<h1>Plan d'une nouvelle tentative de fonctionnement d'un driver ADSL sur BCM963xx</h1> Avant de continuer à lire la suite de cet article, je vous conseil de lire les deux articles pr&#65533;&#65533;ents pour comprendre la mise en situation : Introduction à OpenWRT
Le probl&#65533;e du driver ADSL des BCM963xx Cet article sera mis à jour au fil du temps. Je crois que le flux RSS n'informe pas des mises à jour des articles.

Comme nous l'avons d&#65533;&#65533;vu, quelques tentatives ont eu lieu pour faire fonctionner le driver actuel sur OpenWRT. Pour l'instant, je ne compte pas trop sur l'aide d'un constructeur pour ce driver. La documentation technique n'est toujours pas disponible, il va donc falloir à nouveau contourner le probl&#65533;e. J'entrevois trois solutions a ce probl&#65533;e, un wrapper et l'&#65533;riture d'un nouveau driver bas&#65533;sur une partie du code source d'une puce plus ancienne et l'&#65533;riture d'un nouveau driver bas&#65533;sur du reverse engineering.
 Il y a de fortes chances que je n'arrive pas aux r&#65533;ultats escompt&#65533;. Concernant le wrapper : En premier, il va falloir modifier les binaires des drivers pour pouvoir intercaler le wrapper entre ces fichiers et le kernel. Une fois que nous auront la liste des fonctions non compatible avec OpenWRT, il faudra &#65533;rire le code qui permettra la compatibilit&#65533;

A ce niveau, la difficult&#65533;se situe au niveau des fichiers dev.c et skbuff.c.

Je devrais avoir fini cette partie en septembre et j'int&#65533;rerai le r&#65533;ultat dans cet article.

Quant à la deuxi&#65533;e solution, je vais partir sur la base des fichiers que j'ai trouv&#65533; pour une autre puce. Comme je n'ai pas regard&#65533;le contenu de ces fichiers, je ne peux pas planifier quoi que ce soit &#65533;ce niveau pour cette solution. Et il faudra surement utiliser partiellement la 3&#65533;e solution pour faire fonctionner celle-ci.

La 3&#65533;e solution est la plus longue, la plus difficile et la plus incertaine. Je serai surement d&#65533;ourag&#65533;avant d'y arriv&#65533; Je pense passer par un d&#65533;ompilateur - d&#65533;assembleur pour obtenir soit de l'assembleur (j'en ai trouv&#65533;deux mais je ne les ai pas encore essay&#65533;) soit un m&#65533;ange d'assembleur et de C ou C++ (n'ayant rien trouv&#65533;pour le faire, il me faudra cr&#65533;r l'outil ad&#65533;uat). Une fois ceci fait, il faut encore passer par une phase de compr&#65533;ension du code (on peut d&#65533;erminer quelles parties du code posent probl&#65533;e puisqu'on les a d&#65533;&#65533;identifi&#65533;s avec la 1&#65533;e solution). Puis faire l'inverse, surement en code assembleur puisque l&#65533;on n&#65533;aura pas de fichier totalement en C ou C++.

Ces solutions consistent à continuer les pistes que j'ai abandonn&#65533;s il y a quelques temps. Je le ferais quand j'en aurais le temps. &#65533; <h2><u>Mise à jour de la progression :</u></h2> Ayant r&#65533;up&#65533;&#65533;les sources disponibles sur le dernier SVN d'openWRT, j'ai pu obtenir un firmware. A not&#65533;qu'il ne m'est plus n&#65533;essaire de passer par les outils de constructions fourni par le fabriquant du modem, openWRT g&#65533;&#65533;e un firmware que l'on peut directement mettre à jour par le CFE sur la console.
&#65533;
Par contre, j'ai cru comprendre qu'il y a deux drivers diff&#65533;ents pour le chien de garde (watchdog). Celui qu'il y avait par d&#65533;aut ne fonctionne pas correctement sur mon syst&#65533;e. J'ai donc du remplacer le contenu du fichier dev-wdt.c par celui du driver du fabricant qui se contente de le d&#65533;activer.
&#65533;
Je n'ai que peu de m&#65533;oire disponible sur mon modem, je n'ai donc pas pu y inclure toutes les fonctionnalit&#65533; que je voulais mais l'essentiel est d&#65533;&#65533;la (Ethernet, wifi, interface web, ...).
&#65533;
Il ne reste plus que la partie modem et voip qui ne fonctionne pas. OpenWRT a introduit une mise à jour qui indique qu'ils sont en train de r&#65533;oudre ces probl&#65533;es. La partie voip ne m'int&#65533;esse pas pour l'instant, je me concentre sur la partie adsl. J'esp&#65533;e qu'ils y arriveront avant moi pour que je puisse m'occuper d'autres projets en attente.
 &#65533; J'ai regard&#65533;du cot&#65533;des d&#65533;ompilateur, c'est à dire

ceux qui fournissent du C, et j'en ai trouv&#65533;quelques uns. Le meilleur est un produit commecial. C'est IDA associ&#65533;&#65533;son plugin hex-rays. J'ai pu tester IDA 5.2 avec le plugin 1.0 et malheureusement, il n'est pas compatible mips. On obtient de l'assembleur. Une version 5.5/1.1 est sortie mais je n'ai pas plus d'informations si ce n'est que le code C s'approche encore plus des sources originales.
Il y a aussi &#65533;rec22 qui permet de d&#65533;ompiler mais on n'obtient pas grand chose d'exploitable.
 Je me suis pench&#65533;sur boomerang. J'ai r&#65533;ssit &#65533;le compiler ainsi que son interface graphique mais la partie mips est d&#65533;activ&#65533;. J'ai essay&#65533;de le r&#65533;ctiv&#65533;mais j'obtiens une erreur de lieur. D'apr&#65533; les exemples fourni sur le site officiel, il devrait &#65533;re assez bon sauf que la partie mips n'est pas encore au stade alpha.
 J'en ai trouv&#65533;d'autres mais je n'ai pas pu tous les essayer. Je compte me bas&#65533;sur quelques informations donn&#65533;par IDA associ&#65533;&#65533;boomerang s'il fonctionne et rec22 pour essayer d'obtenir un code source qu'il faudra retravailler notamment pour retrouver les type de retour des fonctions et les types des variables en entr&#65533;/sortie.