

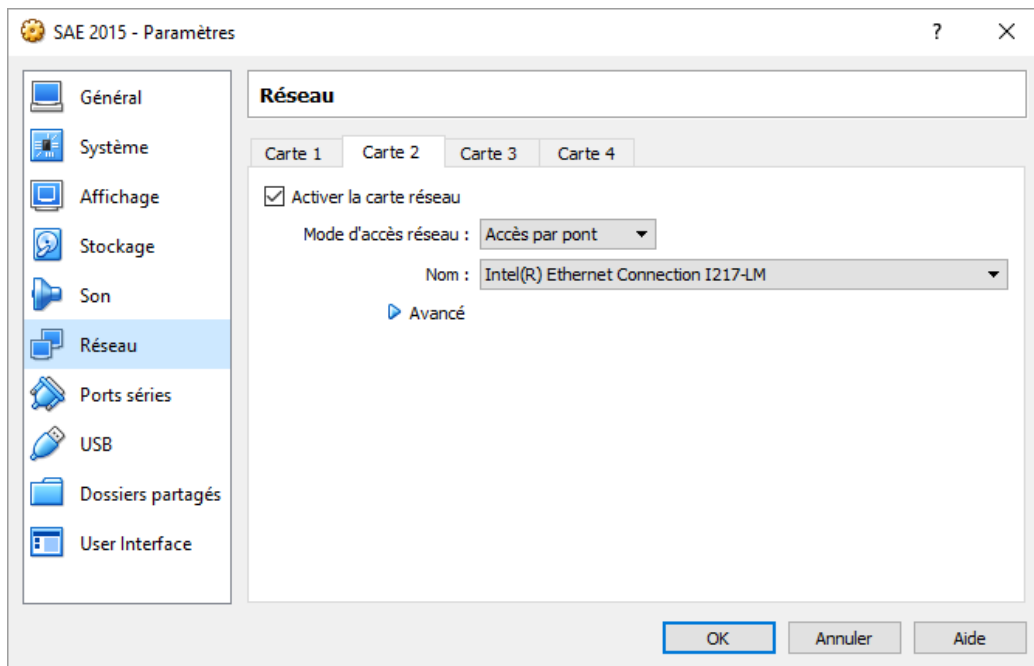
TD n° 11

Système Embarqué : NSLU 2 et OpenWRT

1 Préambule pour la configuration réseau de votre machine virtuelle

Avant de lancer VirtualBox, vous devez effectuer la configuration réseau suivante. En effet, durant le TD, nous allons devoir communiquer avec le NSLU2 et pour ce faire, nous avons besoin de deux interfaces réseau. La configuration s'effectue de la manière suivante :

- Vérifiez que votre machine est bien arrêtée (et non en veille prolongée).
- Aller dans la rubrique Configuration / Réseau
- Vous devez avoir une première carte réseau configurée en NAT
- Vous devez ajouter une deuxième carte réseau en « Accès par pont » (Bridge) et configurer celle-ci sur l'interface filaire de votre machine virtuelle



- Démarrez votre machine virtuelle (il est fort possible que celle-ci se bloque pendant un moment. Si c'est le cas, attendre le timeout de la recherche réseau).
- Une fois votre système démarré, allez dans les règles udev pour votre système pour ajouter une règle sur la détection de votre nouvelle carte réseau (`/etc/udev/rules.d/70-persistent-net.rules`).
- Dupliquez la règle existante et modifiez l'adresse MAC pour mettre celle de votre interface (paramètre avancés de l'image ci-dessus) et modifiez le nom en `eth1`. Vous devriez ainsi avoir une ligne supplémentaire du type :


```
SYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}="08 :00 :27 :.. :.. :..",
ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"
```
- Redémarrez votre système avec cette nouvelle configuration pour la deuxième interface réseau. Vous devrez avec l'interface `eth0` qui utilise votre interface Wifi et vous permet d'accéder au réseau Internet et l'interface `eth1` connectée sur votre interface filaire et qui nous permettra de nous connecter au NSLU2.

TD n° 11

Système Embarqué : NSLU 2 et OpenWRT

Le but de ce TD est de vous familiariser avec le monde de l'embarqué et l'installation et la configuration d'une distribution GNU/Linux sur ce type de cible ainsi que d'y ajouter des fonctionnalités. Nous verrons dans un deuxième temps comme réaliser les différentes étapes qui conduisent à produire ces paquetages pour la cible embarquée.

1 Configuration de votre environnement de travail

Pour réaliser l'opération de compilation complète d'un firmware, vous devez disposer d'au moins 3.5Go d'espace sur la partition sur laquelle vous travaillerez.

1.1 Outils nécessaires à la compilation de l'environnement

Les outils suivants sont nécessaires sur votre machine de développement (dans la machine virtuelle) pour compiler l'environnement de développement. Vérifiez que vous disposez bien de :

```
tar unzip bzip2 zlib1g-dev gawk bison flex autoconf make gcc g++ subversion
```

1.2 Environnement de développement

Vous téléchargerez un disque virtuel contenant la distribution sur laquelle nous allons travailler :

```
https://trolen.polytech.unice.fr/cours/sae/td11/sdd-openwrt.7z
```

Cette image comprend l'environnement de développement OpenWRT qui va permettre de créer une image firmware à télécharger sur votre NSLU. La version sur laquelle nous travaillerons est la distribution *backfire* en version 10.03. Vous veillerez à monter cette image dans le dossier `/work` de votre machine virtuelle de travail.

Si vous ne récupérez pas directement le disque virtuel, vous pouvez télécharger indépendamment la distribution de *backfire* à partir de l'adresse suivante :

```
svn co svn://svn.openwrt.org/openwrt/branches/backfire
```

2 Création de son propre firmware

Pour cette phase, vous devez travailler en tant qu'utilisateur standard (login `user`, mot de passer `user`) et pas en tant que super utilisateur (`root`). Quoi qu'il arrive, si vous oubliez de vous connecter en tant qu'utilisateur `user`, la première phase de compilation vous le rappellera (mais vous devrez effacer tous les fichiers créés en tant que super utilisateur). Si vous avez lancé la compilation en tant que `root`, la première phase crée un dossier `tmp` qu'il faudra effacer avant de relancer la compilation en tant que simple utilisateur.

2.1 Configuration de Backfire

Après avoir récupéré OpenWRT (version *backfire*), vous procéderez à sa configuration. Pour cela les commandes suivantes vous seront utiles (comme pour la configuration du noyau) :

```
cd backfire && make menuconfig
```

Cette commande `make menuconfig` commencera par vérifier que vous avez tous les outils nécessaires à votre distribution pour pouvoir lancer la compilation. Elle permet aussi de sélectionner les packages qui seront compilés et placés dans l'image finale. Trois choix sont possibles :

- `<>` (touche n/N) le paquet concerné ne sera pas compilé
- `<M>` (touche m/M) le paquet concerné sera compilé mais pas inclus dans l'image
- `<*>` (touche y/Y) le paquet concerné sera compilé et installé dans l'image

Vous pouvez ainsi configurer tout ce qui se trouvera dans votre firmware. Le but du jeu étant bien sûr de ne pas dépasser les 8Mo de la mémoire flash, et même d'obtenir l'image la plus petite possible pour pouvoir ajouter de nouveaux logiciels une fois cette image installée.

TD n° 11

Système Embarqué : NSLU 2 et OpenWRT

Il semble que beaucoup d'options mènent vers une compilation qui échoue. Il faut principalement éviter de toucher aux options pour lesquelles il est écrit « *(for developers)* ».

Pour être sûr d'aboutir à une configuration fonctionnelle le plus rapidement possible, vous récupérez le fichier de configuration que nous mettons à votre disposition dans le dossier `backfire` : `config-backfire`.

Après l'avoir copié dans le répertoire `backfire` sous le nom `.config`, vous veillerez à charger ce fichier de configuration (`make oldconfig`). Dans la configuration, nous avons changé l'adresse IP par défaut pour qu'elle soit à la valeur `192.168.1.77` (au lieu de `192.168.1.1`).

2.2 Configuration du noyau

Une configuration par défaut du noyau est déjà réalisée dans la distribution de `backfire`. Dans un premier temps utilisez celle-ci. Mais il est aussi possible de configurer le noyau de l'image

```
make kernel_menuconfig
```

Pour mettre votre propre noyau, vous devez : décompresser les sources du noyau, appliquer les patches souhaités, et lancer la configuration dans l'environnement de compilation croisée.

Dans notre cas de figure, la taille du noyau n'est pas aussi importante car l'image (firmware) est séparée en 4 sections, le boot loader, le microcode, le noyau et l'image disque, qui ont des adresses fixes. Donc la taille ne doit juste pas dépasser l'espace alloué au noyau. C'est à vérifier. Mais pour un contrôle maximal sur ce qui se passe dans le système il est toujours agréable de configurer soi-même le noyau, ou au moins de vérifier que la configuration de base est correcte, principalement pour les périphériques que nous voudrions utiliser.

Pour avoir un système fonctionnel le plus rapidement possible, vous pouvez utiliser la configuration du noyau par défaut qui doit fournir un noyau fonctionnel pour NSLU2.

3 Compilation d'un nouveau firmware

Comme d'habitude, pour compiler, il suffit de lancer la commande :

```
make -j X
```

ou X est le nombre de cœurs disponible dans votre machine virtuel multiplié par deux.

Dans le cas où la compilation s'arrête, pour obtenir les messages d'erreur de compilation, il faut taper :

```
make V=99
```

Par défaut, l'affichage est concis, joli mais ne dit rien en cas d'échec de compilation. L'option `V=99` permet d'avoir les messages standards de compilation.

Lancez la compilation sur votre PC. Cependant, elle peut prendre plusieurs heures (1h30 sur dual core), à cause de la compilation nécessaire des bibliothèques (uClibc, ...), de l'environnement de compilation croisée (gcc pour la cible), un noyau linux ainsi que de tous les paquetages qui ont été configurés. Si à la fin du TD vous avez réussi à obtenir une image, vous pourrez re-flasher le NSLU2 pour l'installer avec le résultat de votre propre compilation.

L'image créée sera dans le dossier :

```
bin/ixp4xx/openwrt-nslu2-squashfs.bin
```

Les paquetages compilés seront quant à eux dans le dossier :

```
bin/ixp4xx/packages/*
```

Pour poursuivre le TD dès à présent, pendant que la compilation s'effectue, nous vous fournissons une image firmware déjà compilée (ce que vous obtiendrez à la fin de votre propre phase de compilation). Vous trouverez celle-ci dans le répertoire `/work/openwrt` sous le nom `openwrt-nslu2-squashfs.bin`. Mais quoi qu'il arrive vous aurez besoin de votre chaîne de compilation croisée pour la cible pour réaliser la dernière partie de ce TD.

TD n° 11

Système Embarqué : NSLU 2 et OpenWRT

L'image du système déjà compilé est à installer en attendant comme expliqué dans la section suivante.

4 Communication avec le NSLU

4.1 Avant de tout casser, utilisez le NSLU2 à sa sortie de la boîte

Avant de commencer à modifier le firmware, nous allons déjà tester que nous arrivons bien à communiquer avec le matériel. Vous connecterez le NSLU2 avec votre ordinateur par la prise réseau avec le câble fourni. L'adresse par défaut du NSLU2 est 192.168.1.77. Vous veillerez à configurer l'interface réseau (`eth1`) de votre machine virtuelle à l'adresse 192.168.1.100 pour permettre la communication avec le NSLU.

```
ifconfig eth1 192.168.1.100 netmask 255.255.255.0 up
```

Vous pourrez tester que vous accédez bien à l'interface Web du dispositif depuis votre machine virtuelle de travail grâce au navigateur de votre choix.

Quoi qu'il arrive, il est nécessaire de faire un boot et d'éteindre avant de vouloir flasher sinon il n'entre pas en mode de mise à jour.

4.2 Installation du firmware fourni (ou celui que vous avez compilé)

4.2.1 Outil Linux pour la mise à jour du firmware

Vous pouvez faire la mise à jour du firmware depuis votre machine virtuelle sous Linux grâce à l'outil `upslug2`. Attention toutefois à la configuration réseau. Sur la nouvelle machine virtuelle fournie, l'interface `eth1` doit être configurée avec l'adresse 192.168.1.100.

Avant de tenter l'upgrade, vérifiez que vous arrivez bien à contacter le NSLU2 avec la commande suivante : `upslug2 -d eth1`. Si tout est correct, vous pouvez passer à l'étape suivante.

4.2.2 Procédure de mise à jour du firmware

Commencez par mettre le NSLU2 en mode de mise à jour. Pour cela, suivez la procédure suivante. Le NSLU2 étant éteint, appuyez sur le bouton « Reset Hardware » au dos de l'appareil (prévoir un trombone, une épingle ou tout objet assez fin pour entrer dans le trou), allumer le NSLU (appuyer et relâcher) tout en maintenant le bouton reset appuyé pendant 10s environ. Relâchez alors rapidement le bouton reset dès que la couleur de la lumière power change légèrement du jaune à l'orange au bout de ce laps de temps. La LED doit alors clignoter en Orange/Vert, signe que le matériel est bien en attente de téléchargement d'un nouveau firmware. Vous pourrez aussi consulter la documentation située à l'adresse suivante en cas de problème :

<http://www.nslu2-linux.org/wiki/HowTo/UseTheResetButtonToEnterUpgradeMode>

Il ne vous reste alors plus qu'à lancer le flash du nouveau firmware :

```
upslug2 -d eth1 --image openwrt-nslu2-squashfs.bin
```

4.3 Se connecter sur le NSLU2

A la première tentative, la connexion devra se faire via `telnet` sur le NSLU.

```
telnet 192.168.1.77
```

Un message vous signalera qu'il faut obligatoirement donner un mot de passe pour `root` pour pouvoir vous connecter en `ssh` sur la cible. Vous configurerez obligatoirement le mot de passe à la valeur suivante à l'aide de la commande `passwd` dans la console.

```
Login: root
Mot de passe: iam-2016
```

Vous pouvez vous déconnecter et vous reconnecter par `ssh` sur le NSLU.

TD n° 11

Système Embarqué : NSLU 2 et OpenWRT

5 Utilisation et création de paquetages

5.1 Gestion de l'extension Wifi USB : Tout est prêt à l'emploi

Le but principal de l'installation d'un nouveau firmware est de pouvoir gérer de nouveaux matériels connectés à notre plate-forme. Dans le fichier de configuration fourni, nous avons inclus le support pour une extension Wifi USB.

5.1.1 Prise en charge du nouveau matériel

Lorsque vous connectez l'adaptateur USB Wifi D-Link DWL-G122, celui-ci doit être reconnu par l'apparition d'un message sur la console ou dans les logs.

```
usb 3-1: new high speed USB device using ehci_hcd and address 2
usb 3-1: configuration #1 chosen from 1 choice
```

Vous pourrez aussi voir à l'aide de la commande `iwconfig` les informations suivantes :

```
wlan0 IEEE 802.11bg ESSID: ""
```

Il ne vous reste plus qu'à configurer la connexion pour accéder au Wifi.

```
ifconfig wlan0 up
iwlist wlan0 scan | grep Cell
```

Ensuite, pour la gestion de la connexion, il faut choisir à quel réseau se connecter :

```
iwconfig wlan0 essid SSID_NAME (vous pouvez tenter ETUD_POLYTECH)
```

Puis, régler le chiffrement si nécessaire, obtenir une IP, mettre en place les bonnes routes, et éventuellement déconnecter l'Ethernet pour être sûr qu'on s'y connecte par wifi.

5.1.2 Configuration pour se connecter à un réseau sécurisé par WPA

Dans le cas où vous vous connectez à un réseau sécurisé, tout ce qu'il faut pour le WPA est fourni (`wpa_supplicant`). Contenu du fichier `/etc/wpa_supplicant.conf`:

```
network={
    ssid="Ubiquarium"
    scan_ssid=1
    key_mgmt=WPA-PSK
    psk="..."
}
```

Il ne vous reste alors plus qu'à lancer la demande de connexion

```
wpa_supplicant -iwlan0 -c/etc/wpa_supplicant.conf -B
```

Si le point d'accès auquel vous vous connectez ne vous fournit pas une adresse par DHCP, vous veillerez à en fixer une et à configurer les routes correctement.

5.2 Gestion de l'extension Wifi Bluetooth : Installation de paquetages

Pour l'activation du Bluetooth, les paquetages nécessaires ont bien été compilés mais n'ont pas été déployés sur la cible. Copiez et installez sur le NSLU le paquetage `km0d-bluetooth_2.6.26.5-ixp4xx-1_armeb.ipk`

Installez aussi les paquetages

```
bluez-libs, bluez-utils, bluez-hcidump, dbus, libdbus, libexpat, libiconv, libintl,
libpthread, librt.
```

Vous pourrez récupérer ses paquetages dans le fichier `openwrt/BTpackages.tgz` ou depuis l'adresse suivante :

<http://downloads.openwrt.org/backfire/10.03/ixp4xx/packages/>

Après installation de ces paquetages avec la commande `opkg`, il ne vous restera plus qu'à réaliser la configuration et à tester :

```
hciconfig hci0 up
```

TD n° 11

Système Embarqué : NSLU 2 et OpenWRT

```
hciconfig  
hcidtool inq  
l2ping 00:00:00:00:00:00 (adresse MAC du dispositif à pinguer)
```

5.3 Gestion des Phidgets : Création de votre propre paquetage

Dans le fichier `openwrt/phidgets.tgz`, vous aurez la structure de base d'un paquetage.

Nous avons aussi inclus les sources des bibliothèques et d'une application de test. Installez tout cela dans `backfire` vérifiez que cela vous a ajouté des nouvelles fonctionnalités à la configuration (`make menuconfig`) et activez ce nouveau paquetage (dans la rubrique Utilities). Après compilation, vous déploierez ce paquetage sur la cible et vous veillerez à le tester.

Vous consulterez les fichiers `Makefile` de l'archive fournie ce qui vous permettra pour une étape ultérieure d'avoir un exemple des fichiers à créer pour mettre en place vous-même un nouveau paquetage à déployer sur le NSLU2.

5.4 Autres possibilités d'extensions

Maintenant à vous de jouer avec vos propres paquetages et de créer vos propres extensions OpenWRT pour NSLU2.

6 Avant de partir...

Avant de rendre le précieux matériel que nous vous avons confié le temps de ce TD et de quitter la salle, vous veillerez à réinstaller le firmware d'origine sur le NSLU2.

Pour cela, récupérez le fichier suivant et installez-le grâce à l'utilitaire de mise à jour du firmware.

https://trolen.polytech.unice.fr/cours/sae/td11/NSLU2_V23R63.zip

En cas de problème, vous pouvez consulter la documentation à l'adresse suivante :

<http://www.nslu2-linux.org/wiki/HowTo/RevertToLinksysFirmware>