Démarrage de la carte Olimex EP930X

Out of the box

Installez gtkterm et le lancer par $gtkterm &
Utilisez un câble **croisé** sur le port RS232_0
configuration du terminal : 57600,8,N,1 pas de contrôle de flux.
Alimentez la carte en **5V regulé** (pas un simple transformateur sortant approximativement 5V) la borne positive est au centre.
Attention cette alimentation va directement sur les ports usb.
NE PAS UTILISER UNE ALIMENTATION NON REGULEE
(pas de simple transfo comme utilisé pour les pics !)

Linux boote, le root file system est un ramdisk.
Configurez l'interface ethernet par ifconfig eth0 pour que la carte soit accessible.
Vérifiez que le serveur web de la carte est accessible.

On obtient donc un système robuste (ramdisk) directement utilisable mais comme il utilise un ramdisk, il faut un périphérique de stockage supplémentaire (carte sd, clé usb,...) pour conserver programmes supplémentaires et données.

Mise en place de la chaîne de compilation µclibc

Décompressez l'archive arm-linux-gcc-4.1.1-920t.tar.bz2
Dans votre terminal ajoutez au PATH le chemin vers le compilateur :
(PATH=$PATH:....; export PATH)
Créer un fichier source c type HelloWorld.c et le compiler par
arm_linux-gcc -Wall -oHello HelloWorld.c
Il faut maintenant transférer le binaire Hello obtenu sur la cible arm.
Le plus simple est d'avoir un serveur web sur votre machine (boa,apache,..)
et de placer le fichier Hello dans le répertoire des documents Web du serveur.
$sudo mv Hello /var/www
Ensuite sur la cible (depuis gtkterm) on le transfère par wget:
#wget http://192.168.0.202/Hello

Puis lui donner les droits d'exécution et l'exécuter :
#chmod +x Hello
#./Hello

Création d'un système complet sur une clé usb

Running a full linux distribution from a flash drive
First you should get a flash drive of sufficient capacity - a 256MB one will suffice. Create an ext2 filesystem on it (make sure it is the first primary partition, e.g. sda1) and mount it. Then change to the mount directory and extract the rootdir.tgz arhive. WARNING! this archive extracts to the current directory and not to a sub-directory. Make sure you are in the mount directory prior to extracting.
After extraction is finished, unmount the flash drive and plug it into the board. Interrupt the boot process and issue the following commands:

```
RedBoot> fis load zImage
RedBoot> exec -c "console=ttyAM0 root=/dev/sda1 rootdelay=10"
```

The root password is "olimex".

- configurez le réseau et vérifiez que la carte accède bien à internet.
- connectez vous en ssh en root sur la carte et configurez le réseau de manière permanente avec l'éditeur vi (/etc/network/interfaces /etc/resovl.conf).
- ajouter un utilisateur (commande adduser).
- connectez vous en ssh avec l'utilisateur crée.
- Mettez à jour votre système (en root): apt-get update puis apt-get upgrade (c'est long!)

- Pendant ce temps, vous pouvez préparer la chaîne de compilation croisée.

- Décompressez l'archive crosstool-linux-gcc-4.0.1-glibc-2.3.5.tar.bz2 dans /opt/crosstool et ajoutez le chemin vers le gcc qui convient dans votre path. Compilez un helloworld.c et vérifiez avec objdump -x que le binaire est bien au format arm et qu'il utilise bien la librairie partagée glibc et pas uclibc.
Transférer le binaire sur la carte (avec rcp par exemple) et exécutez-le.

Téléchargement d'un noyau sur la cible

Running other linux versions
If you want to run other version of linux than the supplied one, you should interrupt the Redboot process by pressing ctrl-C. You will be dropped to the redboot prompt and can use it for loading your custom system. You can do that through either the serial port or using the on-board ethernet. The serial port is universally available but is slow (4-5KB/s) while the network loading is fast but you should configure a server(bootp or http) on your network.

To load a new system using the serial port type the following
RedBoot> load -v -r -b <base_adress> -m ymodem
In your terminal program(HyperTerminal) choose Transfer->Send File. Make sure the protocol is set to "Ymodem" and use the "Browse" button to locate the required file. Click the send button. A progress bar should appear informing you about the ongoing transfer

Pour le chargement par le port série, gtkterm n'est pas suffisant car il ne gère pas les protocoles de transfert binaire (ymodem,zmodem,etc..). Sous linux, on peut utiliser minicom mais il faut se faire à son interface un peu datée.
Nous allons plutôt faire un chargement par le réseau ethernet :
Configurer le réseau dans Redboot (ip_address) puis :

To load a file from the network you should set up a server (bootp or http) on your network and place the file in the server's data directory. Then power up the board and interrupt the startup script. Use the following commands:
RedBoot> load -v -r -b <base_adress> -h <server_adress> <filename>
for bootp loading, or
RedBoot> load -v -r -b <base_adress> -m http -h <server_adress> <filename>
for http loading

Charger le noyau zImage9302_2.6.21 par cette méthode et exécutez-le.

Il est bien sûr possible de sauvegarder ce noyau et la configuration de boot en flash pour éviter d'avoir à refaire cette manip à chaque démarrage mais NE LE FAITES PAS car une fausse manip peut rendre la carte inutilisable.
(Je donne en devoir noté la recompilation de redboot et le flashage par le port JTAG à celui qui me modifiera la configuration en flash de la carte).

Construire son propre noyau

A partir de sources et d'un patch fournit par Peter Ivanov (http://dev.ivanov.eu/projects/cs-e9302/), on se propose de recompiler un noyau pour y inclure les supports des cartes MMC/SD,des modules et d'une webcam Philips.

Décompressez les sources du noyau (linux-2.6.24.7.tar.bz2) et appliquez le path linux-2.6.24-rc8_ep93xx_mmc.patch.gz.

Exécuter "make clean" et ensuite il suffit d'appliquer la méthode suivante :

Building a custom Linux kernel
You can use either the stock kernel (2.6.21 as of writing) with a small patch for the ethernet or use the Cirrus patches. The stock kernel is updated more often but lacks support for some specific devices on board (e.g. IrDA, SPI flash) while the Cirrus one has all the drivers but is updated less often and is supported by Cirrus only. Either way, if you want to build the kernel only, first make sure you have a working cross compiler in your PATH. Then start with some of the default configs, e.g. copy the corresponding .config_xxx file to the kernel directory, rename it to .config and issue:

```
$ make ARCH=arm CROSS_COMPILE=arm-elf- oldconfig
```

Substitute the "arm-elf-" part with whatever your cross-compiler's prefix is. Then you can actually build the kernel with the following command:

```
$ make ARCH=arm CROSS_COMPILE=arm-elf- zImage
```

After some time the kernel should be available as arch/arm/boot/zImage. You can upload this new kernel to the board in any of the ways described above.

En chargeant le paquet libncurses5-dev vous pourrez utiliser le make menuconfig qui est plus convivial.
Téléchargez votre nouveau noyau et bootez avec.
Vérifiez qu'il s'agit bien de votre nouveau noyau avec uname -a.
Vérfiez que la webcam philips est bien reconnue.

Accès au entrées/sorties logiques de la carte depuis l'espace utilisateur

- En étudiant la documentation du EP9302 et le programme exemple button_led_demo.c, expliquez la méthode d'accès aux registres du µP et les particularités des E/S logiques de ce circuit.

- Ecrivez un formulaire html et un programme cgi pour commander les led RED et GREEN par des boutons depuis un navigateur. Renseignez également l'utilisateur sur l'état du bouton BUT.

Utilisation du second port série

Nous souhaitons utiliser le second port série pour y connecter un équipement avec lequel nous souhaitons communiquer (gps, module zigbee, etc..).
Avec le noyau livré, le second port série ne fonctionne pas car le bit U2EN n'est pas à 1 au reset. (bit 21 de DeviceCfg voir doc EP9302).

- Écrivez un programme qui active l'UART 2 (/dev/ttyAM1) et vérifiez-le en vous connectant sur le port série correspondant.

Par défaut ce port est prévu pour y connecter un terminal et pas pour un équipement quelconque.
Il faut donc supprimer getty sur ttyAM1 dans /etc/inittab.

- Vérifier que ça fonctionne en remcompilant vos programmes du tp sur le port série pour échanger des données dans les deux sens avec votre pc via ce port.