinux.org/pub/uClinux/ports/arm7tdmi/>

Portage μ Clinux sur processeur soft NIOS d'Altera :

- La société Altera et son processeur RISC NIOS :
<http://www.altera.com/products/ip/altera/m-alt-nios.html>
http://www.altera.com/products/devkits/altera/kit-dev_linux.html et
- La société Microtronix proposant le portage μ Clinux ; <http://www.microtronix.com/>

Linux sur ordinateur de poche :

- Site de référence : <http://www.handhelds.org>
- Distribution Linux Familiar <http://familiar.handhelds.org>

Crédits photos : photos issues du site μ Clinux www.uclinux.org et du site OpenHardware www.openhardware.net