

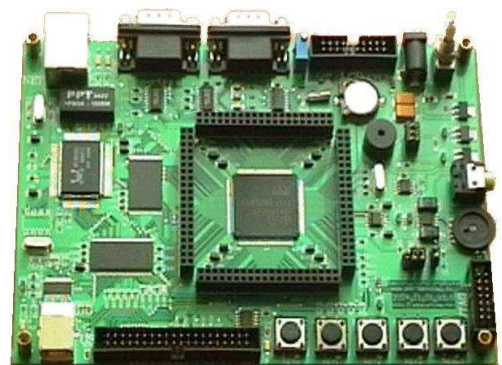
uClinux - Mise en oeuvre

Pragmatec

Produits et services dédiés aux systèmes embarqués

Tutorial uClinux

ARM7 Development Starter Kit





Bâtiment EARHART
ZAC Grenoble Air Parc

38590 St Etienne de St Geoirs - France

www.pragmatec.net

uClinux - Mise en oeuvre



Bâtiment EARHART
ZAC Grenoble Air Parc

38590 St Etienne de St Geoirs - France

www.pragmatec.net

uClinux - Mise en oeuvre

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 n'aborderons pas ici les détails internes du noyau (modifications, drivers, gestion mémoire, ...) ceci étant destiné à un document plus spécifique. Toutefois vous découvrirez comment rapidement et simplement réaliser vos applications sous uClinux, mais aussi comment les déployer et les débiter sur la cible.

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.

TABLE DES MATIERES

1	<i>Préambule</i>	5
	La carte de développement.....	5
	Développement croisé sous Linux.....	6
2	<i>L'environnement de développement</i>	7
	Installation des sources de uClinux	7
	Installation de la chaîne de compilation.....	8
	Paramétrage de minicom.....	9
	Paramétrage de votre serveur tftp.....	9
3	<i>Chargement de uClinux</i>	12
	Compilation de uClinux.....	12
4	<i>Installation d'Eclipse</i>	21
5	<i>Programme DEMO_IHM</i>	30
6	<i>Informations complémentaires</i>	32
	Connexion par telnet.....	32
	Transfert FTP.....	33
	Partage réseau par NFS.....	34
	Modification des paramètres réseaux.....	35
	Nano-X sans LCD.....	35

uClinux - Mise en oeuvre

1 **Préambule**

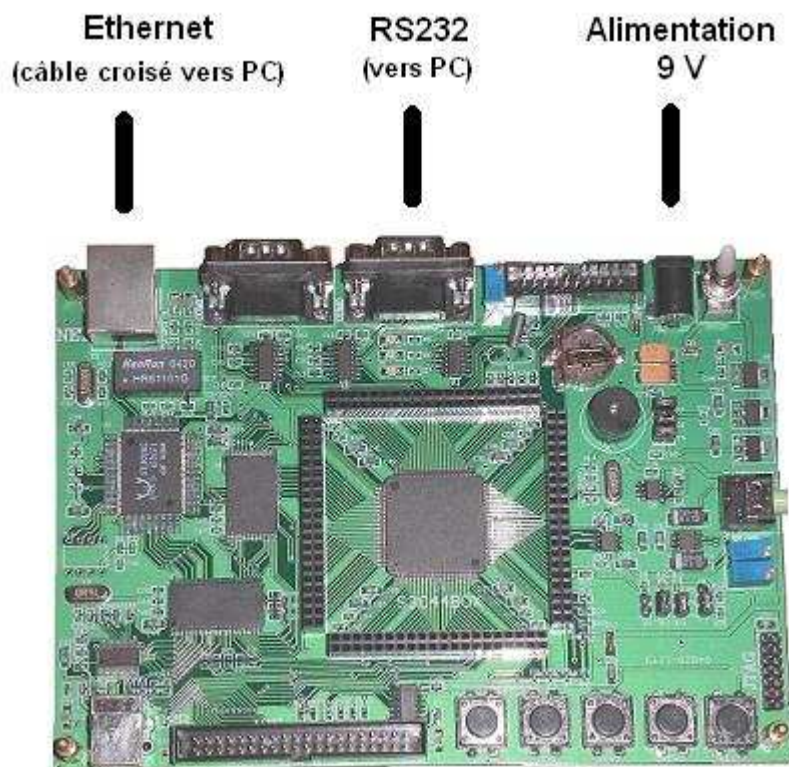


Le but de ce tutorial est de démontrer combien il est simple de développer une application uClinux sans pour autant devoir prétendre à une grande expérience dans le monde Linux. Vous y apprendrez comment installer votre environnement de développement, comment utiliser votre carte ARM7 avec uClinux, les commandes de bases et surtout comment réaliser votre propre application et la débiter sur la cible.

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 passer en revue la totalité de ces périphériques et de leur mise en œuvre. Toutefois il sera enrichi dans le temps afin d'offrir des exemples de mise en œuvre de ces périphériques.

Pour ce qui est des outils de développement, nous utiliserons la RS232 (port0) pour dialoguer avec la carte et le port Ethernet pour transférer des programmes en RAM.



uClinux - Mise en oeuvre

Développement croisé sous Linux

Dans ce tutorial nous allons considérer un développement croisé depuis une plate-forme 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).

Vous pouvez aussi télécharger Eclipse v3.1 directement du site officiel et procéder à son installation. N'oubliez pas d'installer aussi une machine virtuelle Java si vous n'en possédez pas une sur votre station (disponible sur le site de SUN). Pour la suite des opérations, vérifiez la présence de l'interpréteur de script TK « wish » (mettez à jour vos packages dans le cas contraire).

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.

uClinux - Mise en oeuvre

2 L'environnement de développement



Nous abordons ici l'installation des outils et des sources du noyau uClinux qui sera chargé sur la cible.

Installation des sources de uClinux

Nous créons un répertoire de travail sous notre homedirectory : /home/xavier/uClinux. Les sources seront décompressées à cet endroit ainsi que les différents outils comme suit :

Attention de ne jamais installer et compiler vos sources uClinux en tant que ROOT. Loggez vous en tant qu'utilisateur attiré sur votre station Linux !

```
xavier@localhost:~ - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
[xavier@localhost ~]$ tar xvzf uClinux-pragm-20060217.tar.gz
```

```
xavier@localhost:~/uClinux-pragm-20060217 - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
[xavier@localhost uClinux-pragm-20060217]$ ll
total 236
drwxr-xr-x  2 xavier xavier 4096 Apr 15  2004 bin
drwxr-xr-x  3 xavier xavier 4096 Feb 20 05:42 config
lrwxrwxrwx  1 xavier xavier  33 Feb 20 09:02 config.arch -> vendors/Samsung/44BOX/config.arch
-rw-rw-r--  1 xavier xavier 1694 Feb 20 09:02 config.in
-rwxr-xr-x  1 xavier xavier 45598 Feb 20 09:02 config.tk
-rw-r--r--  1 xavier xavier 18007 Apr  8  2004 COPYING
drwxr-xr-x  3 xavier xavier 4096 Apr  8  2004 Documentation
drwxr-xr-x 11 xavier xavier 4096 Dec 17 14:19 firmware
-rw-rw-r--  1 xavier xavier  649 Feb 15 10:30 FS44BOX-II-step1
drwxrwxr-x  2 xavier xavier 4096 Feb 20 09:05 images
drwxr-xr-x 18 xavier xavier 4096 Feb 20 09:02 lib
drwxr-xr-x 18 xavier xavier 4096 Feb 20 09:05 linux-2.4.x
-rw-r--r--  1 xavier xavier 9387 Dec 18 00:08 Makefile
-rw-r--r--  1 xavier xavier 4934 Apr  8  2004 README
-rw-r--r--  1 xavier xavier 4096 Feb 20 09:04 tools
drwxrwxr-x 11 xavier xavier 4096 Feb 20 09:04 tools
-rw-r--r--  1 xavier xavier 1743 Apr  8  2004 SOURCE
drwxr-xr-x  3 xavier xavier 4096 Feb 17 04:52 tools
drwxrwxr-x 10 xavier xavier 4096 Feb 20 10:24 tutorial
drwxr-xr-x 18 xavier xavier 4096 Dec 17 14:30 uclibc
drwxr-xr-x 177 xavier xavier 4096 Feb 20 03:31 user
drwxr-xr-x  7 xavier xavier 4096 Dec 14 21:52 vendors
[xavier@localhost uClinux-pragm-20060217]$
```

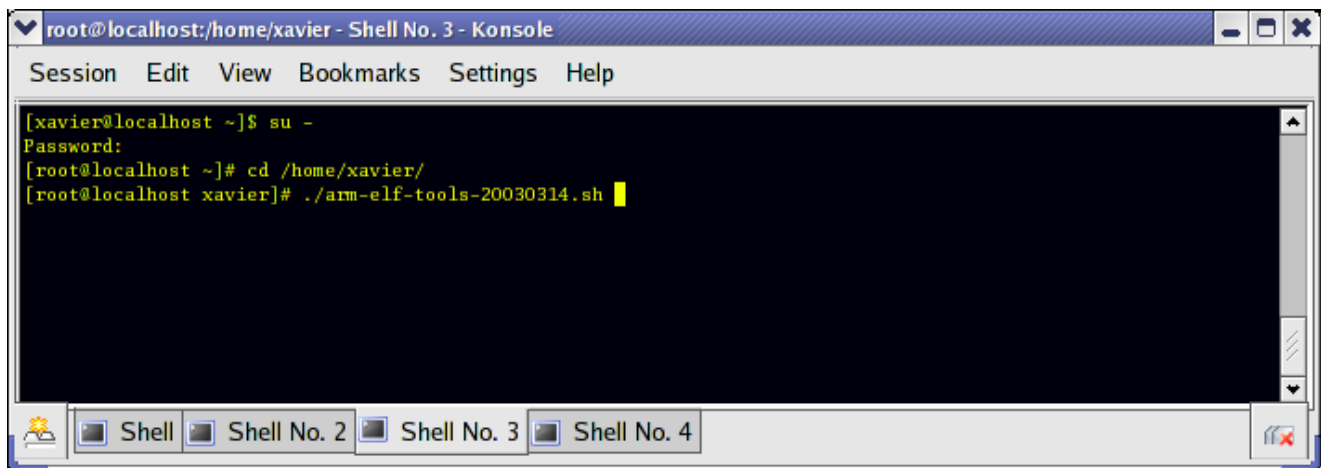
uClinux - Mise en oeuvre

Dans la liste des fichiers et répertoires, les plus significatifs sont :

- ✓ **images** : contient l'ensemble des fichiers résultant de la compilation (linux_bootram.bin, linux_boot_rom.bin et romfs.bin). Ces fichiers seront chargés dans la cible soit en RAM ou en FLASH selon l'utilisation.
- ✓ **linux** : répertoire dans lequel se trouvent les sources du noyau Linux (version uClinux) ainsi que les drivers associés
- ✓ **Makefile** : script de génération (du noyau, du file system, des librairies,...)
- ✓ **romfs** : miroir du file system présent dans la cible. Lors de la génération du binaire de la cible, ce file system sera associé au noyau uClinux au sein des fichiers linux_bootram.bin et linux_bootrom.bin.
- ✓ **step1** : fichier de configuration préformaté pour votre cible.
- ✓ **uClibc** : contient les sources de la librairie C utilisée avec uClinux.
- ✓ **user** : ensemble de programmes destinés à être associés aux file system. On y trouve des programme de base comme le shell, mais aussi des programmes plus élaborés comme Microwindows qui vous permet d'avoir un mini serveurX sur votre carte ARM7 (via le framebuffer). Bien entendu c'est aussi là que ce trouveront les exemples de ce tutorial.
- ✓ **Vendors** : répertoire spécifique à chaque type de cible, le classement se faisant par marque de fabriquant. Les fichiers qui nous concernent sont sous uClinux-dist/vendors/Samsung/S3C44B0.

Installation de la chaîne de compilation

Toujours dans notre homedirectory, passez temporairement ROOT pour installer le package des outils de développement pour ARM7 comme suit :



```
root@localhost:/home/xavier - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
[xavier@localhost ~]$ su -
Password:
[root@localhost ~]# cd /home/xavier/
[root@localhost xavier]# ./arm-elf-tools-20030314.sh
```

Toujours en tant que ROOT tapez la commande suivante : « **mkdir /usr/local/arm-elf/include/config ; touch /usr/local/arm-elf/include/config/autoconf.h** ».

Copiez le fichier arm-elf-gdb fournis sur le CDROM dans le répertoire /usr/local/bin, puis quitter le mode ROOT (commande « exit »).

Attention : n'oubliez pas de passer « root » temporairement pour procéder à cette installation !

uClinux - Mise en oeuvre

Paramétrage de minicom

Comme équivalent d'HyperTerminal sous Windows nous utilisons minicom. Si vous bénéficiez pas d'un port série sur votre ordinateur portable par exemple, utilisez une interface USB/RS232 très bien gérée de nos jours par la plupart des distributions Linux du commerce.

Pour paramétrer minicom, vous avez besoin de passe « root » :

```
A - Port série : /dev/ttyS0
B - Emplacement du fichier de verrouillage : /var/lock
C - Programme d'appel intérieur :
D - Programme d'appel extérieur :
E - D0bit/Parité/Bits : 115200 8N1
F - Contrôle de flux matériel : Non
G - Contrôle de flux logiciel : Non

Changer quel réglage ? [ ]
```

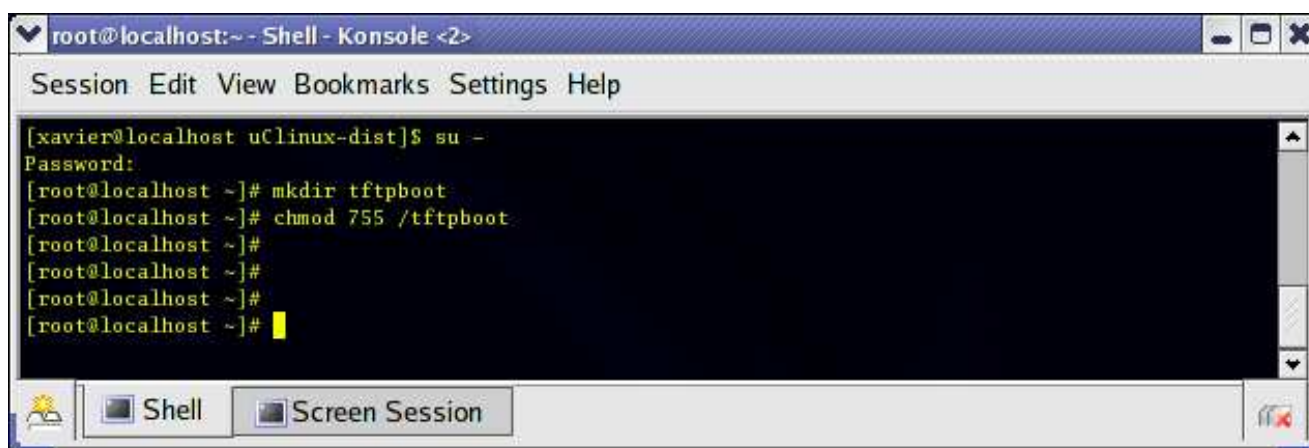
Pour accéder aux diverses commandes, tapez simultanément sur les touches « CONTROL » et le caractère de votre commande (Faire CTRL+A, puis O, puis à l'aide de la flèche du pavé numérique, sélectionnez « configuration du port série » et tapez ENTER).

Vous pouvez aussi utiliser la commande « `minicom -s` » pour directement accéder aux paramètres.

Pour vérifier que votre paramétrage est correct, vous pouvez mettre sous tension votre carte une fois minicom lancé et paramétré : vous devriez arriver au prompt de uClinux !

Paramétrage de votre serveur tftp

Afin de pouvoir charger de façon quasi automatique votre noyau et ses applications sur votre cible, nous allons utiliser le tftp. Il s'agit d'un transfert FTP (donc via le réseau) basé sur le fait qu'un client émet une requête vers un serveur qui est en charge de lui envoyer un certain fichier.



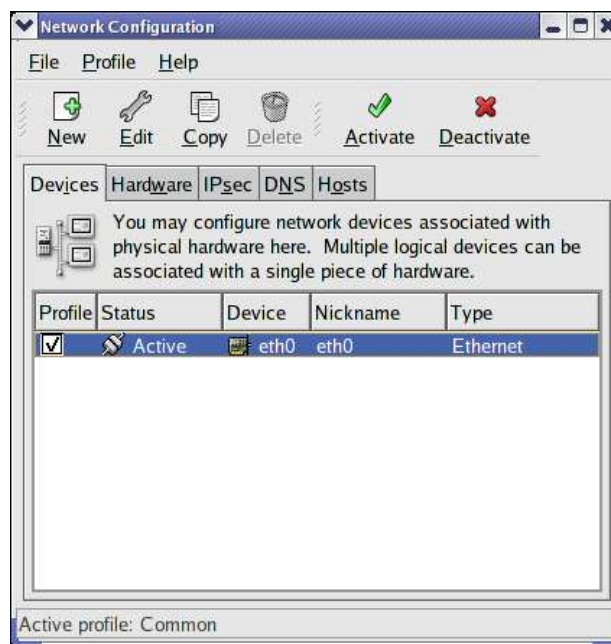
```
root@localhost:~ - Shell - Konsole <2>
Session Edit View Bookmarks Settings Help
[xavier@localhost uClinux-dist]$ su -
Password:
[root@localhost ~]# mkdir tftpboot
[root@localhost ~]# chmod 755 /tftpboot
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
```

uClinux - Mise en oeuvre

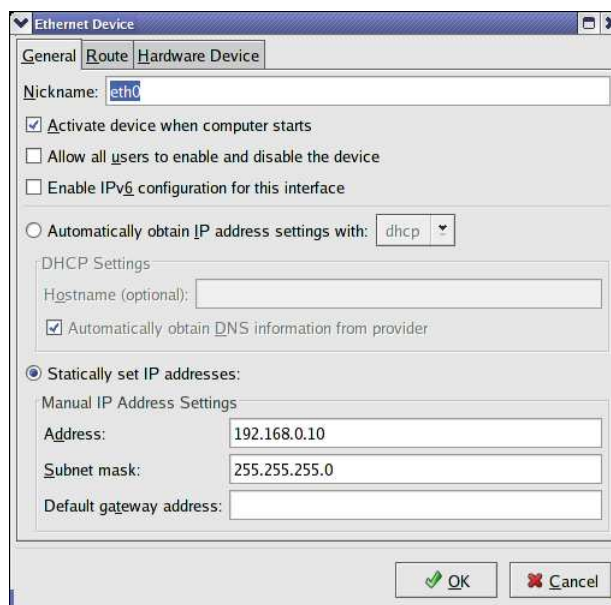
Pour cela nous avons besoin d'un répertoire « tftpboot » à la racine de votre station de développement (qui tiendra le rôle de serveur FTP).

Votre carte est préalablement installée avec l'adresse IP : 192.168.0.30. Si pour une raison quelconque cette adresse réseau ne vous la convient pas vous pouvez la changer sous minicom par une commande BIOS (ipcfg 192.180.0.77 par exemple).

Nous allons à présent donner l'adresse IP 192.168.0.10 à la station de développement. Dans System Settings / Network de la FedoraCore3 :



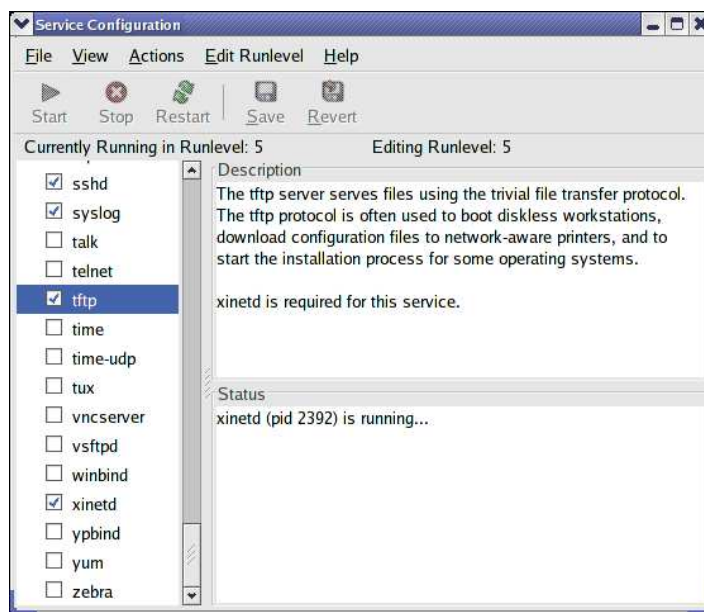
Double cliquez sur la ligne correspondant à votre port Ethernet connecté à la carte ARM7 afin de modifier ces paramètres :



uClinux - Mise en oeuvre

Vous pouvez aussi taper la commande « `ifconfig -a eth0 192.168.0.10` » pour assigner une nouvelle adresse IP à votre carte en eth0. Ensuite dans la fenêtre ci-dessous, désactivez puis réactivez le service pour retrouver vos paramètres d'origine.

Enfin activez les services tftp et xinetd (cliquez sous FedoraCore3 sur System Settings/Services/) :



Modifiez les paramètres du serveur tftp pour qu'il utilise le répertoire /tftpboot :

```
xavier@localhost:~/uClinux/uClinux-dist - Shell - Konsole
Session Edit View Bookmarks Settings Help
[xavier@localhost uClinux-dist]$ more /etc/xinetd.d/tftp
# default: off
# description: The tftp server serves files using the trivial file transfer \
#               protocol. The tftp protocol is often used to boot diskless \
#               workstations, download configuration files to network-aware printers, \
#               and to start the installation process for some operating systems.
service tftp
{
    socket_type           = dgram
    protocol              = udp
    wait                  = yes
    user                  = root
    server                = /usr/sbin/in.tftpd
    server_args           = -c -s /tftpboot
    disable               = no
    per_source            = 11
    cps                   = 100 2
    # flags                = IPv4
}
[xavier@localhost uClinux-dist]$
```

Surtout n'oubliez pas de dévalider votre parefeu (firewall) ou au moins d'autoriser les transferts depuis l'adresse IP de la carte (192.168.0.30).

uClinux - Mise en oeuvre

3 Chargement de uClinux



Nous allons à présente compiler le noyau uClinux tel qu'il a été préconfiguré par Pragmatec. La carte dont vous avez fait l'acquisition est déjà pré-installé avec un noyau uClinux et tout un ensemble de programmes utilisateurs, il n'est donc pas nécessaire de charger un nouveau noyau dans la carte. Par contre il est nécessaire de la recompiler complètement pour pouvoir développer de nouvelles applications.

Compilation de uClinux

Les fichiers de paramètres du noyau et des applications sont déjà présents dans votre distribution et se nomment (l'exemple ci-dessous est donné pour une **DEV-44B0X-II**) :

- ✓ **./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

Avant de compiler le noyau et les applications, faites un « make clean » destiné à effacer les traces de références à un autre environnement de développement. Ensuite les autres étapes de la compilation seront :

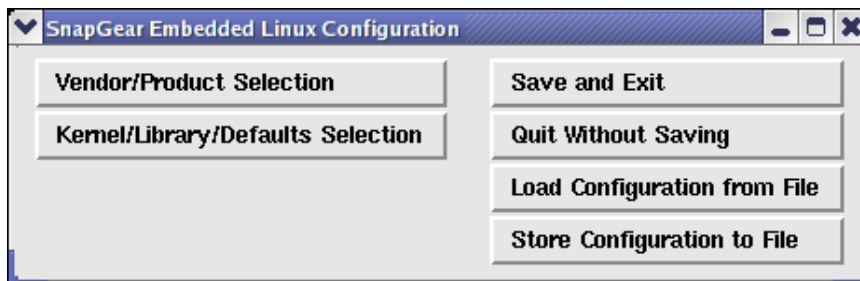
```
xavier@localhost: ~/uClinux-pragm-20060217 - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
[xavier@localhost ~]$ cd uClinux-pragm-20060217/
[xavier@localhost uClinux-pragm-20060217]$ make clean
[xavier@localhost uClinux-pragm-20060217]$ make xconfig
[xavier@localhost uClinux-pragm-20060217]$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

Le « make xconfig » qui suit permet de paramétrer le noyau et les applications.

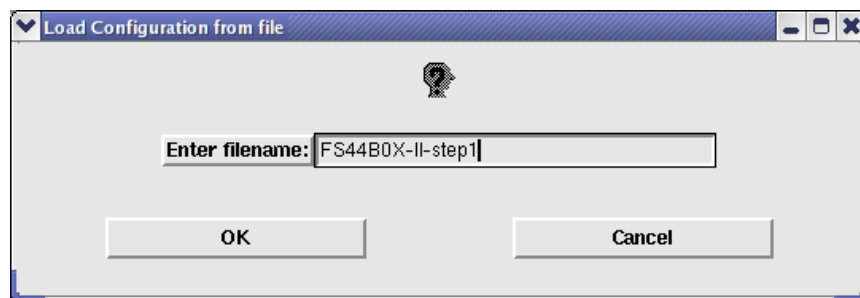
Nous allons passer en revue les principales fenêtres qui apparaîtront à l'écran.

uClinux - Mise en oeuvre

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

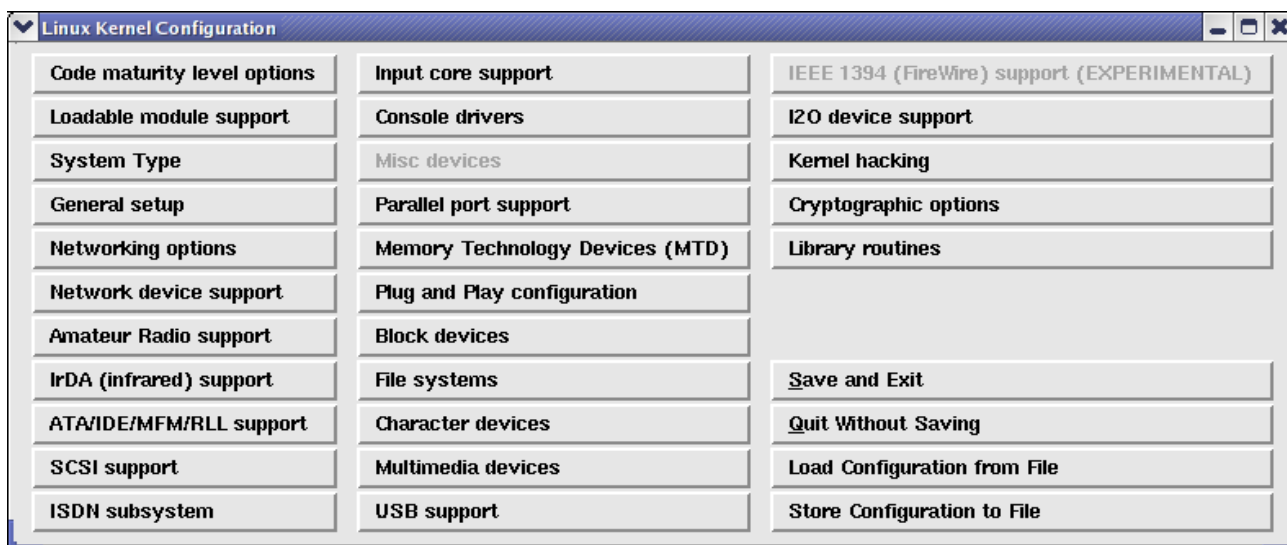


Cliquez sur « **Load Configuration from File** » pour charger la configuration correspondante à votre carte.



Cliquez sur **OK** pour valider la sélection, puis sur « **Target Platform Selection** » pour connaître les choix de la configuration.

Pour passer à la fenêtre suivante, cliquez sur « **Save and Exit** ». La seconde fenêtre principale apparaît à l'écran :



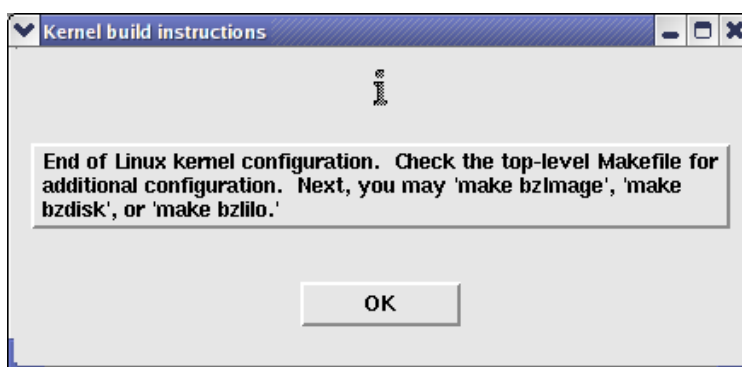
Cliquez à nouveau sur « **Load Configuration from File** » et chargez le fichier **DEVII-step2**.

uClinux - Mise en oeuvre

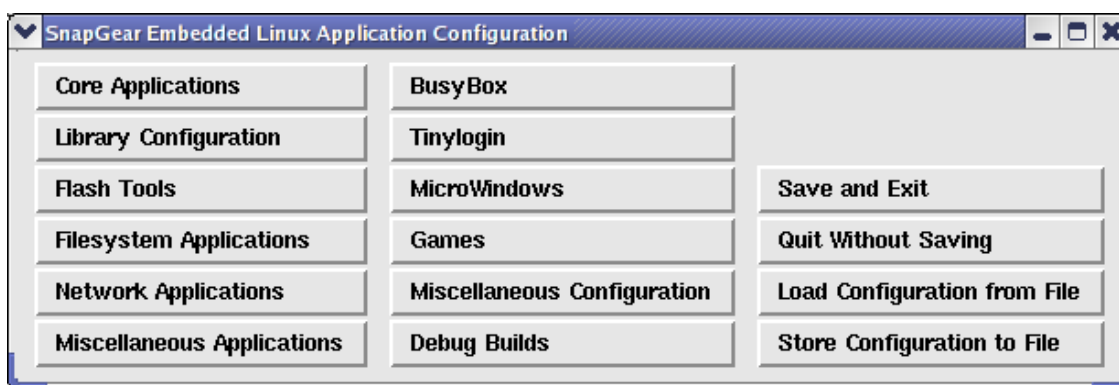
Vous pouvez visionner les paramètres sélectionner dans les différentes fenêtres mais ne modifiez rien si vous n'êtes pas sûr de savoir ce que vous faites. Vous y trouverez les paramètres propres aux différents périphériques de la carte.

Cliquez finalement 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. Cliquez sur OK pour fermer cette fenêtre.



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

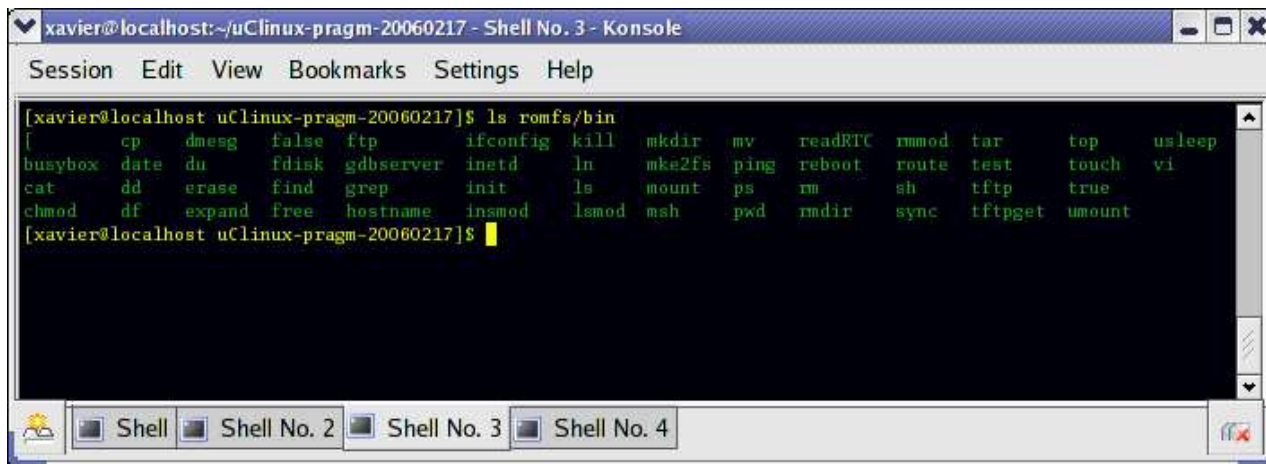


Cliquez sur « **Load Configuration from File** » est chargé le fichier **DEVII-step3**.

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

uClinux - Mise en oeuvre

Voici les différents programmes qui ont été sélectionnés :



```
[xavier@localhost uClinux-pragm-20060217]$ ls romfs/bin
[
  cp      dmesg  false  ftp      ifconfig  kill    mkdir   mv      readRTC  rmdir   tar     top     usleep
busybox  date   du     fdisk   gdbserver  inetd   ln      mke2fs  ping    reboot  route  test   touch   vi
cat      dd     erase  find    grep      init    ls      mount   ps      rm      sh     tftp   true
chmod   df     expand  free    hostname  insmod  lsmmod  msh     pwd     rmdir  sync   tftpget  umount
[xavier@localhost uClinux-pragm-20060217]$
```

Busybox :

La busybox est une sorte de couteau suisse composé d'un shell basic et d'une multitude de commandes shell classiques mais spécialement repensé pour les systèmes embarqués.

- cat : affichage de fichier ASCII
- chmod : modification des droits des fichiers ou répertoires
- cp : copie de fichiers ou répertoires
- date : affiche la date sous forme ASCII sur le terminal
- dd : copie et conversion de fichiers
- df : donne l'occupation des disques en %
- dmesg : liste les derniers affichage kernel (via printk)
- du : affichage la taille des fichiers et répertoires courants
- false : indicateur de fautes
- find : recherche un fichier dans l'arborescence
- free : affiche la mémoire disponible
- grep : recherche d'une chaîne de caractère dans l'arborescence
- hostname : définit le nom « réseau » de la cible
- ifconfig : paramètre les liaisons Ethernet
- insmod : chargement dynamique des modules
- kill : arrêt d'un processus
- ln : création de liens (symboliques ou non)
- ls : liste les fichiers et répertoires de l'arborescence
- lsmmod : liste les modules chargés en mémoire

uClinux - Mise en oeuvre

- `mkdir` : crée un nouveau répertoire
- `mount` : permet le montage de node dans le filesystem
- `msh` : shell busybox
- `mv` : déplacement de fichiers ou répertoires
- `ping` : test de connexion réseau basic
- `ps` : liste les processus en mémoire
- `pwd` : fournit le répertoire courant
- `reboot` : permet de redémarrer la cible
- `rm` : effacement d'un fichier ou répertoire
- `rmdir` : effacement d'un répertoire
- `rmod` : déchargement d'un module présent en mémoire
- `route` : définition des tables de routage réseau
- `sh` : shell busybox
- `sync` : mise à jour d'un filesystem
- `tar` : outil d'archivage
- `test` : test d'expressions
- `top` : affiche de la charge moyenne des processus
- `touch` : création de fichiers
- `true` : indicateur à vrai
- `umount` : démontage d'un filesystem
- `usleep` : attente d'un certain délai
- `vi` : éditeur en ligne

erase : permet d'effacer le contenu d'une partition MTD (mémoire NAND)

expand : création d'un espace mémoire

fdisk : création et modification de partitions

ftp : transfert réseau via TCP/IP

gdbserver : déboguer d'applications côté cible

inetd : démon couche réseau

init : premier processus à démarrer (processus père)

mke2fs : création d'une partition de type ext2

readRTC : mise à jour de l'horloge système Linux

uClinux - Mise en oeuvre

fttp : transfert réseau udp

fttpget : macro de transfert udp rapide

Cette fois-ci vous quittez définitivement l'environnement de paramétrage de Linux. Tapez « make clean », puis « make dep » et enfin « make » pour créer les binaires uClinux qui seront chargés dans votre carte.

```
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]$
```

L'étape de compilation se termine par une édition de lien et la recopie des binaires de programmation dans le répertoire « images »

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 sous minicom 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.

Si vous souhaitez reflasher entièrement votre carte ou si celle-ci est vierge, il vous faudra faire un « netload » suivi d'une commande « prog xxxx », ou xxxx vaut 100000 pour le fichier « romfs.img » et 10000 pour le fichier « image.rom ». Vous aurez donc 2 transferts à effectuer.

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 de 2Mo.

Pour charger un binaire en mémoire RAM et l'exécuter, rebootez la cible et tapez la commande « **netload** ».

uClinux - Mise en oeuvre

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

```
xavier@localhost:~/uClinux-pragm-20060217/images - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
xavier@localhost:~/uClinux-pragm-20060217/images
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.rom
-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.rom
-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.rom
```

Côté cible, opérez comme suit :

```
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]
```




Bâtiment EARHART
ZAC Grenoble Air Parc

38590 St Etienne de St Geoirs - France

www.pragmatec.net

uClinux - Mise en oeuvre

Tous les programmes livrés avec votre distribution uClinux se trouvent sous /bin.

Utilisez le répertoire /var pour y stocker vos informations ou vos programmes temporaires (ils auront disparu au prochain redémarrage de la carte).

Les répertoires /home et /usr permettent de conserver toutes les données qui y sont inscrites, tout comme le ferait un disque dur. Stockez y les informations que vous souhaitez conserver.

uClinux - Mise en oeuvre

4 Installation d'Eclipse



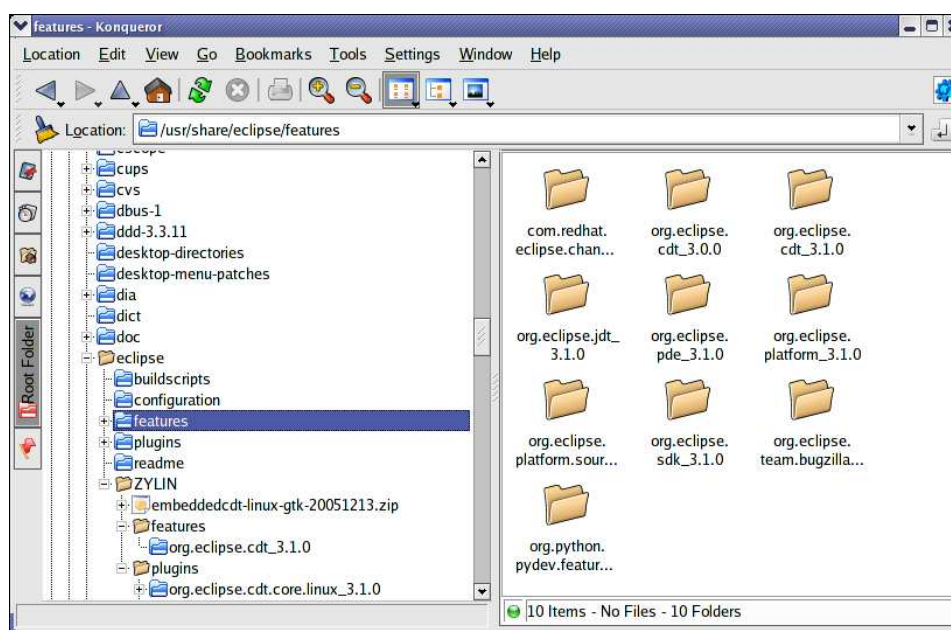
Eclipse est un environnement de développement intégré ouvert et gratuit. Il permet de gérer des projets complets de façon graphique, et y compris de débiter le code développé.

Nous allons utiliser Eclipse pour développer nos programmes et pour les débiter directement depuis la cible via le réseau Ethernet.



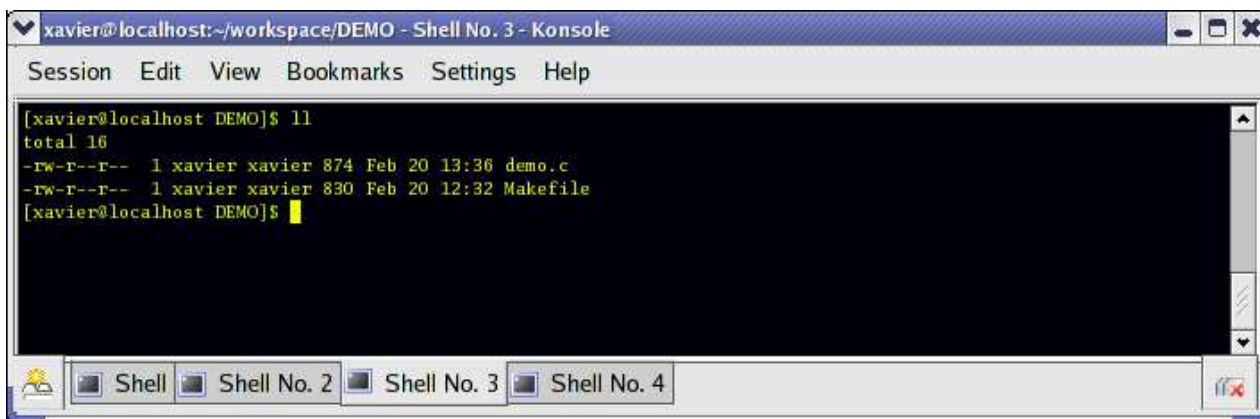
Le problème d'Eclipse est qu'il n'est pas conçu à la base pour effectuer un débiter distant via le réseau Ethernet, et pour corriger ce problème il est nécessaire d'y ajouter un correctif sous la forme d'un plugin.

Pour cela utilisez le plugin ZYLIN qui se trouve sur le CDROM (**embeddedcdt-linux-gtk-20051213**). Copiez les répertoires « features » et « plugins » sous les répertoires correspondant de Eclipse (présent sous **/usr/share/eclipse**).



uClinux - Mise en oeuvre

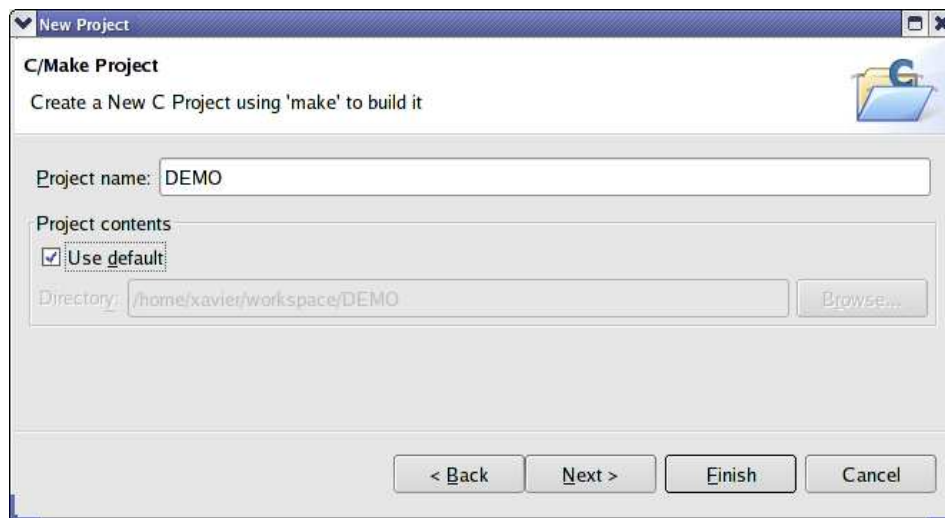
Ceci étant fait commencez par copier le projet DEMO sous votre workspace Eclipse (par exemple /home/xavier/workspace).



```
xavier@localhost:~/workspace/DEMO - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help

[xavier@localhost DEMO]$ ll
total 16
-rw-r--r-- 1 xavier xavier 874 Feb 20 13:36 demo.c
-rw-r--r-- 1 xavier xavier 830 Feb 20 12:32 Makefile
[xavier@localhost DEMO]$
```

Vous allez à présent créer le projet sous Eclipse (**File/New/Project...Standart Make C project**) :



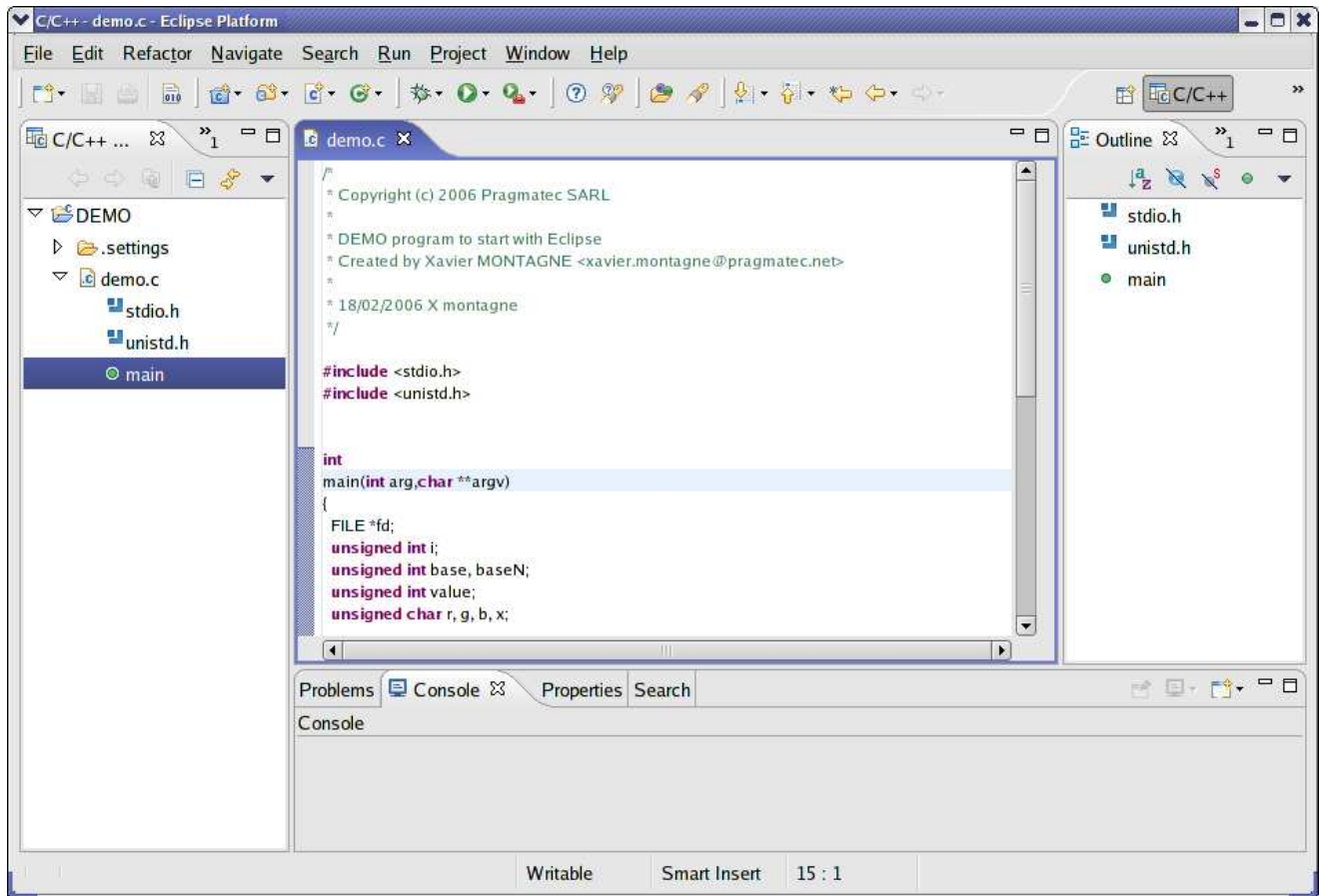
Puis cliquez sur « **Finished** ».

Importez les fichiers du projet fourni en exemple dans votre environnement Eclipse. Pour cela cliquez à l'aide du bouton droit sur le nom de votre projet et choisissez « Import... ».

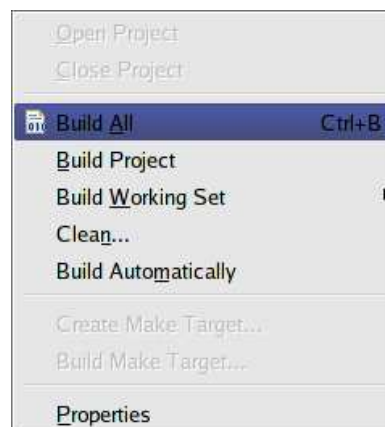
Faites de même pour activer le sous-menu « Build make target... » et sélectionnez les modes de compilation (all et clean devraient suffire).

uClinux - Mise en oeuvre

Vous devez obtenir la vue suivante :



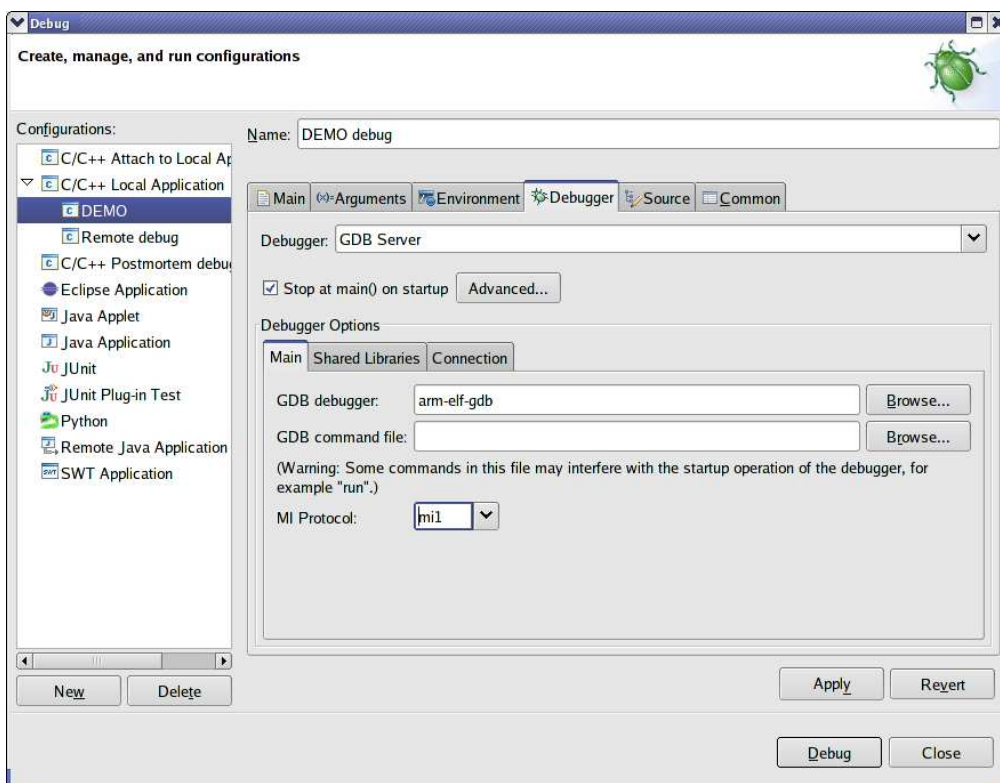
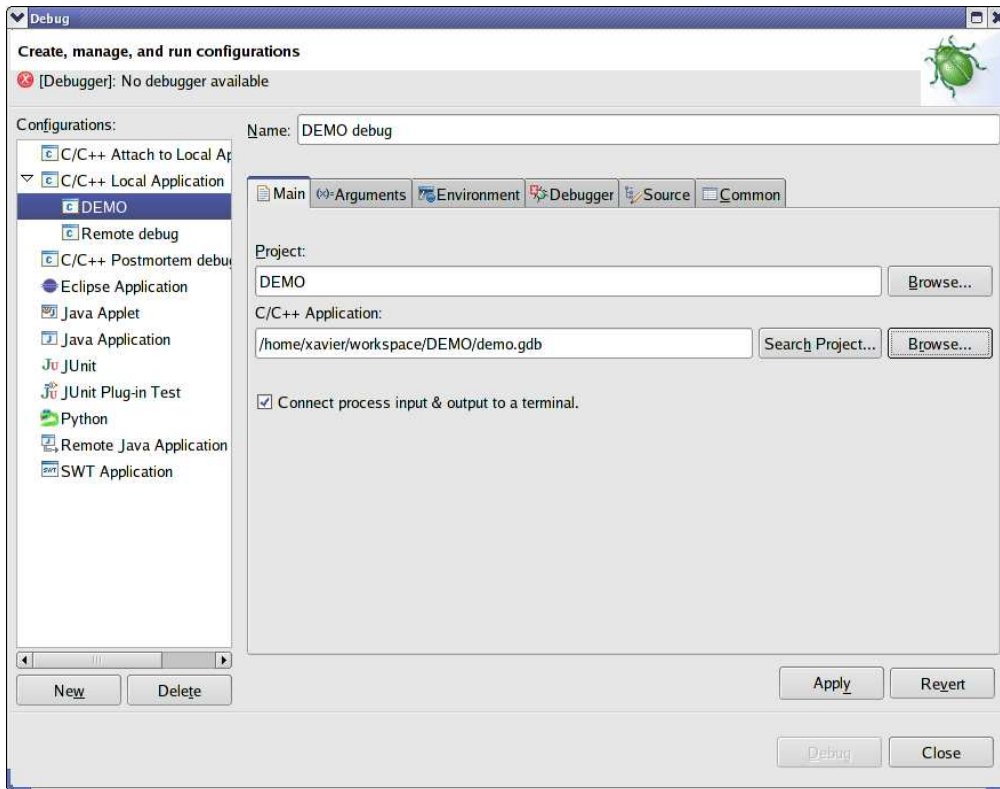
Enfin compilez votre projet à l'aide du menu « **Project** » :



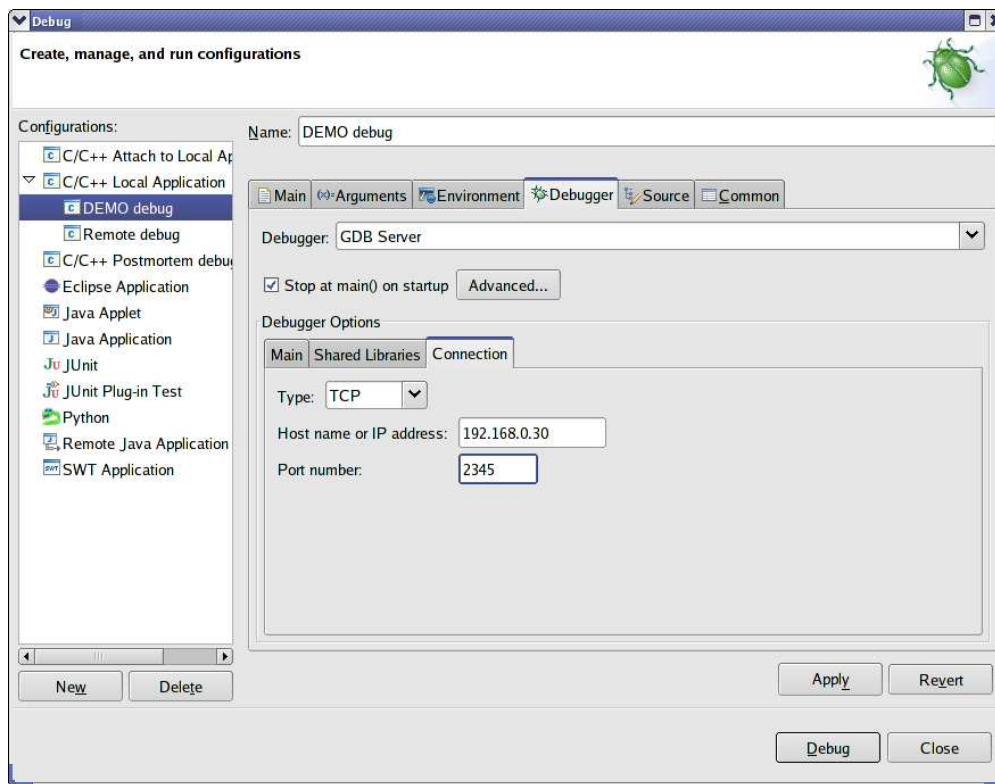
L'affichage dans la fenêtre « Console » doit vous renseigner sur le résultat de la compilation.

uClinux - Mise en oeuvre

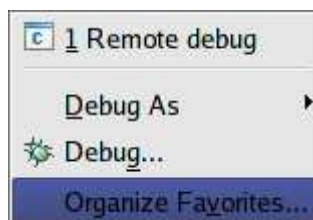
Dans la nouvelle fenêtre, sélectionnez « C/C++ Local application » et renseignez la nouvelle perspective comme suit :



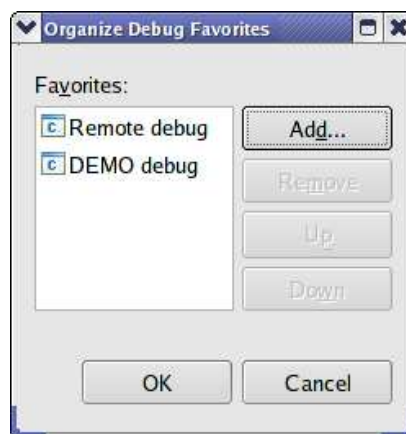
uClinux - Mise en oeuvre



Il faut maintenant ajouter votre nouvelle perspective en cliquant à nouveau sur la flèche descendante de l'icône de débogage :



Puis cliquez sur « **Add...** » :



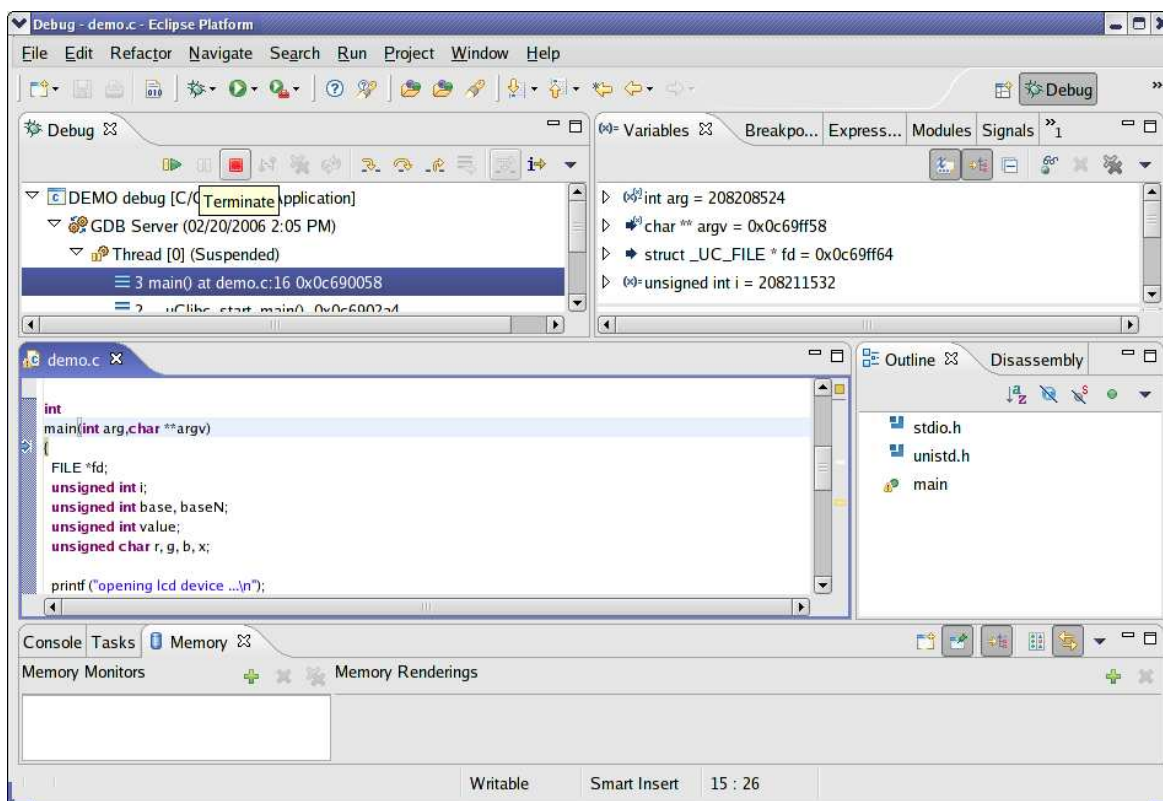
uClinux - Mise en oeuvre

Vous pouvez maintenant changer de perspective en cliquant sur les doubles flèches qui se situent en haut et à droite d'Eclipse. Sélectionnez la perspective « **Debug** ».

Vous pouvez à présent tenter une connexion à la cible en mode debug en cliquant encore une fois sur la flèche descendante de l'icône debug, puis en sélectionnant le menu « **DEMO debug** » :



Vous devez obtenir la vue suivante sous Eclipse (débugger arrêté sur le main) :



Vous pouvez à présent vous déplacer en pas à pas en cliquant sur la flèche du milieu de la fenêtre Debug, inspecter les variables locales à la fonction courante, placer des breakpoints... N'hésitez pas à manipuler le débogueur sous Eclipse, quitte à relancer le débogueur plusieurs fois, en prenant bien soin de lancer **D'ABORD** le débogueur sur la cible à l'aide la commande « **./debug** ».

uClinux - Mise en oeuvre

Vous pouvez aussi directement modifier le code dans la perspective de debug et recompiler le code à la volée. Ensuite relancer « **.debug** » sur la cible pour transférer le nouveau programme automatiquement et lancer le mode debug.

L'application utilise le framebuffer qui est la mémoire vidéo primaire. Au démarrage de votre noyau Linux vous avez pu peut-être constaté qu'il y avait un logo Pragmatec qui s'affichait sur le LCD 256 couleurs. Le programme demo n'a d'autre but que d'essayer d'effacer le logo Pragmatec. Au bout d'un certain temps voici le résultat obtenu :



Avant



Après

uClinux - Mise en oeuvre

5 Programme DEMO_IHM

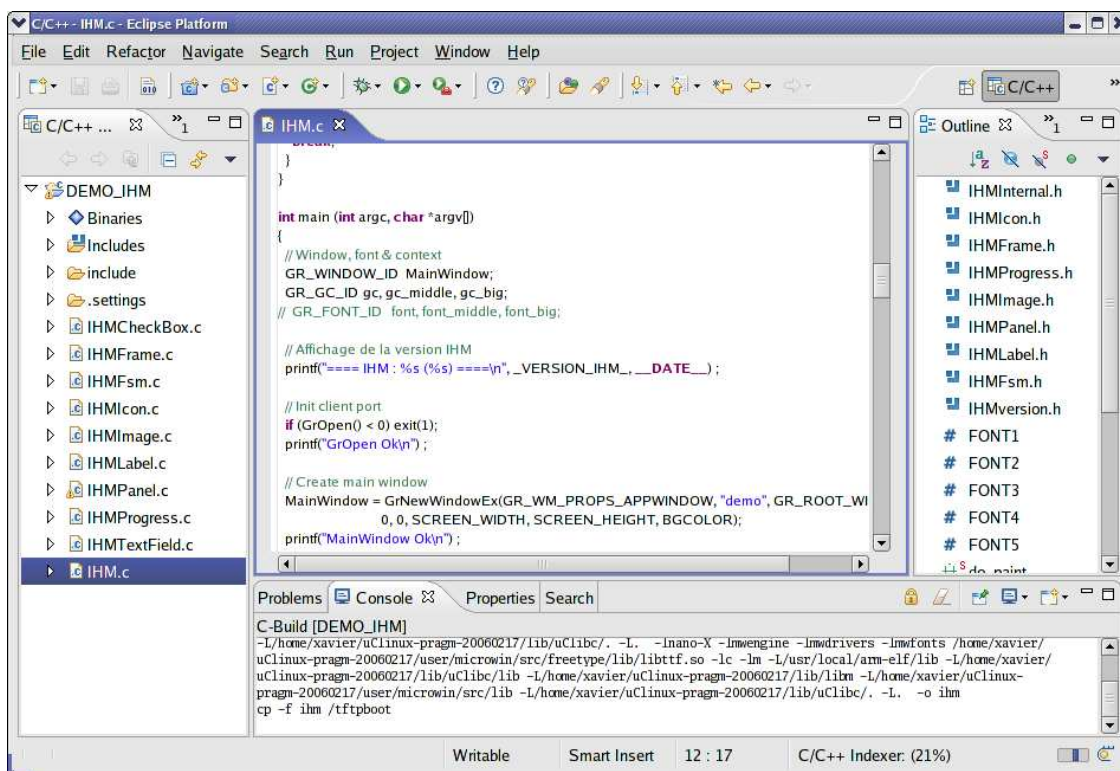


Le programme précédent était un exemple simple dont l'objectif était de démontrer la relative simplicité qu'il y a à débiter un programme sur cible avec Eclipse.

Pourtant ce programme semble quelque peu limité, au vu de ce qu'il est possible de faire avec uClinux et un LCD connecté à notre ARM7.

Pour voir l'étendu des possibilités graphiques qu'offre uClinux, créez un nouveau répertoire dans votre workspace Eclipse : DEMO_IHM... puis recopier le projet DEMO_IHM.

Créez un nouveau projet sous Eclipse, ainsi qu'une nouvelle perspective de debug propre à ce projet. Vous devez obtenir le résultat suivant :



Sur la cible, créez un répertoire **IHM** sous **/usr**.

Transférer le fichier « **arialb.ttf** » présent sous **uClinux-pragm-20060217/user/microwin/src/fonts/**.

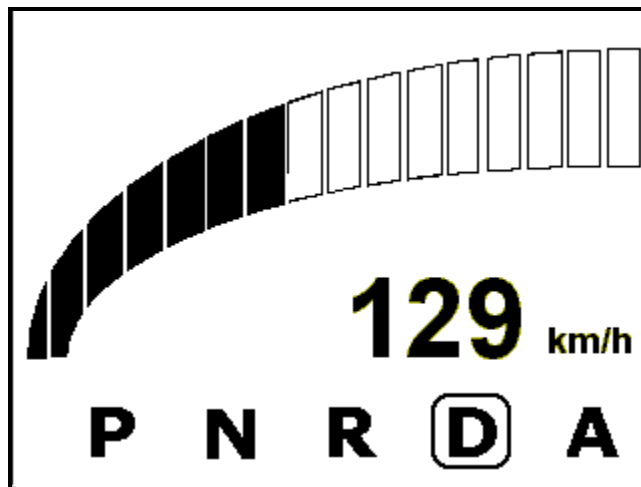
Sous IHM, il convient de transférer toutes les images GIF du projet. Pour cela vous pouvez utiliser la commande « **tar** » sur la station : « **tar cvf images.tar *.gif** ». Puis copiez le fichier **images.tar** sous **/tftpboot**, et transférez le sur la cible à l'aide de la commande **tftpget** sous IHM (« **tftpget 192.168.0.10 get images.tar**»). Enfin désarchivez les images sur la cible : « **tar xvf images.tar** ».

uClinux - Mise en oeuvre

Afin de pouvoir débiter sur la cible n'oubliez pas de créer un fichier « **.debug** » qui doit faire référence au programme appelé « ihm » à présent et non plus « demo ».

Le programme DEMO_IHM s'appuie sur Nano-X, un serveur X extrêmement allégé qui permet d'afficher des images, du texte et des objets graphiques sur un LCD via le framebuffer.

L'affichage sur le LCD présente une simulation d'un afficheur LCD de voiture :



uClinux - Mise en oeuvre

6 Informations complémentaires



Ce chapitre présente un certain nombre d'informations utiles liées à l'utilisation de votre carte, dans le cas où disposez d'une connexion réseau avec votre carte et pour tout ceux qui ne possèdent pas de LCD graphique.

Connexion par telnet

Votre carte possède un système d'exploitation uClinux 2.4 assez complet qui vous permet entre autre de vous y connecter par Telnet. Ceci permet par exemple d'ouvrir un shell distant sur la cible sans avoir besoin d'utiliser un cordon série.

Dans une entreprise cela vous permet aussi d'installer la carte à un endroit éloigné du poste de travail, en la connectant au réseau d'entreprise Ethernet.

```
Telnet 192.168.0.30
PRAGMATEC login: root
Password:
Welcome to

          uClinux

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.

[~]ls -l /dev/i2c*
crw----- 1 root   root   89,  0 Jan 1 1970 /dev/i2c0
[~]
```

Vous pouvez ouvrir une session Telnet sous Windows™ en cliquant sur le bouton « Démarrer » puis en sélectionnant le menu « Exécuter... ».

Tapez alors « telnet 192.168.0.30 » si l'adresse IP de votre carte est 192.168.0.30.

uClinux - Mise en oeuvre

Transfert FTP

Pour transférer vos fichiers sur la cible (images, programmes, ...) nous avons utilisés la commande « `ftppget` ». Il est aussi possible de vous connecter à la cible par FTP et d'effectuer ainsi un transfert FTP du ou des fichiers vers la cible ou depuis la cible :



```
C:\WINNT\system32\ftp.exe
Connecté à 192.168.0.30.
220 Pragmatec FTP server (GNU inetutils 1.4.1) ready.
Utilisateur (192.168.0.30:(none)) : root
331 Password required for root.
Mot de passe :
230-
230- Welcome to
230-
230-
230-
230-
230-
230-
230-
230-
230-
230-
230-
230-
230-
230-
230-
230- For further information check:
230- http://www.uclinux.org/
230- http://www.pragmatec.net/
230 User root logged in.
ftp>
ftp>
```

Vous pouvez ouvrir/démarrer une connexion FTP sous Windows™ en cliquant sur le bouton « Démarrer » puis en sélectionnant le menu « Exécuter... ».

Tapez alors « `ftp 192.168.0.30` » si l'adresse IP de votre carte est 192.168.0.30.

Tapez la commande « `help` » sous le prompt FTP pour connaître l'ensemble des commandes autorisées.

Sachez toutefois que les 4 commandes les plus utilisées sont :

- **cd** : précise le répertoire courant sur la station
- **lcd** : précise le répertoire courant sur la cible
- **put** : permet de transférer un fichier de la station vers la cible
- **get** : permet de récupérer un fichier depuis la cible

Des variantes de `put` et de `get` permettent aussi de transférer automatiquement plusieurs fichiers dans l'une ou l'autre des directions.

uClinux - Mise en oeuvre

De plus si vous utilisez les font TRUETYPE avec votre projet il faut copier la font choisie dans votre homedirectory. Par exemple dans le cadre du projet DEMO_IHM qui utilise la font « arialb.ttf » vous devez copier ce fichier situé sous « uClinux-prag-20061017/user/microwin/src/fonts/truetype/ » dans /home/xavier (votre compte correspondant à la variable d'environnement \$HOME).

Sur la cible vous devez faire de même pour préciser au client qu'il va devoir envoyer ces consignes via le réseau. Tapez :

- **export REMOTE=1**
- **export NXDISPLAY=192.168.0.10**

Ensuite lancez votre programme client sur la cible.

Par exemple vous trouverez dans le répertoire « bin » de nano-X (par exemple « uClinux-prag-20060829/user/microwin/src/bin ») le programme « nxview » qui permet d'afficher une image sur le LCD ainsi que l'image « tux.gif ».

Transférez les sur la cible à l'aide de « tftpget », « FTP » ou « NFS » et lancez le programme comme suit :

- **nxview tux.gif**

Vous devriez obtenir sur la station de travail l'affichage suivant :

