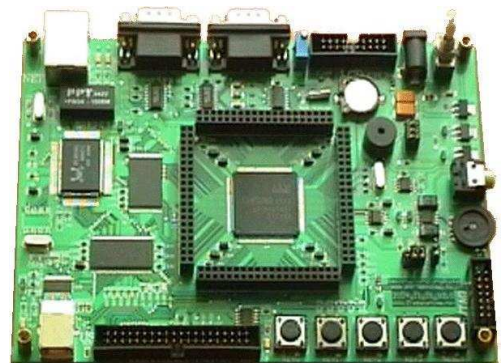


Pragmatec

Produits et services dédiés aux systèmes embarqués

uClinux Kernel

ARM7 Development Starter Kit





Bâtiment EARHART
ZAC Grenoble Air Parc
38590 St Etienne de St Geoirs - France
www.pragmatec.net

uClinux - Kernel



Bâtiment EARHART
ZAC Grenoble Air Parc
38590 St Etienne de St Geoirs - France
www.pragmatec.net

uClinux - Kernel

Le kit de développement ARM7 est un kit réalisé par la société PRAGMATEC S.A.R.L., société située à Grenoble (www.pragmatec.net). Il est basé sur une carte de développement ARM7, largement utilisée en Asie depuis de nombreuses années. Il s'agit donc d'un produit efficace, fiable et disponible.

Pragmatec s'est attaché à faire de ce kit un environnement de développement complet et immédiatement opérationnel, avec une introduction en français et le reste des documents et exemples en langue anglaise. En cas de difficultés techniques vous bénéficiez de plus d'un support technique de la part de l'équipe support de Pragmatec : support@pragmatec.net.

Ce document a pour but de démontrer la simplicité d'utilisation d'une telle plate-forme avec le système d'exploitation Linux, et tout particulièrement sa version embarquée : uClinux. Nous aborderons ici les détails internes du noyau (modifications, drivers, gestion mémoire, ...), la génération du shell et des programmes associés ainsi que la programmation de la carte.

Ce document est la propriété de la société PRAGMATEC S.A.R.L. Il ne peut être reproduit et distribué sans l'accord de cette société.



Bâtiment EARHART
ZAC Grenoble Air Parc
38590 St Etienne de St Geoirs - France
www.pragmatec.net

uClinux - Kernel

TABLE DES MATIERES

1	<i>Préambule</i>	5
	La carte de développement.....	5
	Station Linux ou Windows™.....	6
2	<i>L' espace d'adressage</i>	7
	Mapping mémoire	7
	Démarrage de la carte.....	8
3	<i>Modification du noyau uClinux</i>	10
	Compilation de uClinux.....	10
	Résultats de la compilation.....	14
	Test du nouveau noyau	16
	Sauvegarde en flash NOR.....	20
4	<i>Modification du système de fichiers</i>	22
	Présentation du système de fichiers.....	22
	Modification de la busybox.....	24
	Sauvegarde de romfs.img en flash NOR	27
5	<i>Rechargement du BIOS</i>	29
	Chargement du BIOS en flash NOR	29
	Chargement d'uClinux et du romfs.....	32

uClinux - Kernel

1 Préambule

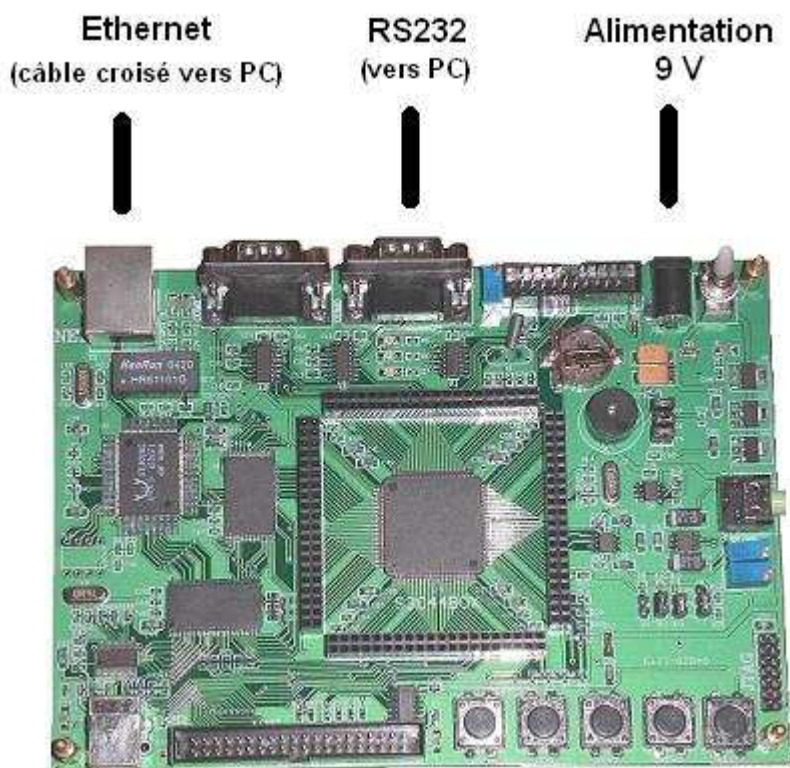


Le but de ce tutorial est de montrer simplement comment modifier le noyau uClinux de la distribution Pragmatec et comment télécharger sur la cible les nouvelles images ainsi créées. Il est recommandé de consulter au préalable le document « uClinux_tutorial_S3C44B0.pdf » pour bénéficier d'une installation complète des outils de développement et du paramétrage de « minicom ».

La carte de développement

La carte ARM7 qui vous est proposée est basée sur le processeur S3C44BOX de Samsung. Elle possède de nombreux périphériques (USB, Ethernet, RS232, IDE, ...) et ce tutorial n'a pas la prétention de vous former au noyau uClinux et à l'utilisation de ces drivers.

En plus de ce document et de votre carte de développement, vous aurez besoin d'une connexion RS232, d'un câble parallèle, d'un cordon Ethernet, de la sonde JTAG et de l'alimentation.



uClinux - Kernel

Station Linux ou Windows™

Dans ce document nous recommandons l'utilisation d'une station sous **Windows™** pour effectuer les transferts Ethernet vers la cible ou bien pour reflasher la carte avec la sonde JTAG. Toutefois de façon générale nous préconisons l'utilisation d'une station **Linux**. Cela signifie que nous développerons sur un PC Linux et que nous compilerons le noyau et les programmes utilisateurs pour notre cible S3C44B0.

N'importe quel PC sous Linux fera l'affaire, mais nous avons choisi de réaliser ce tutorial depuis la FedoraCore4, téléchargeable depuis le site de RedHat. Nous vous recommandons vivement d'utiliser cette distribution car l'environnement de développement Eclipse utilisé pour le debug y est déjà présent.



Un développement croisé depuis Linux est préférable par rapport à un développement depuis Windows/Cygwin. En effet, au fur et à mesure vous rajouterez des programmes et des services à votre carte ARM7 qui sont disponibles à l'identique sur votre PC Linux. N'hésitez pas à installer la FedoraCore4 sur votre PC déjà équipé de Windows sur une partition libre : un logiciel d'amorce sera installé afin de vous proposer au boot un choix quand au démarrage sous Windows ou Linux (logiciel GRUB).

Le noyau linux compilé pour votre cible est de la génération v2.4.
La chaîne de compilation GCC utilisée comme cross-compiler est la v2.95.3.

Enfin nous rappelons que le noyau destiné à la cible est un noyau uClinux. La différence majeure entre un noyau Linux et uClinux réside dans le fait que uClinux est une version dédiée aux processeurs qui ne bénéficient pas de la gestion de mémoire virtuelle (MMU) et d'unité de calcul à virgule flottante (FPU). Cette version est aussi appelée « NO_MMU / NO_FPU ». La librairie C nécessaire à la compilation des programmes utilisateurs a été aussi grandement allégée afin d'être plus efficace sur de petits systèmes embarqués.

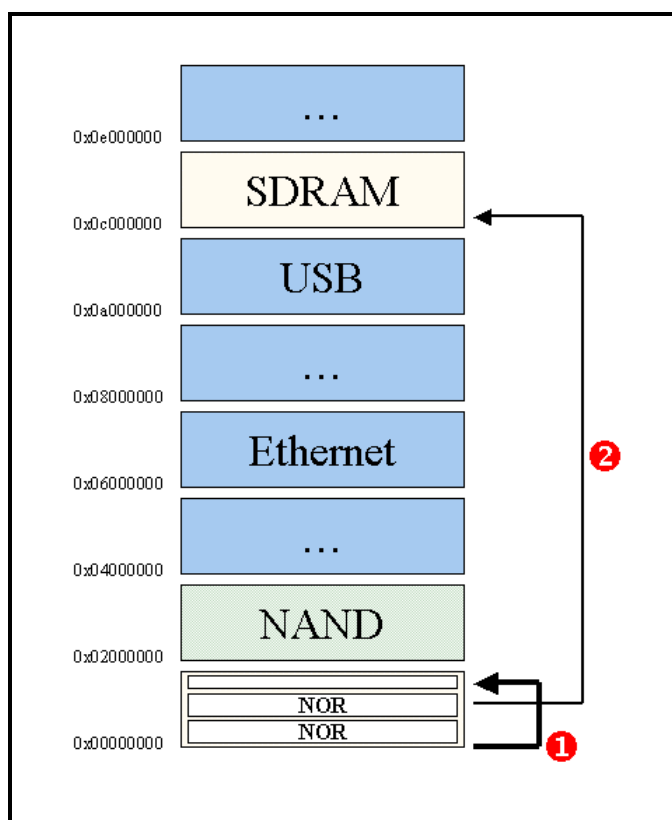


2 L' espace d'adressage

Ce chapitre présente la façon dont est répartie l'espace mémoire et présente de façon succincte la séquence de démarrage de la carte de développement.

Mapping mémoire

Le processeur S3C44B0 possède 8 banks de 32 Mo chacune. Chaque bank permet de gérer de façon automatique les accès à un périphérique. De ce fait les signaux CS (ChipSelect), WE (WriteEnable), OE (OutputEnable) ainsi que les bus de données et d'adresse sont automatiquement positionnés lorsque que vous accédez à n'importe quelle adresse en mémoire :



Le processeur démarre tout naturellement à l'adresse 0x00000000, donc il n'est pas surprenant de retrouver la flash NOR (qui contient le BIOS) sur la bank 0.

La mémoire RAM (SDRAM dans notre cas) est située en bank 6 et la mémoire flash NAND (pour les extensions du filesystem) est en bank 1 lorsqu'elle est présente.

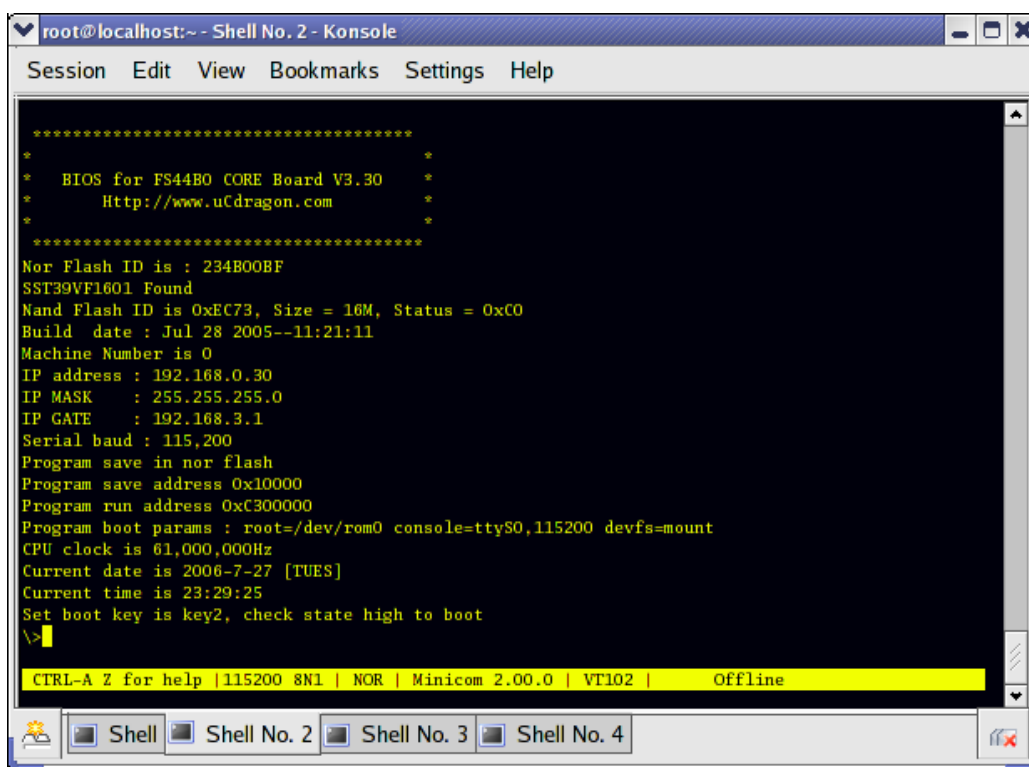
D'autres périphériques comme les contrôleurs Ethernet (RTL8019) ou USB (SL811) occupe une bank spécifique pour des raisons de confort et de performances, même si l'espace de 32Mo n'est pas requis pour des contrôleurs de protocole.

uClinux - Kernel

Démarrage de la carte

La bank 0 est donc la première concernée par le démarrage du processeur S3C44B0. Elle contient le code qui est joué en premier par le CPU et dont la fonction première est la prise en charge de l'ensemble des périphériques : le BIOS (Basic Input and Output Software).

Démarrez la carte en gardant le doigt appuyé sur le bouton 2 (le second en partant de la gauche) afin d'interdire le lancement de uClinux. Le BIOS affiche les périphériques détectés :



```
root@localhost:~ - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
*****
*          *
*  BIOS for FS44B0 CORE Board V3.30  *
*  Http://www.uCdragon.com          *
*          *
*****
Nor Flash ID is : 234B00BF
SST39VF1601 Found
Nand Flash ID is 0xEC73, Size = 16M, Status = 0xC0
Build date : Jul 28 2005--11:21:11
Machine Number is 0
IP address : 192.168.0.30
IP MASK    : 255.255.255.0
IP GATE    : 192.168.3.1
Serial baud : 115,200
Program save in nor flash
Program save address 0x10000
Program run address 0xC300000
Program boot params : root=/dev/rom0 console=ttyS0,115200 devfs=mount
CPU clock is 61,000,000Hz
Current date is 2006-7-27 [TUES]
Current time is 23:29:25
Set boot key is key2, check state high to boot
\>
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.00.0 | VT102 | Offline
```

En fait, le BIOS ne se trouve pas à proprement parlé à l'adresse 0x00000000, mais tout à la fin de la flash NOR. Quelques lignes en début de flash NOR permette au processeur de faire le saut vers la fin de la mémoire NOR et d'exécuter le code du BIOS. Ceci correspond à la phase ① indiquée en rouge sur le schéma.

Comme vous avez gardé le doigt appuyé sur le bouton 2, le BIOS s'arrête et ne lance pas l'application sauvegardée en flash NOR (uClinux dans notre cas).

Si vous resetez la carte sans utiliser le bouton 2, le BIOS va recopier le programme uClinux de la flash NOR vers la mémoire SDRAM. Uclinux va alors automatiquement se décompresser (l'image de uClinux étant un zip auto-décompactable) pour finalement s'exécuter. Ceci correspond à la phase ② sur le schéma.

uClinux - Kernel

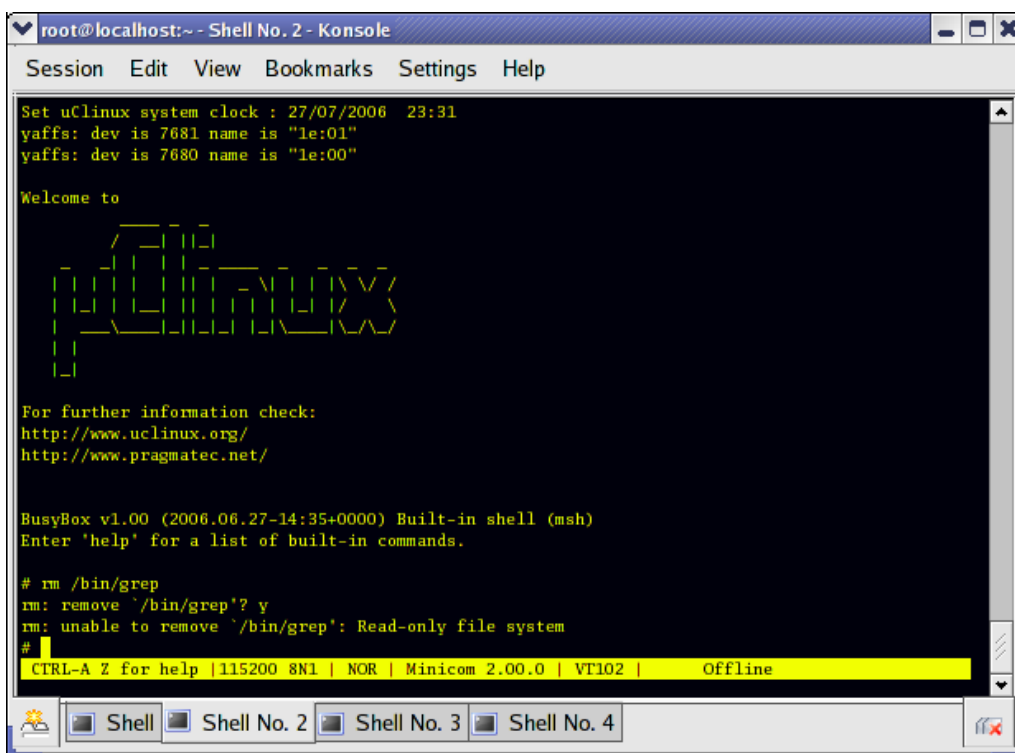
Pour ce faire la flash NOR est séparée en 2 zones :

- La zone haute qui contient l'image du filesystem en « read only »
- La zone basse qui contient l'image uClinux

Linux comme uClinux est un OS basé sur un système de fichiers (filesystem) où toute interface est considérée comme un fichier. Par exemple, écrire dans une clef USB insérée dans le premier slot USB revient à écrire dans le fichier « /dev/sda1 ». Ceci peut paraître étrange pour une personne étrangère à cet OS mais cela a le mérite d'homogénéiser l'ensemble. Ainsi pour connaître le nombre d'interruption qui sont apparus depuis le démarrage de la carte, il suffit d'afficher le contenu du fichier « /proc/interrupts », fichier géré par l'OS.

UClinux étant destiné à l'embarqué, il doit être particulièrement robuste. Par exemple on ne peut pas admettre que la carte ne puisse pas redémarrer parce qu'une opération a endommagé le système de fichiers !

Par conséquent uClinux utilise un système de fichier en « read only » pour lui permettre de démarrer. Impossible par la suite de modifier cette partie du système de fichiers. Il est donc nécessaire d'utiliser des extensions au système de fichiers (IDE, SDCARD, USBkey, flash NAND, ...) pour pouvoir y stocker vos applications. Le système de fichiers en lecture seule utilisé pour le démarrage d'uClinux se trouve dans la partie haute de la flash NOR.



```
root@localhost:~ - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
Set uclinux system clock : 27/07/2006 23:31
yaffs: dev is 7681 name is "1e:01"
yaffs: dev is 7680 name is "1e:00"

Welcome to

  _ _ _ _ _
 / _ _ _ _ \
|_| |_| |_|_|_|
|_| |_| |_|_|_|
|_| |_| |_|_|_|
|_| |_| |_|_|_|
|_| |_| |_|_|_|
|_| |_| |_|_|_|
|_| |_| |_|_|_|
|_| |_| |_|_|_|
|_| |_| |_|_|_|
|_| |_| |_|_|_|
|_| |_| |_|_|_|
|_| |_| |_|_|_|
|_| |_| |_|_|_|

For further information check:
http://www.uclinux.org/
http://www.pragmatec.net/

BusyBox v1.00 (2006.06.27-14:35+0000) Built-in shell (msh)
Enter 'help' for a list of built-in commands.

# rm /bin/grep
rm: remove `/bin/grep'? y
rm: unable to remove `/bin/grep': Read-only file system
#

CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.00.0 | VT102 | Offline
```

Comme le montre la capture d'écran ci-dessus il n'est pas possible de supprimer des programmes du répertoire /bin, car il fait parti des répertoires du système de fichiers en lecture seule.

uClinux - Kernel



3 Modification du noyau uClinux

Nous allons à présent vous montrer comment modifier le noyau uClinux pour votre cible et aussi comment mettre à jour votre carte avec le nouveau noyau.

Nous supposons par ailleurs que votre chaîne de développement est installée comme indiqué dans le document « uClinux_tutorial_s3c44B0.pdf » et que vous avez déjà effectué une première re-compilation du noyau.

Compilation de uClinux

Les fichiers de paramètres du noyau et des applications sont déjà présents dans votre distribution et se nomment :

- ✓ **./DEVII-step1** : sélection de la cible et de la libc
- ✓ **./linux-2.4.x/DEVII-step2** : paramétrage du noyau
- ✓ **./config/DEVII-step3** : sélection des applications

La procédure pour générer le noyau uClinux est la suivante :

```
xavier@localhost:~/uClinux-pragm-20060217 - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
[xavier@localhost ~]$ cd uClinux-pragm-20060217/
[xavier@localhost uClinux-pragm-20060217]$ make clean
[xavier@localhost uClinux-pragm-20060217]$ make xconfig
[xavier@localhost uClinux-pragm-20060217]$ make dep
[xavier@localhost uClinux-pragm-20060217]$ make
[xavier@localhost uClinux-pragm-20060217]$
```

Le « make clean » n'est pas indispensable, bien au contraire vous n'avez normalement pas à l'utiliser sauf si vous ne parvenez pas à compiler le noyau et qu'il vous faille repartir entièrement de 0.

Le « make xconfig » vous permet de sélectionner graphiquement les éléments de votre noyau, nous y reviendrons par la suite.

Le « make dep » est nécessaire si vous avez ajouté de nouveau périphérique et dans tous les cas à la suite d'un « make clean ». Dans la plupart des cas il n'est pas indispensable.

uClinux - Kernel

Le « make » est bien entendu la commande que vous utiliserez inmanquablement pour reconstruire le noyau. Ce « make » est un appel général à plusieurs construction spécifique. Ainsi vous pourriez juste lancer « make lib » ou encore « make user » pour ne compiler que ce qui est nécessaire.

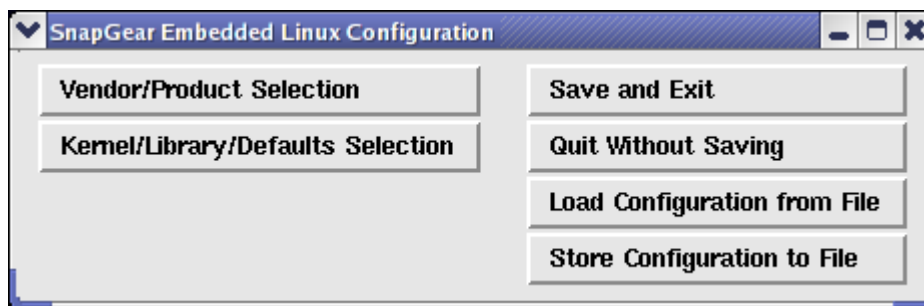
Toutefois nous vous conseillons d'utiliser « make » : dans ce cas la construction sera complète mais les librairies qui n'ont pas besoin d'être reconstruites ne le seront pas.

Examinons à présent les 3 grandes étapes de la construction du noyau :

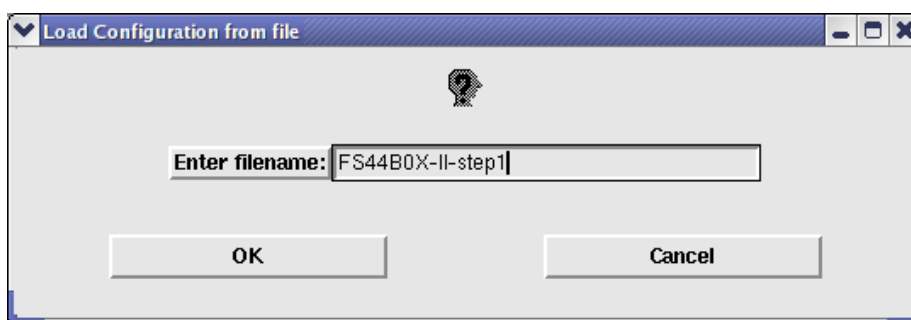
- Sélection de la cible
- Paramétrage du noyau et sélection des périphériques
- Sélection des applications embarquées

Lancez la commande « make xconfig » pour débiter la reconstruction du noyau.

La première fenêtre est la fenêtre principale de sélection de la cible :

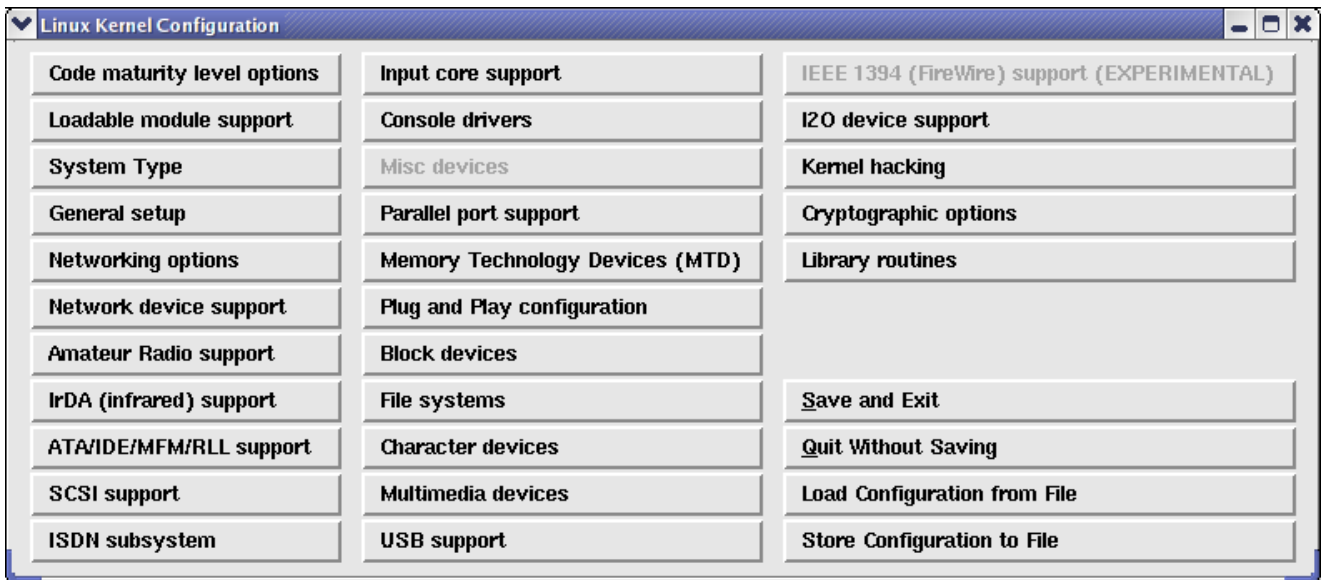


Cliquez sur « **Load Configuration from File** » pour charger la configuration correspondante au step1 (par exemple **DEVII-step1**).



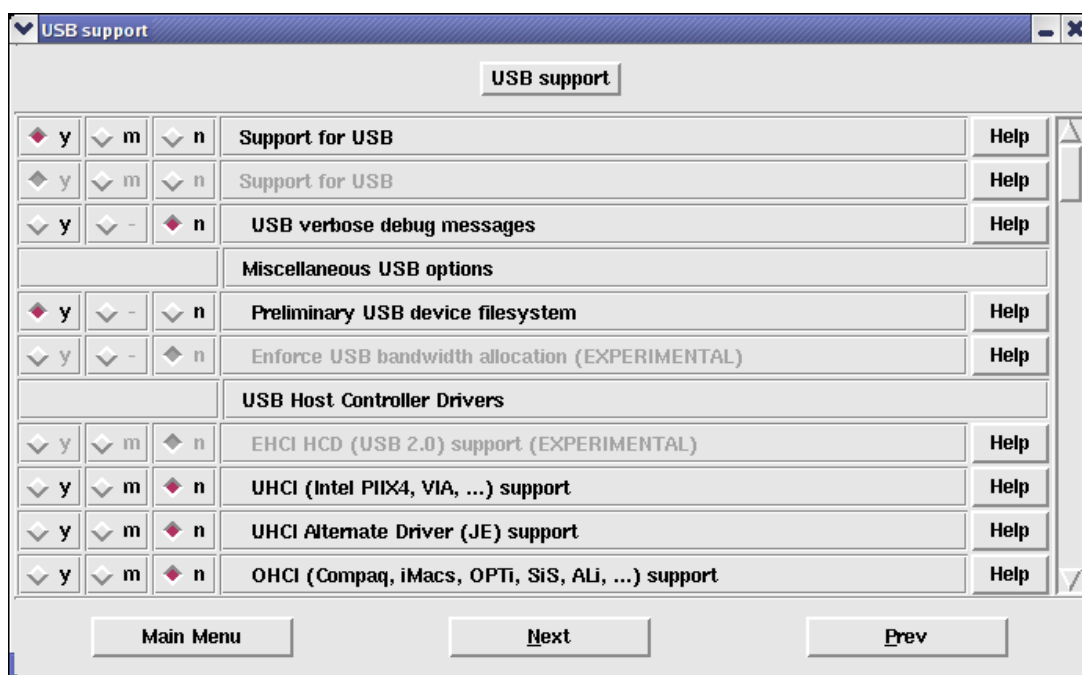
uClinux - Kernel

Cliquez sur **OK** pour valider la sélection, puis « **Save and Exit** ». La seconde fenêtre principale apparaît à l'écran :



Cliquez à nouveau sur « **Load Configuration from File** » et chargez le fichier **DEVII-step2**.

Cette fenêtre est la fenêtre principale de modification du noyau. Vous y trouverez les paramètres des différents périphériques. Par exemple si vous ne souhaitez pas activer le périphérique **USB HOST** pour bénéficier d'un noyau plus petite et plus efficace, cliquez sur le bouton « USB support » :

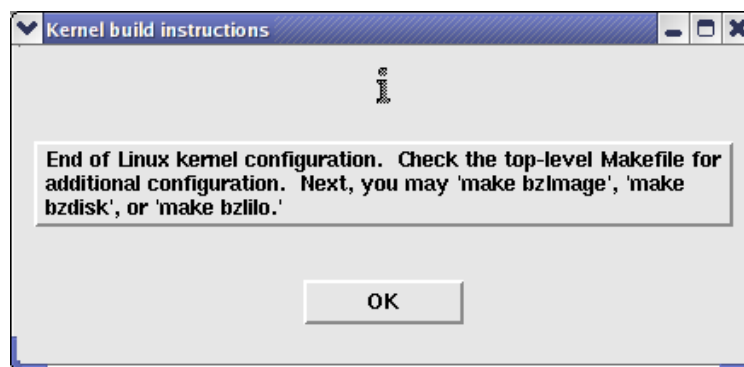


uClinux - Kernel

Sur la première ligne se trouve l'activation et la désactivation des drivers du périphérique **USB host**. « y » signifie que les drivers sont compilés avec le noyau, le « n » signifie que vous souhaitez pas que ces drivers soient associés au noyau, et enfin le « m » signifie que vous compilez ces drivers sous la forme de modules (ils sont compilés mais pas linkés avec le noyau) afin d'être chargés dynamiquement lors de l'exécution du noyau à l'aide de la commande « insmod ».

Cliquez sur « **Main Menu** » et enfin sur « **Save and Exit** ».

La fenêtre suivante apparaît. Il s'agit de la fenêtre de fin de la configuration de Linux. Elle précise les actions à mener pour générer correctement le noyau linux. Dans notre cas il ne s'agit pas réellement de la dernière fenêtre principale.



Ne tenez pas compte des informations qui sont indiquées sur cette fenêtre. Cliquez sur **OK** pour fermer cette fenêtre.

La 3^{ème} fenêtre principale apparaît alors pour la sélection de la partie applicative :



Cliquez sur « **Load Configuration from File** » est chargé le fichier **DEVII-step3**.

Vous pouvez terminer le paramétrage de votre système Linux embarqué en cliquant sur « **Save and Exit** ».

Vous devez à nouveau retrouver le shell qui vous a permis de lancer un « make xconfig ».

Tapez la commande « **make dep** » puis la commande « **make** » pour compiler le noyau... plus exactement pour ne compiler que les fichiers qui le nécessitent réellement. S'il ne s'agit donc pas de votre première compilation, cette étape devrait prendre moins d'une minute.

uClinux - Kernel

Résultats de la compilation

La compilation du noyau depuis la commande « make » se termine par 4 grandes étapes :

- ✓ **Compilation du noyau** : compilation et édition de lien des fichiers sélectionnés dans chacun des sous-répertoires du répertoire linux-2.4.x.
- ✓ **Compilation des programmes** : compilation des programmes sélectionnés lors du « step3 ».
- ✓ **Compilation du système de fichiers** : les programmes sélectionnés pendant le « step3 » sont regroupés afin de former le système de fichiers en lecture seule.
- ✓ **Génération des images** : le noyau uClinux créé est compacté sous forme d'un fichier binaire appelé « image ». 2 images correspondent au noyau uClinux : une pour une exécution en SDRAM et l'autre pour une écriture en flash NOR. Le système de fichiers est lui aussi compacté sous forme d'une image qui lui est propre.

```
xavier@localhost:~/uClinux-pragm-20060217 - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
1 sbin [0x305, 0x1d80bc] 0120777, sz 4, at 0xeac40
1 var [0x305, 0x1dd34b] 0040775, sz 0, at 0xeac70
2 .. [0x305, 0x1d9864] 0040775, sz 0, at 0xeac90 [link to 0x20 ]
2 .. [0x305, 0x1dd34b] 0040775, sz 0, at 0xeacb0 [link to 0xeac70 ]
1 usr [0x305, 0x1dd34a] 0040775, sz 0, at 0xeacd0
2 .. [0x305, 0x1d9864] 0040775, sz 0, at 0xeacf0 [link to 0x20 ]
2 .. [0x305, 0x1dd34a] 0040775, sz 0, at 0xead10 [link to 0xeacd0 ]
cp /home/xavier/uClinux-pragm-20060217/tools/tftp /home/xavier/uClinux-pragm-20060217/images
arm-elf-ld -r -o /home/xavier/uClinux-pragm-20060217/linux-2.4.x/romfs.o -b binary /home/xavier/uClinux-pragm-20060217/images/romfs.img
arm-elf-objcopy -O binary -R .note -R .comment -S /home/xavier/uClinux-pragm-20060217/linux-2.4.x/linux /home/xavier/uClinux-pragm-20060217/images/image.ram
cp /home/xavier/uClinux-pragm-20060217/linux-2.4.x/arch/armnommu/boot/zImage /home/xavier/uClinux-pragm-20060217/images/image.rom
make[1]: Leaving directory '/home/xavier/uClinux-pragm-20060217/vendors/Samsung/44BOX'
[xavier@localhost uClinux-pragm-20060217]$ ll images/
total 3600
-r-xr-xr-x 1 xavier xavier 60 Feb 20 01:37 BOOT.BIN
-rwxrwxr-x 1 xavier xavier 1822540 Feb 20 11:36 image.ram
-rwxr-xr-x 1 xavier xavier 837688 Feb 20 11:36 image.rom
-rw-rw-r-- 1 xavier xavier 962560 Feb 20 11:36 romfs.img
-rwxrwxr-x 1 xavier xavier 24951 Feb 20 11:36 tftp
[xavier@localhost uClinux-pragm-20060217]$
```

Pour vous en convaincre, tapez les commandes suivante indépendamment afin de mieux apprécier la portée de chaque étape de compilation :

« **make user** » : pour construire les programmes du « step3 ». Chaque programme correspond à un sous-répertoire du répertoire « user » de la distribution, et les programmes résultant de la compilation se trouvent dans leur sous-répertoire respectif.

uClinux - Kernel

« **make romfs** » : pour générer le système de fichier en lecture seule. Vous pouvez vérifier le système de fichiers ainsi créé en allant sous le répertoire « romfs » de la distribution.

« **make image** » : pour créer les images du noyau et du système de fichier.

Astuce :

Il peut être très pratique de *court-circuiter* le « make » général, par exemple si vous voulez supprimer des programmes du système de fichiers en lecture seule et les remplacer par votre propre programme : il vous suffit alors de supprimer les programmes indésirables sous « romfs/bin » puis de copier votre programme au même endroit. Vous n'aurez alors qu'à régénérer les images, sans avoir eu recours à une quelconque compilation !

uClinux - Kernel

Test du nouveau noyau

Parmi les fichiers images créés sous le répertoire « images » de la distribution, il existe 2 fichiers correspondant au noyau uClinux :

- ✓ **image.ram** : noyau pouvant être stocké en SDRAM exécuté en SDRAM.
- ✓ **image.rom** : noyau pouvant être stocké en flash NOR et exécuté en SDRAM.

Plutôt que d'effacer l'image du noyau en flash NOR, nous allons tout d'abord tester le nouveau noyau dans la mémoire RAM de la carte.

Resetez la carte en gardant le doigt appuyé sur le bouton 2 afin d'inhiber le lancement d'uClinux (ou bien appuyer sur ENTER selon le BIOS installé sur la carte).

Vous devez obtenir le prompt du BIOS :

```
root@localhost:~ - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help
*****
*
*   BIOS for FS44B0 CORE Board V3.30   *
*   Http://www.uCdragon.com           *
*
*****
Nor Flash ID is : 234B00BF
SST39VF1601 Found
Nand Flash ID is 0xEC73, Size = 16M, Status = 0xC0
Build date : Jul 28 2005--11:21:11
Machine Number is 0
IP address : 192.168.0.30
IP MASK    : 255.255.255.0
IP GATE    : 192.168.3.1
Serial baud : 115,200
Program save in nor flash
Program save address 0x10000
Program run address 0xC300000
Program boot params : root=/dev/rom0 console=ttyS0,115200 devfs=mount
CPU clock is 61,000,000Hz
Current date is 2006-7-27 [TUES]
Current time is 23:29:25
Set boot key is key2, check state high to boot
\>
```

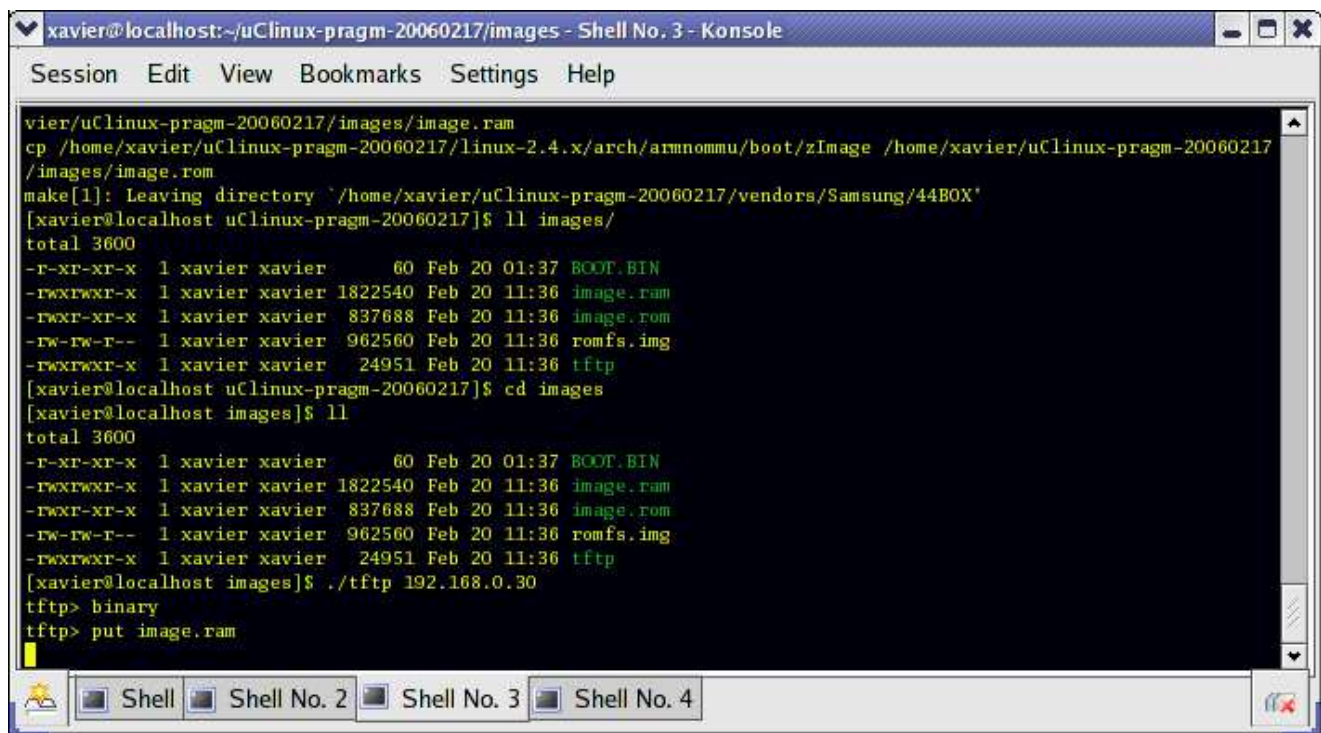

uClinux - Kernel

Pour charger votre binaire « image.ram » en RAM (qui contient le nouveau noyau uClinux recompilé), vous devez utiliser la commande « **netload** » et « **run** » du BIOS comme indiqué sur l'image suivante. Le fichier « image.ram » sera automatiquement chargé en mémoire RAM à l'adresse 0x0C008000, qui est une adresse physique de la mémoire SDRAM.

Vous ne risquez strictement rien à charger un nouveau noyau uClinux en RAM. Si vous rencontrez le moindre problème vous aurez toujours la possibilité de rebooter votre carte (bouton reset) et laisser la carte démarrer d'elle même depuis le noyau uClinux flashé dans la mémoire NOR.

Pour charger un binaire en mémoire RAM, resetez la cible et tapez la commande « **netload** ».

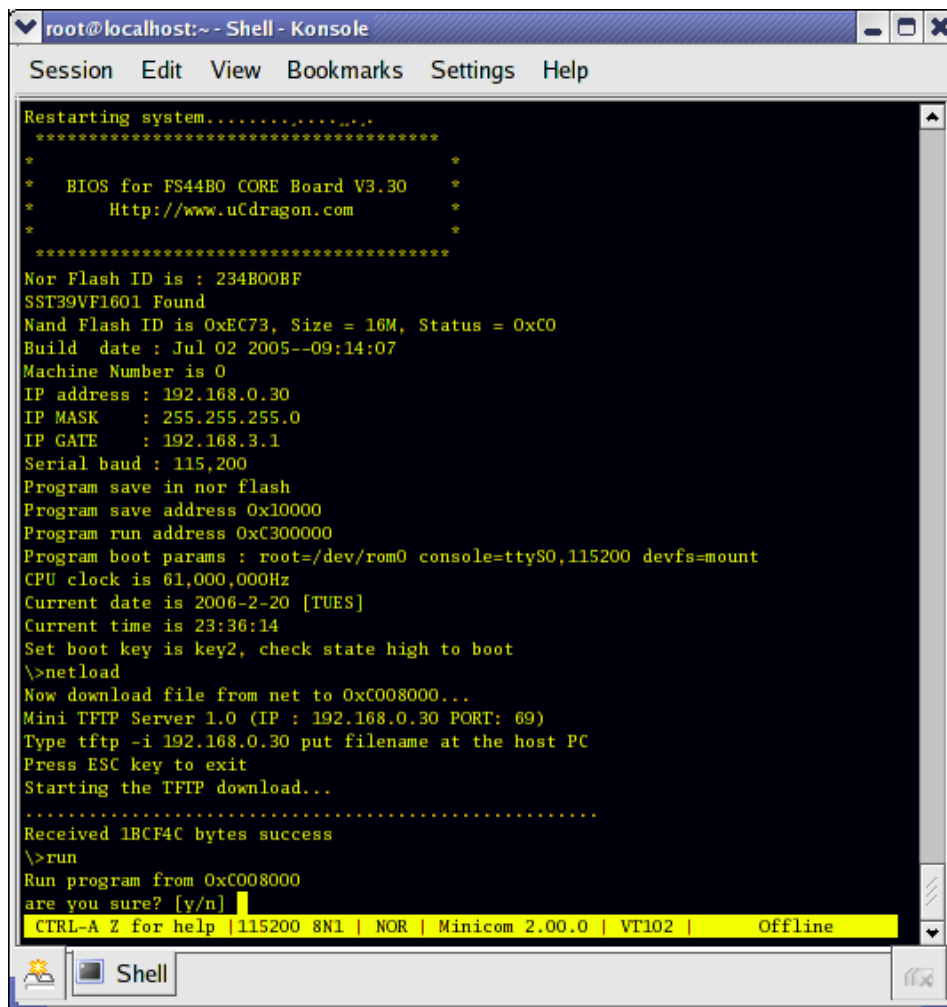
Sur la station de développement, allez sous le répertoire « images » et tapez « **./tftp 192.168.0.30** », puis transférer comme suit votre binaire vers la cible :



```
xavier@localhost:~/uClinux-pragm-20060217/images - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
vlier/uClinux-pragm-20060217/images/image.ram
cp /home/xavier/uClinux-pragm-20060217/linux-2.4.x/arch/armnommu/boot/zImage /home/xavier/uClinux-pragm-20060217
/images/image.ram
make[1]: Leaving directory `/home/xavier/uClinux-pragm-20060217/vendors/Samsung/44BOX'
[xavier@localhost uClinux-pragm-20060217]$ ll images/
total 3600
-r-xr-xr-x 1 xavier xavier 60 Feb 20 01:37 BOOT.BIN
-rwxrwxr-x 1 xavier xavier 1822540 Feb 20 11:36 image.ram
-rwxr-xr-x 1 xavier xavier 837688 Feb 20 11:36 image.rom
-rw-rw-r-- 1 xavier xavier 962560 Feb 20 11:36 romfs.img
-rwxrwxr-x 1 xavier xavier 24951 Feb 20 11:36 tftp
[xavier@localhost uClinux-pragm-20060217]$ cd images
[xavier@localhost images]$ ll
total 3600
-r-xr-xr-x 1 xavier xavier 60 Feb 20 01:37 BOOT.BIN
-rwxrwxr-x 1 xavier xavier 1822540 Feb 20 11:36 image.ram
-rwxr-xr-x 1 xavier xavier 837688 Feb 20 11:36 image.rom
-rw-rw-r-- 1 xavier xavier 962560 Feb 20 11:36 romfs.img
-rwxrwxr-x 1 xavier xavier 24951 Feb 20 11:36 tftp
[xavier@localhost images]$ ./tftp 192.168.0.30
tftp> binary
tftp> put image.ram
```

uClinux - Kernel

Côté cible, vous devez obtenir :



```
root@localhost:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

Restarting system.....
*****
*
*   BIOS for FS44B0 CORE Board V3.30   *
*   Http://www.uCdragon.com           *
*
*****
Nor Flash ID is : 234B00BF
SST39VF1601 Found
Nand Flash ID is 0xEC73, Size = 16M, Status = 0xC0
Build date : Jul 02 2005--09:14:07
Machine Number is 0
IP address : 192.168.0.30
IP MASK   : 255.255.255.0
IP GATE   : 192.168.3.1
Serial baud : 115,200
Program save in nor flash
Program save address 0x10000
Program run address 0xC300000
Program boot params : root=/dev/rom0 console=ttyS0,115200 devfs=mount
CPU clock is 61,000,000Hz
Current date is 2006-2-20 [TUES]
Current time is 23:36:14
Set boot key is key2, check state high to boot
\>netload
Now download file from net to 0xC008000...
Mini TFTP Server 1.0 (IP : 192.168.0.30 PORT: 69)
Type tftp -i 192.168.0.30 put filename at the host PC
Press ESC key to exit
Starting the TFTP download...
.....
Received 1BCF4C bytes success
\>run
Run program from 0xC008000
are you sure? [y/n]
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.00.0 | VT102 | Offline
```

Vous n'avez plus qu'à taper la touche « y » de votre clavier pour valider le choix d'exécuter le programme chargé en SDRAM à l'adresse 0x0C008000.

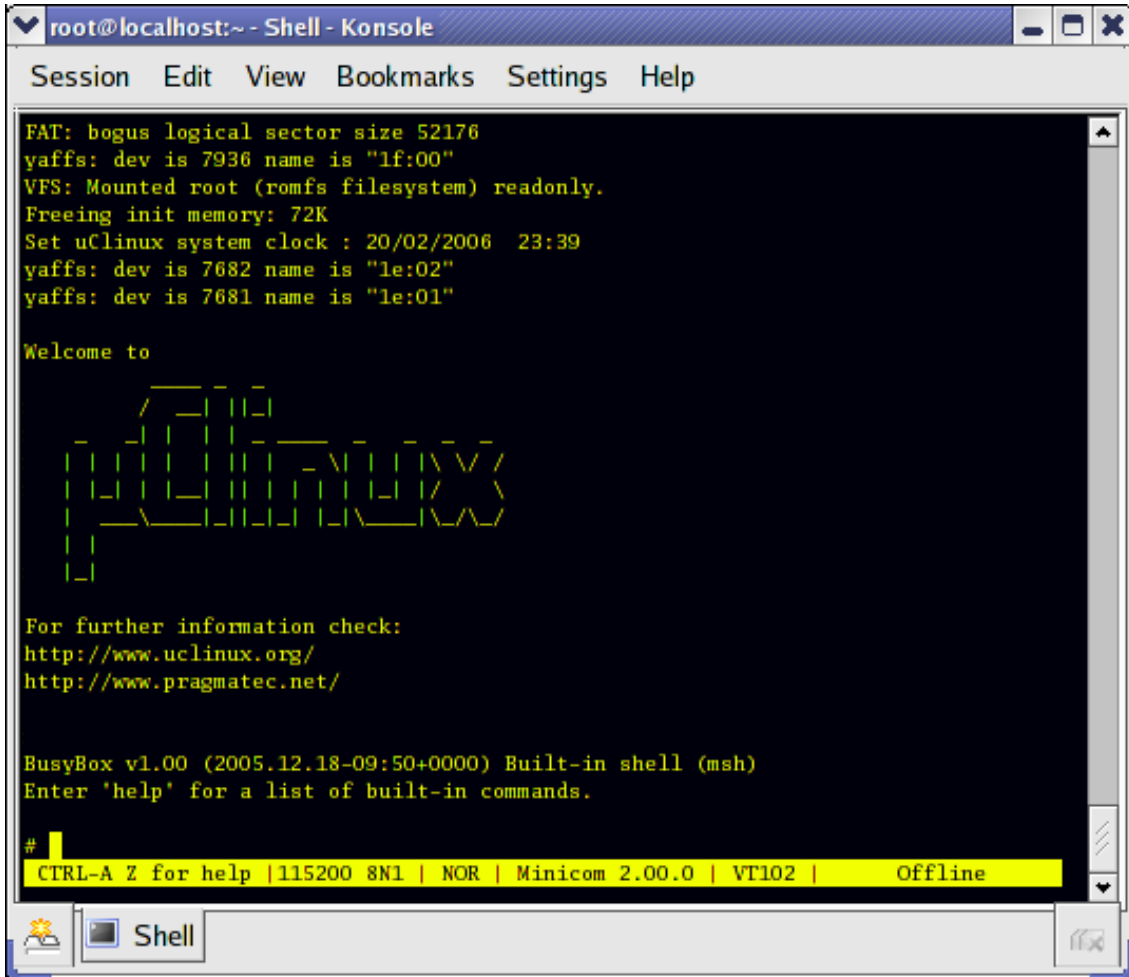
Durant le démarrage de uClinux vous verrez un certain nombre d'informations s'afficher sur votre écran. Si vous avez ajouté ou supprimé la gestion des disques IDE ou de l'USB par exemple, les messages correspondant devraient être présents ou non à l'écran.

Astuce :

Vous pouvez faire un test simple si vous disposez un LCD couleur Pragmatec connecté à votre carte de développement : désélectionnez l'utilisation du « framebuffer » (bouton console drivers du « step2 »). Avec le nouveau noyau, l'écran de démarrage aura disparu alors qu'avec le noyau présent en flash NOR, le logo Pragmatec apparaît au centre de l'écran.

uClinux - Kernel

A l'issu du démarrage du nouveau noyau, le logo uClinux et un prompt doivent apparaître.



```
root@localhost:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

FAT: bogus logical sector size 52176
yaffs: dev is 7936 name is "1f:00"
VFS: Mounted root (romfs filesystem) readonly.
Freeing init memory: 72K
Set uclinux system clock : 20/02/2006 23:39
yaffs: dev is 7682 name is "1e:02"
yaffs: dev is 7681 name is "1e:01"

Welcome to

      /_ _ _ \
     /  _ _  \
    / _ _ _ \
   / _ _ _ \
  / _ _ _ \
 / _ _ _ \
/_ _ _ _ \
 \_ _ _ _ /
  \_ _ _ /
   \_ _ /
    \_ /
     /
    /

For further information check:
http://www.uclinux.org/
http://www.pragmatec.net/

BusyBox v1.00 (2005.12.18-09:50+0000) Built-in shell (msh)
Enter 'help' for a list of built-in commands.

#
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.00.0 | VT102 | Offline
```

uClinux - Kernel

Sauvegarde en flash NOR

Avant toute chose, nous vous recommandons vivement de tester la version RAM de votre nouveau noyau avant d'écraser le noyau présent en flash NOR. Quoiqu'il en soit, et même si le nouveau noyau généré ne se décompresse pas correctement en RAM, il est toujours possible de charger le noyau de référence présent sur le CDROM.

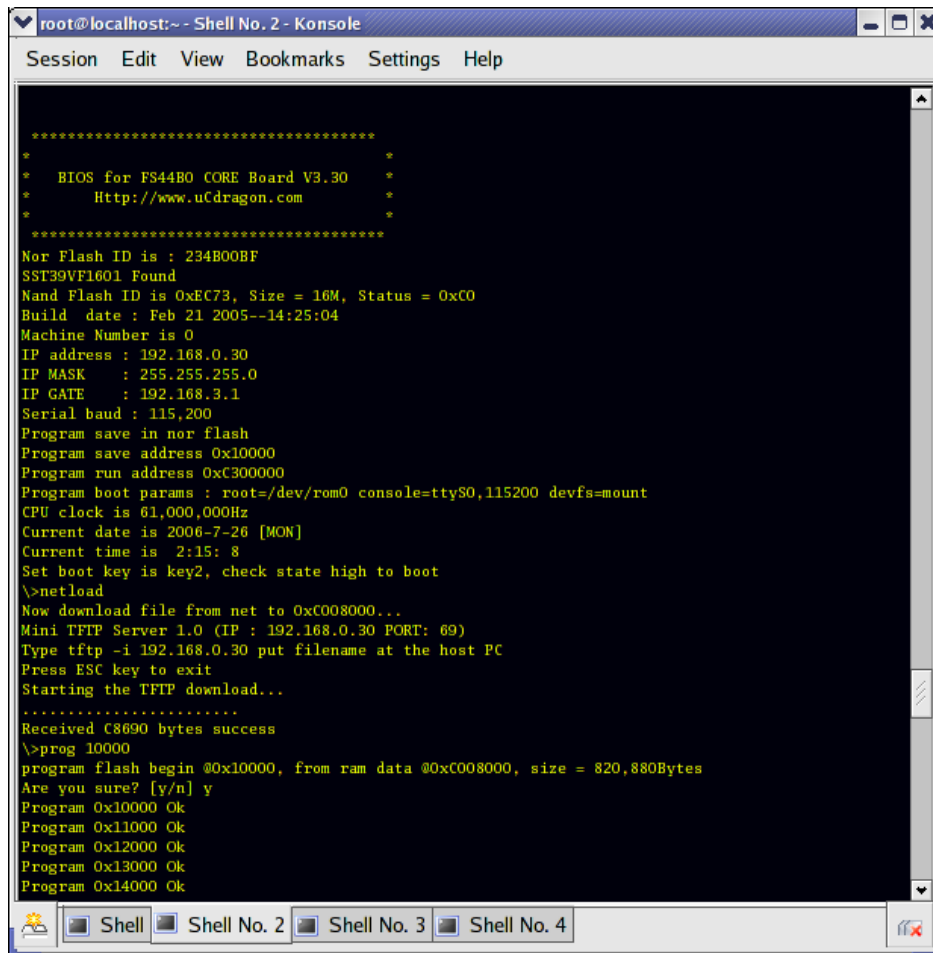
Pour charger un binaire en mémoire flash NOR, resetez la cible et tapez la commande « **netload** ».

Sur la station de développement, allez sous le répertoire « images » et tapez « **./tftp 192.168.0.30** », puis transférer comme suit votre binaire vers la cible :

```
xavier@localhost:~/uClinux-pragm-new/images - Shell No. 5 - Konsole
Session Edit View Bookmarks Settings Help
2 gdbserver [0x305 , 0xa40d9 ] 0100755, sz 50260, at 0xb9a70
2 busybox [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xc5ef0 [link to 0x27780 ]
2 portmap [0x305 , 0xa40de ] 0100744, sz 81556, at 0xc5f10
2 . [0x305 , 0xa4075 ] 0040775, sz 0, at 0xd9dd0 [link to 0x55d0 ]
2 ls [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xd9df0 [link to 0x27780 ]
2 cat [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xd9e10 [link to 0x27780 ]
2 date [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xd9e30 [link to 0x27780 ]
2 sync [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xd9e50 [link to 0x27780 ]
2 inetd [0x305 , 0xa40da ] 0100744, sz 38888, at 0xd9e70
2 kill [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xe3680 [link to 0x27780 ]
2 ping [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xe36a0 [link to 0x27780 ]
2 sh [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xe36c0 [link to 0x27780 ]
1/sbin [0x305 , 0xa40cc ] 0120777, sz 4, at 0xe36e0
1 var [0x305 , 0xa407f ] 0040775, sz 0, at 0xe3710
2 .. [0x305 , 0xa4077 ] 0040775, sz 0, at 0xe3730 [link to 0x20 ]
2 . [0x305 , 0xa407f ] 0040775, sz 0, at 0xe3750 [link to 0xe3710 ]
1 usr [0x305 , 0xa407e ] 0040775, sz 0, at 0xe3770
2 .. [0x305 , 0xa4077 ] 0040775, sz 0, at 0xe3790 [link to 0x20 ]
2 . [0x305 , 0xa407e ] 0040775, sz 0, at 0xe37b0 [link to 0xe3770 ]
cp /home/xavier/uClinux-pragm-new/tools/tftp /home/xavier/uClinux-pragm-new/images
arm-elf-ld -r -o /home/xavier/uClinux-pragm-new/linux-2.4.x/romfs.o -b binary /home/xavier/uClinux-pragm-new/images/romfs.img
arm-elf-objcopy -O binary -R .note -R .comment -S /home/xavier/uClinux-pragm-new/linux-2.4.x/linux /home/xavier/uClinux-pragm-
cp /home/xavier/uClinux-pragm-new/linux-2.4.x/arch/armnommu/boot/zImage /home/xavier/uClinux-pragm-new/images/image.rom
make[1]: Leaving directory `/home/xavier/uClinux-pragm-new/vendors/Samsung/44BOX'
[xavier@localhost uClinux-pragm-new]$ cd images/
[xavier@localhost images]$ ll
total 3512
-rwxrwxr-x 1 xavier xavier 1780284 Jul 26 02:18 image.rom
-rwxrwxr-x 1 xavier xavier 820880 Jul 26 02:18 image.rom
-rw-rw-r-- 1 xavier xavier 931840 Jul 26 02:18 romfs.img
-rwxrwxr-x 1 xavier xavier 24951 Jul 26 02:18 tftp
[xavier@localhost images]$ ./tftp 192.168.0.30
tftp> bin
tftp> put image.rom
Sent 820880 bytes in 3.3 seconds
tftp>
```

uClinux - Kernel

Côté cible, vous devez alors programmer l'image téléchargée à l'adresse 0x10000 (adresse de stockage du noyau uClinux en flash NOR) :



```
root@localhost:~ - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help

*****
*
*   BIOS for FS44B0 CORE Board V3.30   *
*   Http://www.uCdragon.com           *
*
*****

Nor Flash ID is : 234B00BF
SST39VF1601 Found
Nand Flash ID is 0xEC73, Size = 16M, Status = 0xC0
Build date : Feb 21 2005--14:25:04
Machine Number is 0
IP address : 192.168.0.30
IP MASK    : 255.255.255.0
IP GATE    : 192.168.3.1
Serial baud : 115,200
Program save in nor flash
Program save address 0x10000
Program run address 0xC300000
Program boot params : root=/dev/rom0 console=ttyS0,115200 devfs=mount
CPU clock is 61,000,000Hz
Current date is 2006-7-26 [MON]
Current time is 2:15: 8
Set boot key is key2, check state high to boot
\>netload
Now download file from net to 0xC008000...
Mini TFTP Server 1.0 (IP : 192.168.0.30 PORT: 69)
Type tftp -i 192.168.0.30 put filename at the host PC
Press ESC key to exit
Starting the TFTP download...
.....
Received C8690 bytes success
\>prog 10000
program flash begin @0x10000, from ram data @0xC008000, size = 820,880Bytes
Are you sure? [y/n] y
Program 0x10000 Ok
Program 0x11000 Ok
Program 0x12000 Ok
Program 0x13000 Ok
Program 0x14000 Ok
```

Vous pouvez désormais reseter votre cibler, le BIOS va transférer l'image uClinux de la flash NOR vers la mémoire SDRAM et le noyau va s'auto décompacter puis d'exécuter.

Au final vous devrez donc voir apparaître le banner uClinux et le prompt du shell, et obtenir le même comportement qu'avec l'image « image.ram » chargée et exécutée en RAM.

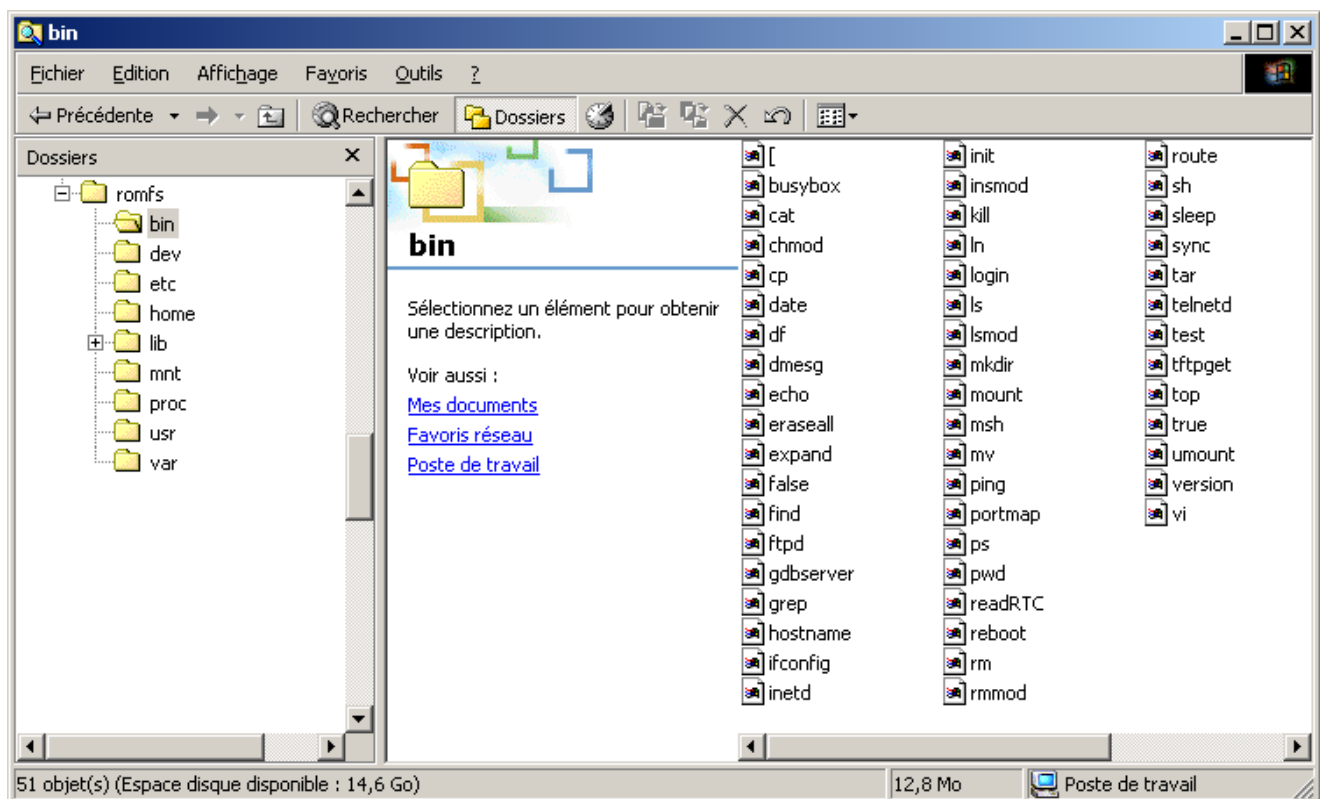
4 Modification du système de fichiers



Comme nous l'avons vu précédemment, le système de fichier de uClinux dispose d'une partie en lecture seule. Cet ensemble de fichiers constitue la base du système de fichiers, une sorte de référence inaltérable qui permettra au système de redémarrer quelque soit l'erreur commise.

Présentation du système de fichiers

Ce système de fichier en lecture seule, que nous appellerons « romfs » par la suite, se présente comme suit :



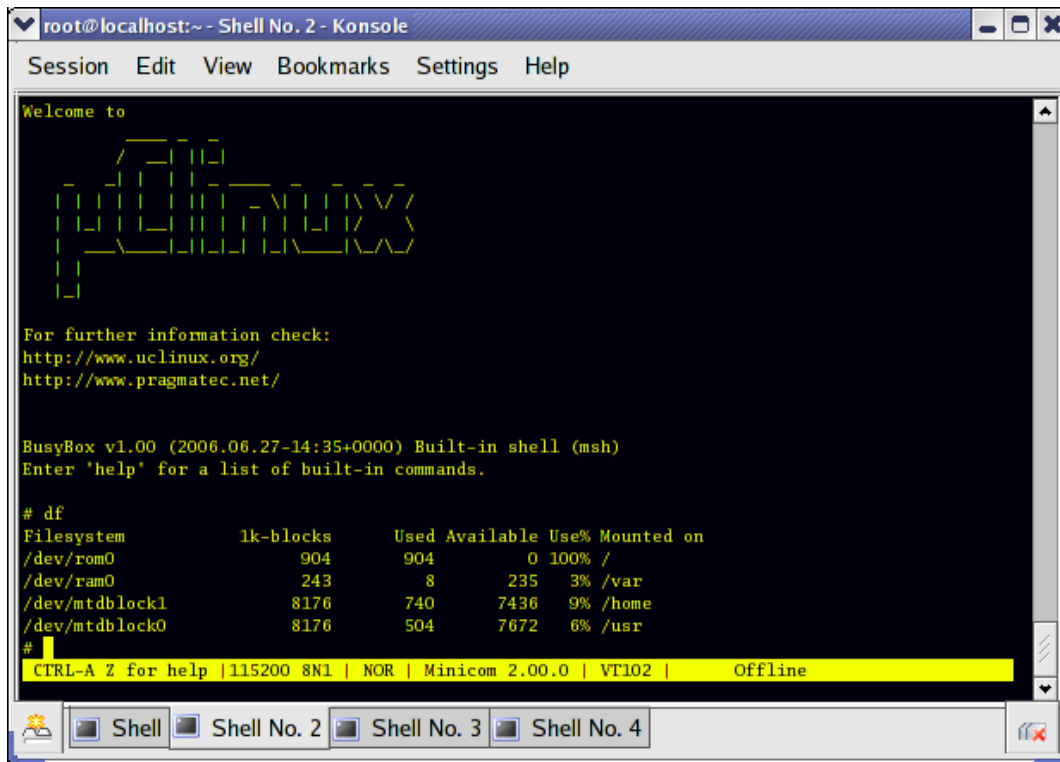
Conformément à un noyau Linux, notre système de fichier est décomposé en un certain nombre de répertoires :

- **/bin** : ensemble des programmes de base du système, dont le programme « init » (processus parent de tous les autres) et le shell.
- **/dev** : contient les fichiers d'interfaces des périphériques gérés par la carte.
- **/etc** : four tout destiné à stocker des fichiers de paramétrage (comme passwd, services ou inetd.conf).
- **/home** : répertoire racine de l'utilisateur.

uClinux - Kernel

- **/lib** : répertoire vide mais originellement destiné à contenir les bibliothèques partagées par les applications.
- **/mnt** : point de montage d'un périphérique (disque IDE, Sdcard, flash NAND, ...).
- **/proc** : ce répertoire est virtuel et ne prend pas réellement de la place dans le système de fichiers. Il permet d'interroger le noyau ou les drivers via une interface fichiers (par exemple /proc/meminfo renseigne sur la consommation mémoire).
- **/usr** : répertoire vide dans notre cas. Il peut être utilisé comme point de montage d'un périphérique.
- **/var** : seul espace du système de fichiers qui soit en lecture / écriture. Vous disposez de 256 Ko d'espace disque pour ce répertoire. Il est habituellement destiné à recevoir des fichiers temporaires comme des « pipes nommées » ou des fichiers de « lock », mais vous pouvez vous en servir pour y stocker n'importe quel type d'information.

Si votre carte possède une mémoire NAND de 16Mo, vous pouvez taper la commande « df » sous le shell et vous devriez obtenir des informations similaires aux suivantes :



```
root@localhost:~ - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help

Welcome to

   _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
  |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_| |_|

For further information check:
http://www.uclinux.org/
http://www.pragmatec.net/

BusyBox v1.00 (2006.06.27-14:35+0000) Built-in shell (msh)
Enter 'help' for a list of built-in commands.

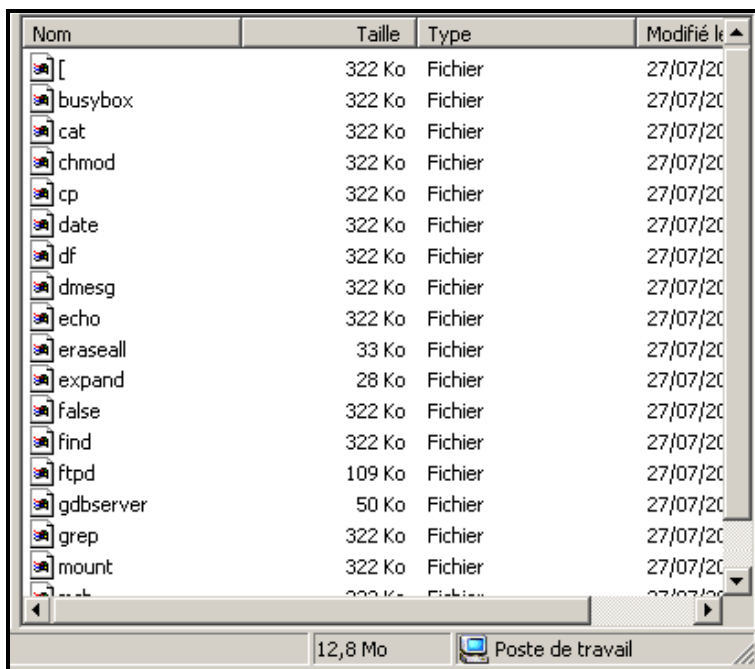
# df
Filesystem          1k-blocks    Used Available Use% Mounted on
/dev/rom0            904          904         0 100% /
/dev/ram0            243           8        235   3% /var
/dev/mtdblock1     8176         740       7436   9% /home
/dev/mtdblock0     8176         504       7672   6% /usr
#
CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.00.0 | VT102 | Offline
```

Comme vous pouvez le constater la flash NAND a été scindée en 2 partitions de taille égale (8 Mo chacune) et montée respectivement en /home et /usr.

uClinux - Kernel

Modification de la busybox

Si vous regardez en détail l'ensemble des fichiers présents sous /bin, vous verrez que beaucoup ont la même taille :



Nom	Taille	Type	Modifié le
[322 Ko	Fichier	27/07/20
busybox	322 Ko	Fichier	27/07/20
cat	322 Ko	Fichier	27/07/20
chmod	322 Ko	Fichier	27/07/20
cp	322 Ko	Fichier	27/07/20
date	322 Ko	Fichier	27/07/20
df	322 Ko	Fichier	27/07/20
dmesg	322 Ko	Fichier	27/07/20
echo	322 Ko	Fichier	27/07/20
eraseall	33 Ko	Fichier	27/07/20
expand	28 Ko	Fichier	27/07/20
false	322 Ko	Fichier	27/07/20
find	322 Ko	Fichier	27/07/20
ftpd	109 Ko	Fichier	27/07/20
gdbserver	50 Ko	Fichier	27/07/20
grep	322 Ko	Fichier	27/07/20
mount	322 Ko	Fichier	27/07/20
...

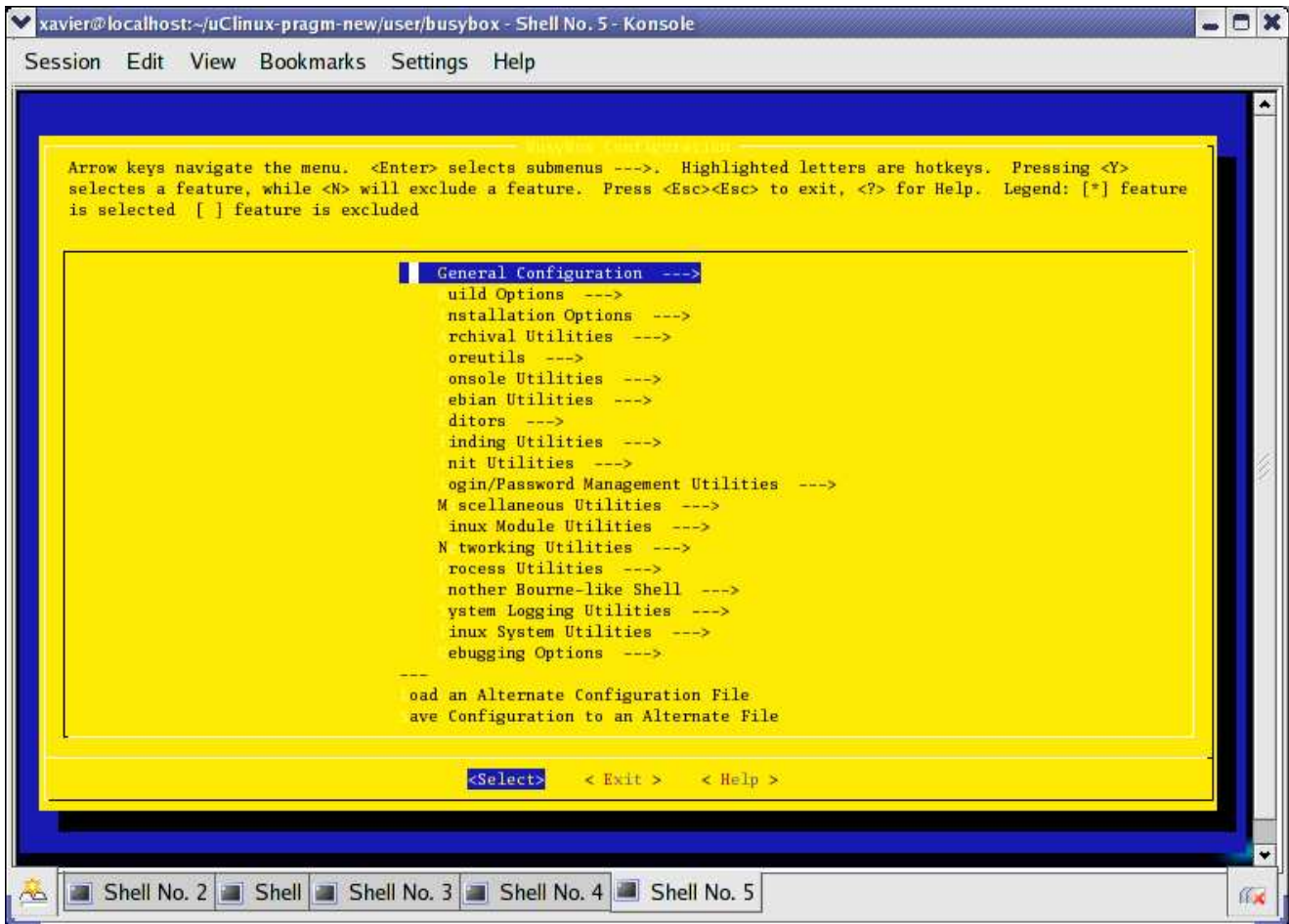
Ces programmes font en fait référence à la busybox, sorte de couteau suisse rassemblant l'ensemble de ce qui est nécessaire à un système d'exploitation et au shell associé. Ainsi la commande echo n'est autre qu'un lien sur la commande « busybox echo ».

Les programmes qui ne sont pas issus de la busybox sont :

- **eraseall** : efface une partition d'une mémoire de type mtd (votre flash NAND par exemple). Pour effacer complètement votre première partition, tapez « eraseall /dev/mtd0 ».
- **expand** : utilisé lors du démarrage pour créer l'espace en mémoire RAM du répertoire /var.
- **ftpd** : démon ftp vous permettant de vous connecter à la cible via le protocole FTP (login « root », password « root »).
- **gdbserver** : programme destiné au debug des programmes et présenté dans le document « uClinux_tutorial_s3C44B0.pdf ».
- **inetd** : démon réseau destiné entre autre à lancer les démons telnetd et ftpd.
- **portmap** : programme nécessaire à l'utilisation de NFS sur la carte.
- **readRTC** : permet de lire les informations contenues dans l'horloge temps réel (RTC) du processeur et de mettre à jour l'heure et la date système de uClinux.
- **reboot** : si vous souhaitez redémarrer la carte sans utiliser le bouton reset...
- **telnetd** : démon telnet vous permettant d'ouvrir un shell à distance depuis une station distante via le protocole TELNET (login « root », password « root »).
- **tftpget** : outil de transfert de programme de la station vers la cible en TFTP.
- **version** : affiche à l'écran la version du noyau compilé.

uClinux - Kernel

Pour modifier les programmes du shell vous n'avez en fait pas besoin de lancer un « make xconfig ». Allez en fait sous le répertoire « user/busybox » et tapez la commande « make menuconfig », vous devez obtenir l'écran suivant :



Vous pouvez à présent vous déplacer dans les menus à l'aide des touches ENTER et TAB, et valider vos choix avec la touche ESPACE.

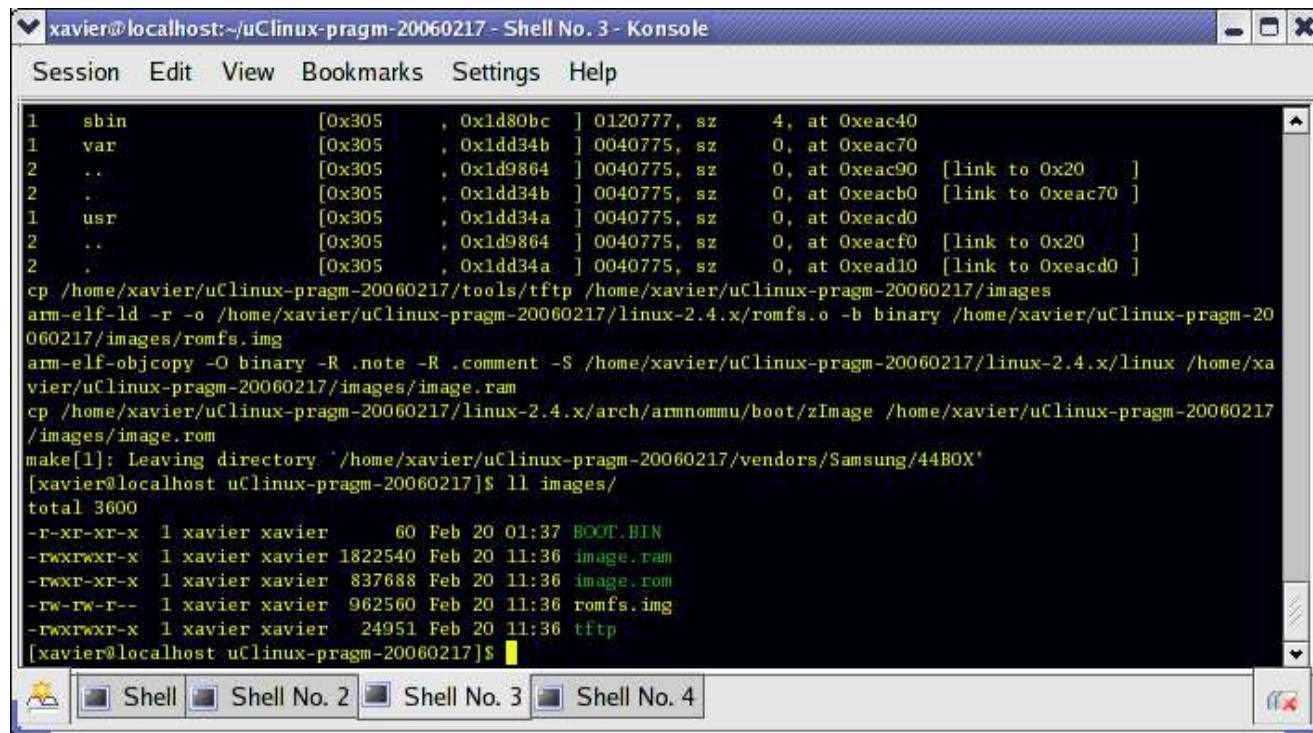
Ceci étant fait quittez en allant sur EXIT et sauvegardez votre configuration. Revenez ensuite à la racine de votre distribution (cd ../..) et tapez « **make user** ». Comme nous l'avons vu lors du chapitre précédent, vous pouvez ne construire que les applications correspondant au « step3 » sans devoir lancer une construction complète par la commande « make ».

Mais ce n'est pourtant pas tout ! Vos applications sont certes construites, mais il faut encore créer le « romfs ». Pour cela tapez les commandes « **rm romfs/*** » et « **make romfs** » pour voir votre sélection de programme apparaître sous « romfs/ ».

Pour générer l'image « romfs.img » chargée par la suite dans la flash NOR, tapez « **make image** ».

uClinux - Kernel

Enfin, la dernière opération consiste à vérifier que l'image ainsi créée puisse être logée dans l'espace qui lui est imparti en flash NOR :



```
xavier@localhost:~/uClinux-pragm-20060217 - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
1 sbin [0x305, 0x1d80bc] 0120777, sz 4, at 0xeac40
1 var [0x305, 0x1dd34b] 0040775, sz 0, at 0xeac70
2 .. [0x305, 0x1d9864] 0040775, sz 0, at 0xeac90 [link to 0x20 ]
2 .. [0x305, 0x1dd34b] 0040775, sz 0, at 0xeacb0 [link to 0xeac70 ]
1 usr [0x305, 0x1dd34a] 0040775, sz 0, at 0xeacd0
2 .. [0x305, 0x1d9864] 0040775, sz 0, at 0xeacf0 [link to 0x20 ]
2 .. [0x305, 0x1dd34a] 0040775, sz 0, at 0xead10 [link to 0xeacd0 ]
cp /home/xavier/uClinux-pragm-20060217/tools/tftp /home/xavier/uClinux-pragm-20060217/images
arm-elf-ld -r -o /home/xavier/uClinux-pragm-20060217/linux-2.4.x/romfs.o -b binary /home/xavier/uClinux-pragm-20060217/images/romfs.img
arm-elf-objcopy -O binary -R .note -R .comment -S /home/xavier/uClinux-pragm-20060217/linux-2.4.x/linux /home/xavier/uClinux-pragm-20060217/images/image.rom
cp /home/xavier/uClinux-pragm-20060217/linux-2.4.x/arch/armnommu/boot/zImage /home/xavier/uClinux-pragm-20060217/images/image.rom
make[1]: Leaving directory '/home/xavier/uClinux-pragm-20060217/vendors/Samsung/44BOX'
[xavier@localhost uClinux-pragm-20060217]$ ll images/
total 3600
-r-xr-xr-x 1 xavier xavier 60 Feb 20 01:37 BOOT.BIN
-rwxrwxr-x 1 xavier xavier 1822540 Feb 20 11:36 image.ram
-rwxr-xr-x 1 xavier xavier 837688 Feb 20 11:36 image.rom
-rw-rw-r-- 1 xavier xavier 962560 Feb 20 11:36 romfs.img
-rwxrwxr-x 1 xavier xavier 24951 Feb 20 11:36 tftp
[xavier@localhost uClinux-pragm-20060217]$
```

La taille des images « image.rom » et « romfs.img » ne doivent pas dépasser 983 000 octets !!

Sinon vous risquez de dépasser la zone allouée lors de la programmation en flash NOR et ainsi d'écraser le BIOS, ce qui rendrait la carte inutilisable. Toutefois le BIOS est équipé d'un test avant programmation et vérifie que les limites ne soient pas atteintes. Quoiqu'il en soit, le bon réflexe consiste à regarder tout de même la taille des images pour quelle puisse être logées en flash NOR.

Comme vous l'avez sans doute remarqué, la taille de l'image « romfs.img » est très proche de la taille maximale autorisée, aussi n'ajoutez pas trop de programmes sous /bin, utilisez plutôt les mémoires d'extension montée en divers endroit (comme la flash NAND en /usr et /home).

uClinux - Kernel

Sauvegarde de romfs.img en flash NOR

Attention : une image « romfs.img » incorrecte (ne contenant pas par exemple le binaire « init ») ne permettra pas à uClinux de démarrer correctement. S'il vous n'arrivez plus à démarrer votre carte avec le nouveau système de fichiers, nous vous recommandons de transférer en flash NOR l'image « romfs.img » fournie sur le CDROM.

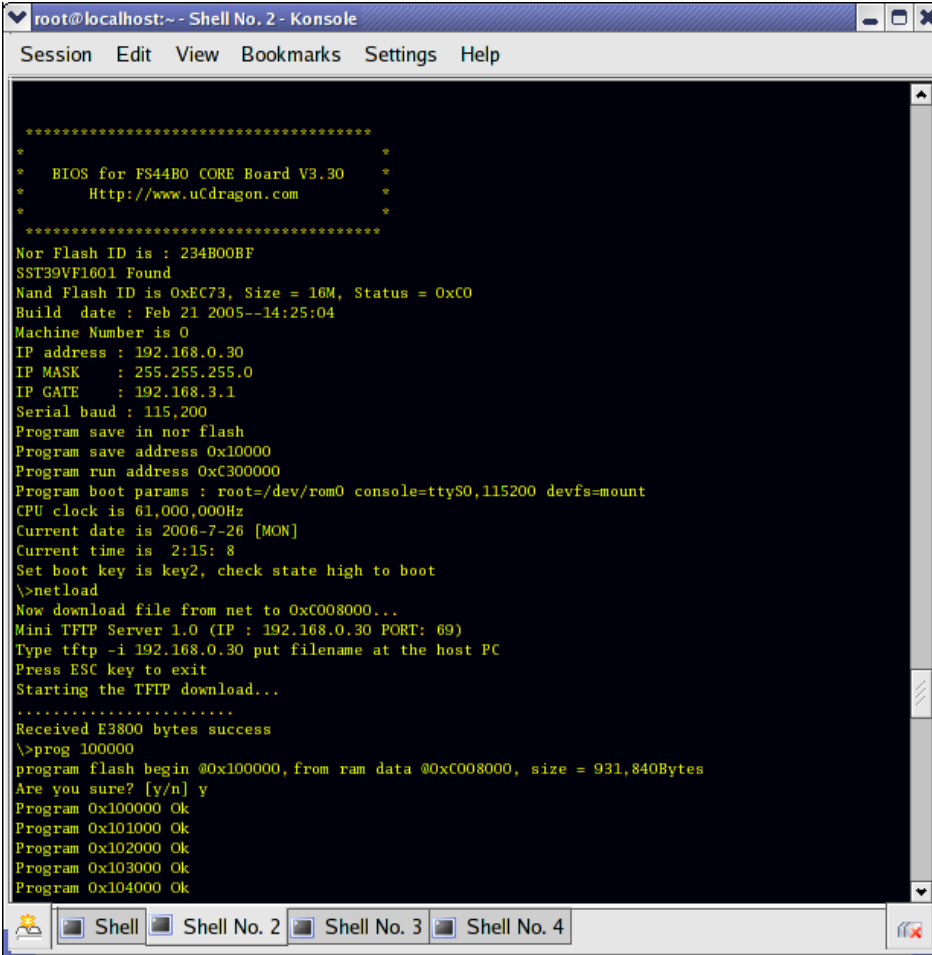
Pour charger le nouveau système de fichiers en flash NOR, il faut suivre la même procédure que pour l'image « image.rom » : resetez la cible et tapez la commande « netload ».

Sur la station de développement, allez sous le répertoire « images » et tapez « ./tftp 192.168.0.30 », puis transférer comme suit votre image « romfs.img » vers la cible :

```
xavier@localhost:~/uClinux-pragm-new/images - Shell No. 5 - Konsole
Session Edit View Bookmarks Settings Help
2 gdbserver [0x305 , 0xa40d9 ] 0100755, sz 50260, at 0xb9a70
2 busybox [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xc5ef0 [link to 0x27780 ]
2 portmap [0x305 , 0xa40de ] 0100744, sz 81556, at 0xc5f10
2 . [0x305 , 0xa4075 ] 0040775, sz 0, at 0xd9dd0 [link to 0x55d0 ]
2 ls [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xd9df0 [link to 0x27780 ]
2 cat [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xd9e10 [link to 0x27780 ]
2 date [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xd9e30 [link to 0x27780 ]
2 sync [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xd9e50 [link to 0x27780 ]
2 inetd [0x305 , 0xa40da ] 0100744, sz 38888, at 0xd9e70
2 kill [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xe3680 [link to 0x27780 ]
2 ping [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xe36a0 [link to 0x27780 ]
2 sh [0x305 , 0xa40d7 ] 0100744, sz 0, at 0xe36c0 [link to 0x27780 ]
1 sbin [0x305 , 0xa40cc ] 0120777, sz 4, at 0xe36e0
1 var [0x305 , 0xa407f ] 0040775, sz 0, at 0xe3710
2 .. [0x305 , 0xa4077 ] 0040775, sz 0, at 0xe3730 [link to 0x20 ]
2 . [0x305 , 0xa407f ] 0040775, sz 0, at 0xe3750 [link to 0xe3710 ]
1 usr [0x305 , 0xa407e ] 0040775, sz 0, at 0xe3770
2 .. [0x305 , 0xa4077 ] 0040775, sz 0, at 0xe3790 [link to 0x20 ]
2 . [0x305 , 0xa407e ] 0040775, sz 0, at 0xe37b0 [link to 0xe3770 ]
cp /home/xavier/uClinux-pragm-new/tools/tftp /home/xavier/uClinux-pragm-new/images
arm-elf-ld -r -o /home/xavier/uClinux-pragm-new/linux-2.4.x/romfs.o -b binary /home/xavier/uClinux-pragm-new/images/romfs.img
arm-elf-objcopy -O binary -R .note -R .comment -S /home/xavier/uClinux-pragm-new/linux-2.4.x/linux /home/xavier/uClinux-pragm-
cp /home/xavier/uClinux-pragm-new/linux-2.4.x/arch/armnommu/boot/zImage /home/xavier/uClinux-pragm-new/images/image.rom
make[1]: Leaving directory `/home/xavier/uClinux-pragm-new/vendors/Samsung/44BOX'
[xavier@localhost uClinux-pragm-new]$ cd images/
[xavier@localhost images]$ ll
total 3512
-rwxrwxr-x 1 xavier xavier 1780284 Jul 26 02:18 image.rom
-rwxrwxr-x 1 xavier xavier 820880 Jul 26 02:18 image.rom
-rw-rw-r-- 1 xavier xavier 931840 Jul 26 02:18 romfs.img
-rwxrwxr-x 1 xavier xavier 24951 Jul 26 02:18 tftp
[xavier@localhost images]$ ./tftp 192.168.0.30
tftp> bin
tftp> put romfs.img
Sent 931840 bytes in 8.3 seconds
tftp>
```

uClinux - Kernel

Côté cible, vous devez alors programmer l'image téléchargée à l'adresse 0x100000 (adresse de stockage du « romfs » en flash NOR) :



```
root@localhost:~ - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help

*****
*
*   BIOS for FS44B0 CORE Board V3.30   *
*   Http://www.uCdragon.com           *
*
*****

Nor Flash ID is : 234B00BF
SST39VF1601 Found
Nand Flash ID is 0xEC73, Size = 16M, Status = 0xC0
Build date : Feb 21 2005--14:25:04
Machine Number is 0
IP address : 192.168.0.30
IP MASK   : 255.255.255.0
IP GATE   : 192.168.3.1
Serial baud : 115,200
Program save in nor flash
Program save address 0x100000
Program run address 0xC300000
Program boot params : root=/dev/rom0 console=ttyS0,115200 devfs=mount
CPU clock is 61,000,000Hz
Current date is 2006-7-26 [MON]
Current time is 2:15: 8
Set boot key is key2, check state high to boot
\>netload
Now download file from net to 0xC008000...
Mini TFTP Server 1.0 (IP : 192.168.0.30 PORT: 69)
Type tftp -i 192.168.0.30 put filename at the host PC
Press ESC key to exit
Starting the TFTP download...
.....
Received E3800 bytes success
\>prog 100000
program flash begin @0x100000, from ram data @0xC008000, size = 931,840Bytes
Are you sure? [y/n] y
Program 0x100000 Ok
Program 0x101000 Ok
Program 0x102000 Ok
Program 0x103000 Ok
Program 0x104000 Ok
```

Vous pouvez désormais reseter votre cibler, le BIOS va transférer l'image uClinux de la flash NOR vers la mémoire SDRAM et le noyau va s'auto décompresser puis d'exécuter en utilisant le nouveau système de fichiers.

Au final vous devrez donc voir apparaître le banner uClinux et le prompt du shell, et obtenir le même comportement qu'avec l'ancien système de fichiers chargés.



5 Rechargement du BIOS

Le BIOS possède une sécurité qui vous interdit de flasher la mémoire NOR avec des fichiers trop gros ce qui aurait pour conséquence d'effacer le BIOS.

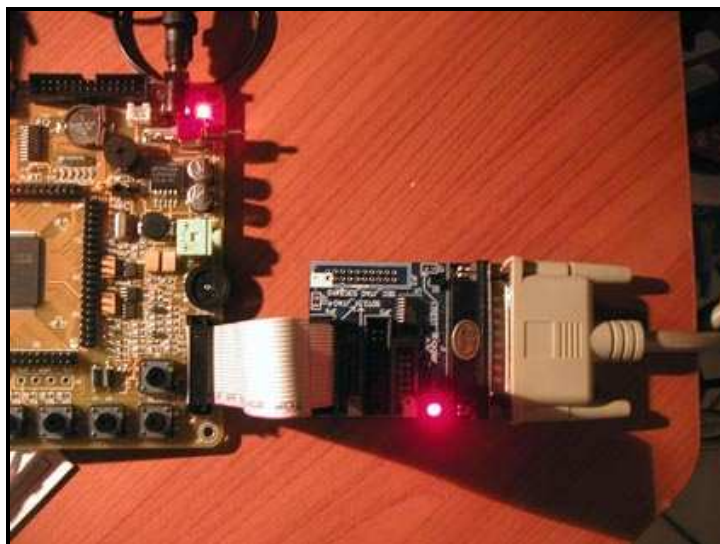
Toutefois si vous effectuez une opération non conforme (flashage en NOR à partir de l'adresse 0) ou bien que vous souhaitez mettre à jour le BIOS, nous décrivons ici la suite des opérations à effectuer pour remettre la carte dans son état d'origine :

Chargement du BIOS en flash NOR

Pour recharger les BIOS dans votre carte de développement, vous devez disposer de :

- ✓ Un PC sous Windows™ (98, 2000, ou XP) avec un port parallèle,
- ✓ Un cordon parallèle (DB25),
- ✓ Un cordon Ethernet croisé,
- ✓ La sonde JTAG fournie dans votre kit de développement,
- ✓ Le cordon RS232 livré avec votre kit de développement,
- ✓ La carte de développement alimentée et sous tension.

Commencez par connecter la sonde au cordon DB25 et à la carte de développement. Connectez le cordon parallèle au PC sous Windows™ :

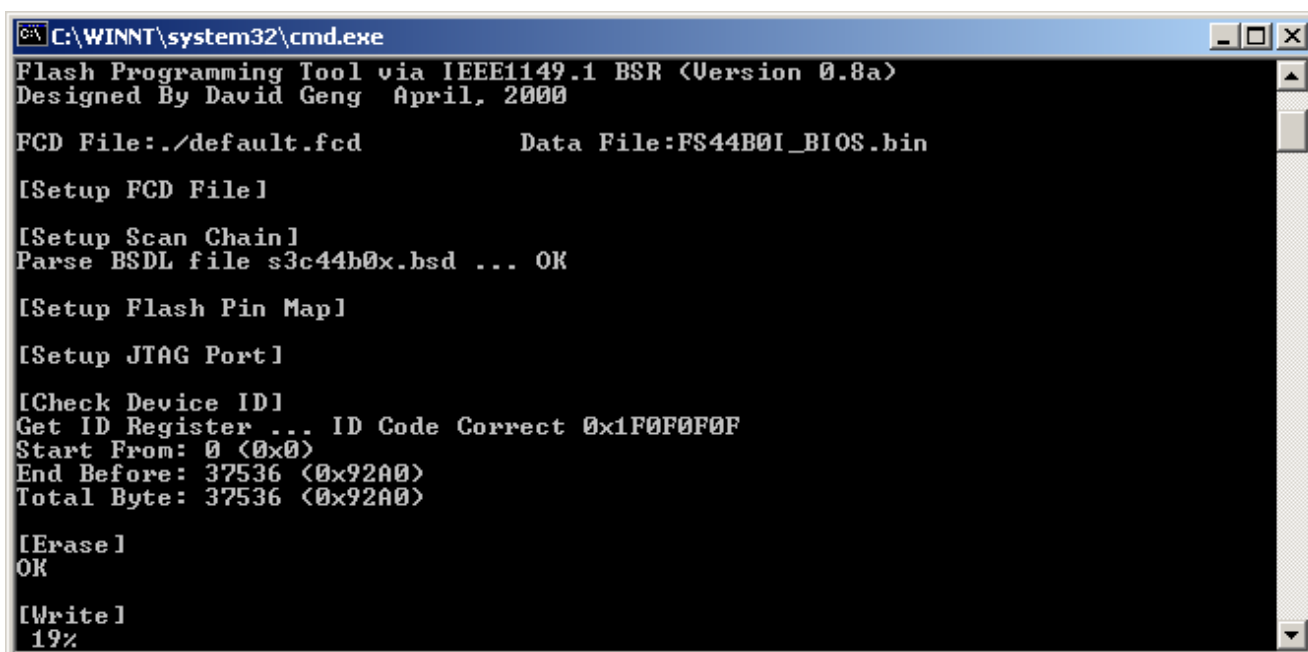


Recopiez ensuite le répertoire BIOS (contenu le programme FLUTED.exe) du CDRom sur votre disque dur et retirez l'attribut « lecture » des fichiers (click bouton droit sur le fichier, puis propriété).

uClinux - Kernel

Si vous disposez d'un PC sous Windows XP ou 2000, lancez le fichier « **BIOS_PRAGMATEC_8M.BAT** » ou « **BIOS_PRAGMATEC_16M.BAT** » suivant que vous utilisez une carte disposant de 8Mo ou 16Mo de SDRAM.

Vous devez voir apparaître une nouvelle fenêtre :



```
C:\WINNT\system32\cmd.exe
Flash Programming Tool via IEEE1149.1 BSR (Version 0.8a)
Designed By David Geng April, 2000
FCD File: ./default.fcd      Data File: FS44B0I_BIOS.bin
[Setup FCD File]
[Setup Scan Chain]
Parse BSDL file s3c44b0x.bsd ... OK
[Setup Flash Pin Map]
[Setup JTAG Port]
[Check Device ID]
Get ID Register ... ID Code Correct 0x1F0F0F0F
Start From: 0 (0x0)
End Before: 37536 (0x92A0)
Total Byte: 37536 (0x92A0)
[Erase]
OK
[Write]
19%
```

Il se peut que le programme trouve des erreurs lors de la programmation, aussi n'hésitez pas à renouveler l'opération.

Sur cette machines, il est parfois nécessaire de la redémarrer avant de lancer la programmation par sonde JTAG. Aussi n'utilisez pas votre PC pendant le transfert JTAG et attendez la fin de la programmation.

Une fois la programmation terminée, resetez la carte en gardant le doigt sur le bouton 2 (pour inhiber l'auto démarrage).

Vous devez alors voir apparaître sur votre terminal (HyperTerminal ou minicom) le BIOS d'origine.

A présent, tapez les commandes suivantes dans le BIOS apparu sur le terminal :

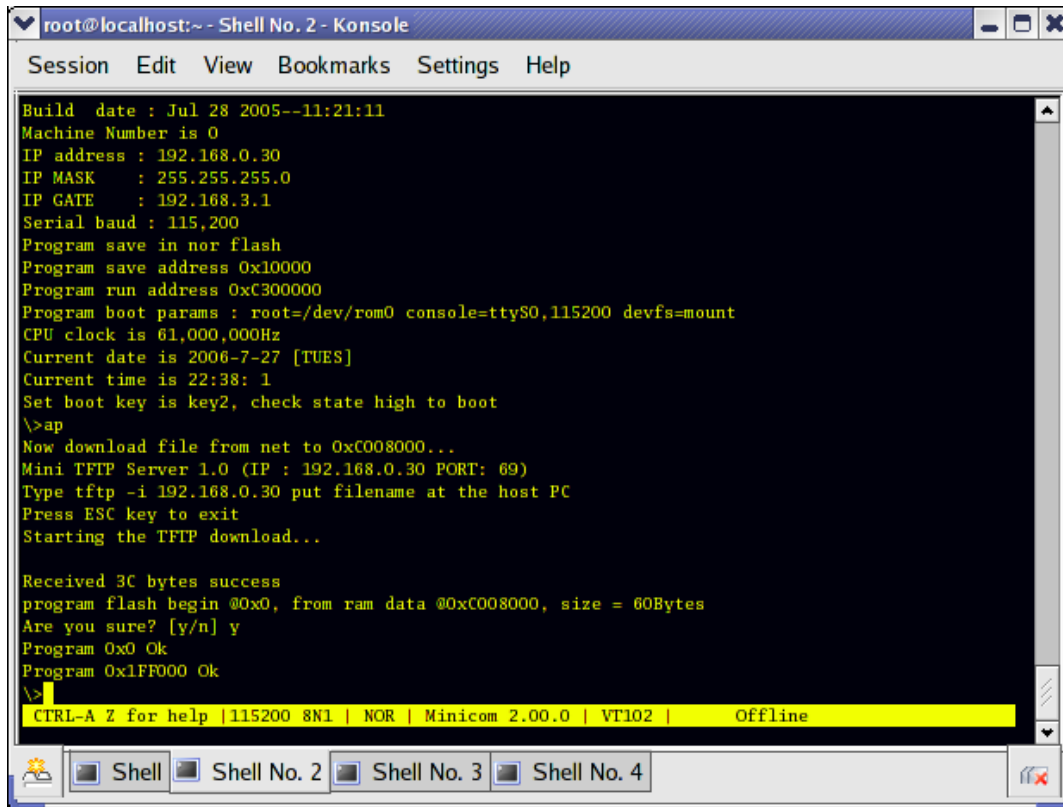
- **backup** (puis « y »)
- **defset**

De nouveau, resetez la carte en gardant le doigt sur le bouton 2 (pour inhiber l'auto démarrage). Si l'adresse IP qui apparaît n'est pas celle que vous souhaitez, tapez « **ipcfg 192.168.0.30** » par exemple puis la commande « **senv** ».

Connectez à présent le cordon Ethernet entre votre PC et la carte de développement.

uClinux - Kernel

Enfin, pour terminer l'opération de chargement du BIOS, vous devez taper la commande « **ap** » sous le BIOS, puis lancer le fichier « **BOOT.BAT** » du répertoire uClinux du CDROM ou bien lancer vous même la commande « `tftp -i 192.168.0.30 put BOOT.bin` ». A la question « Are you sure ? », répondez « **y** ».



```
root@localhost:~ - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help

Build date : Jul 28 2005--11:21:11
Machine Number is 0
IP address : 192.168.0.30
IP MASK   : 255.255.255.0
IP GATE   : 192.168.3.1
Serial baud : 115,200
Program save in nor flash
Program save address 0x10000
Program run address 0xC300000
Program boot params : root=/dev/rom0 console=ttyS0,115200 devfs=mount
CPU clock is 61,000,000Hz
Current date is 2006-7-27 [TUES]
Current time is 22:38: 1
Set boot key is key2, check state high to boot
\>ap
Now download file from net to 0xC008000...
Mini TFTP Server 1.0 (IP : 192.168.0.30 PORT: 69)
Type tftp -i 192.168.0.30 put filename at the host PC
Press ESC key to exit
Starting the TFTP download...

Received 3C bytes success
program flash begin @0x0, from ram data @0xC008000, size = 60Bytes
Are you sure? [y/n] y
Program 0x0 Ok
Program 0x1FF000 Ok
\>
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.00.0 | VT102 | Offline
```

Resetez à nouveau la cible : votre BIOS est à présent correctement installé, il ne vous reste plus qu'à transférer en flash NOR vos 2 images « **image.rom** » en 10000 et « **romfs.img** » en 100000, comme indiqué dans les chapitres précédents.

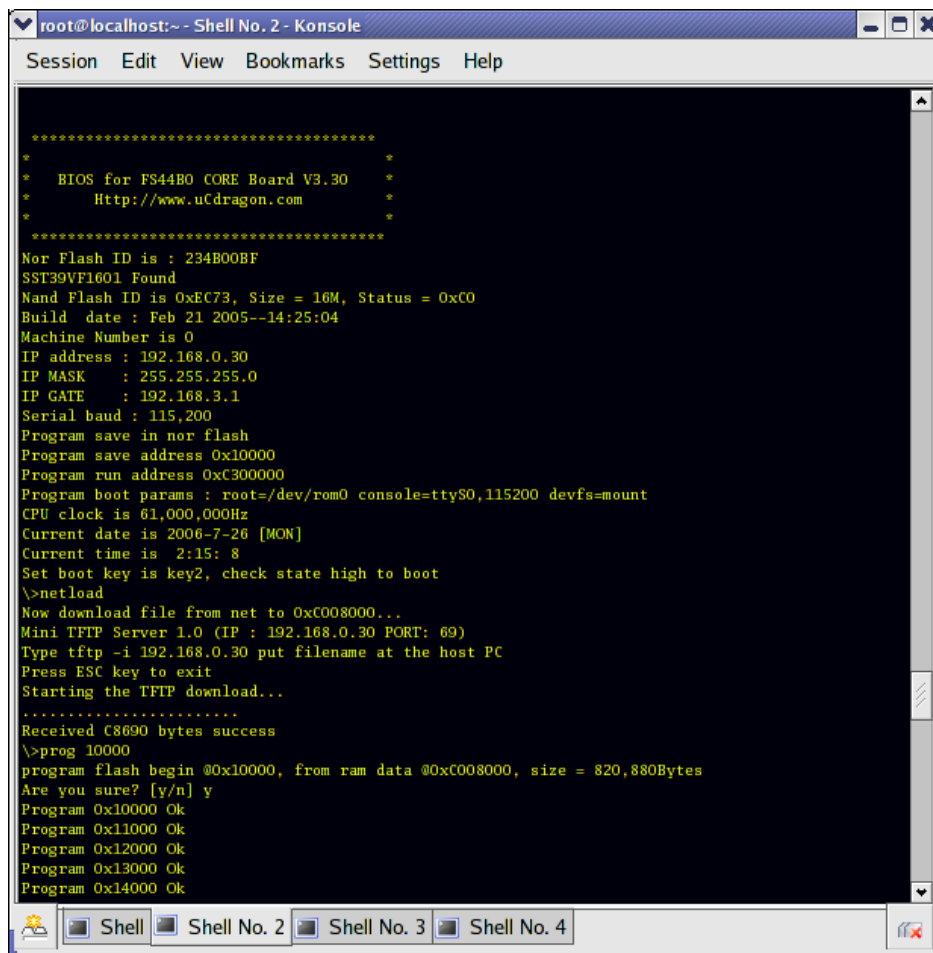
uClinux - Kernel

Chargement d'uClinux et du romfs

Pour charger un binaire en mémoire flash NOR, tapez la commande « **netload** » sous le BIOS.

Côté station Windows™, vous devez alors transférer le fichier « image.rom » en lançant le fichier « **image_rom.bat** » présent sous le répertoire uClinux.

Ensuite tapez la commande « **prog 10000** » pour écrire l'image transférée en mémoire flash NOR à l'adresse 10000 :



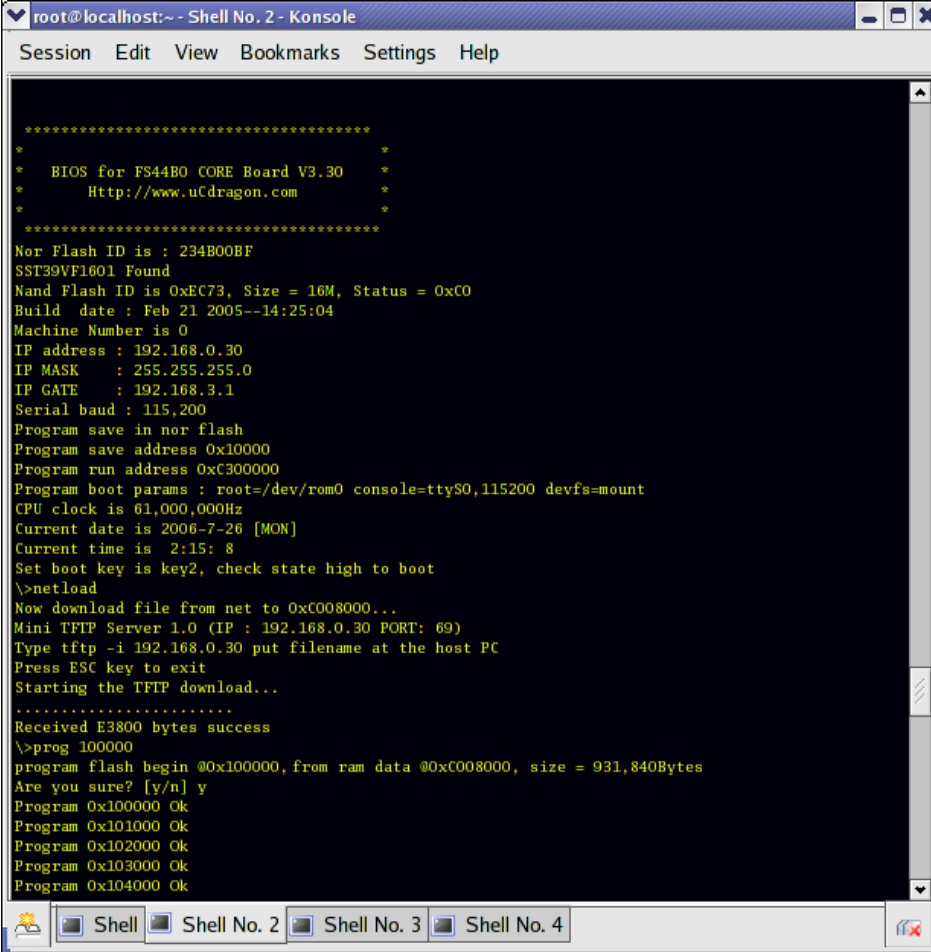
```
root@localhost:~ - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help

*****
*   BIOS for FS44B0 CORE Board V3.30   *
*   Http://www.uCdragon.com           *
*****
Nor Flash ID is : 234B00BF
SST39VF1601 Found
Nand Flash ID is 0xEC73, Size = 16M, Status = 0xC0
Build date : Feb 21 2005--14:25:04
Machine Number is 0
IP address : 192.168.0.30
IP MASK   : 255.255.255.0
IP GATE   : 192.168.3.1
Serial baud : 115,200
Program save in nor Flash
Program save address 0x10000
Program run address 0xC300000
Program boot params : root=/dev/rom0 console=ttyS0,115200 devfs=mount
CPU clock is 61,000,000Hz
Current date is 2006-7-26 [MON]
Current time is 2:15: 8
Set boot key is key2, check state high to boot
\>netload
Now download file from net to 0xC008000...
Mini TFTP Server 1.0 (IP : 192.168.0.30 PORT: 69)
Type tftp -i 192.168.0.30 put filename at the host PC
Press ESC key to exit
Starting the TFTP download...
.....
Received C8690 bytes success
\>prog 10000
program flash begin @0x10000, from ram data @0xC008000, size = 820,880Bytes
Are you sure? [y/n] y
Program 0x10000 Ok
Program 0x11000 Ok
Program 0x12000 Ok
Program 0x13000 Ok
Program 0x14000 Ok
```


uClinux - Kernel

Ceci étant fait, suivez à nouveau l'ensemble de ces opérations pour flasher la NOR avec l'image « romfs.img » en 100000 :

- **netload** (sur la cible)
- **romfs_img.bat** (sur la station)
- **prog 100000** (sur la cible)



```
root@localhost:~ - Shell No. 2 - Konsole
Session Edit View Bookmarks Settings Help

*****
*
*   BIOS for FS44B0 CORE Board V3.30
*   Http://www.uCdragon.com
*
*****
Nor Flash ID is : 234B00BF
SST39VF1601 Found
Nand Flash ID is 0xEC73, Size = 16M, Status = 0xC0
Build date : Feb 21 2005--14:25:04
Machine Number is 0
IP address : 192.168.0.30
IP MASK   : 255.255.255.0
IP GATE   : 192.168.3.1
Serial baud : 115,200
Program save in nor flash
Program save address 0x100000
Program run address 0xC300000
Program boot params : root=/dev/rom0 console=ttyS0,115200 devfs=mount
CPU clock is 61,000,000Hz
Current date is 2006-7-26 [MON]
Current time is 2:15: 8
Set boot key is key2, check state high to boot
\>netload
Now download file from net to 0xC008000...
Mini TFTP Server 1.0 (IP : 192.168.0.30 PORT: 69)
Type tftp -i 192.168.0.30 put filename at the host PC
Press ESC key to exit
Starting the TFTP download...
.....
Received E3800 bytes success
\>prog 100000
program flash begin @0x100000, from ram data @0xC008000, size = 931,840Bytes
Are you sure? [y/n] y
Program 0x100000 Ok
Program 0x101000 Ok
Program 0x102000 Ok
Program 0x103000 Ok
Program 0x104000 Ok
```

Ceci étant terminé, resetez la cible sans appuyer sur le bouton 2 : le BIOS va démarrer automatiquement uClinux !