OpenWrt Developer Guide

Rev 1.4

January 2, 2012

Table of Contents

1 Revision History

Revision	Date	Summary of Changes
1.0	Jan 30, 2009	Initial revision.
1.1	Mar 17, 2009	Updated for final OpenWrt 8.09 release.
1.2	Aug 28, 2009	Updated for OpenWrt 8.09.1 release.
1.3	Sep 4, 2009	Removed Setting Up Disk Quota Source Code section as it is not necessary.
1.4	Jan 2, 2012	Updated for OpenWrt 10.03.1 release.

2 Introduction

This document describes how to build and use OpenWrt in Micrel platforms. OpenWrt is a Linux distribution for embedded devices. It provides a framework to build a firmware by including specific device drivers and applications. All source codes are downloaded from the Internet so an Internet connection is required for initial setup.

3 Preparing for Initial Setup

OpenWrt base source codes can only be retrieved from Subversion repositories, so Subversion is required. In the case that Internet connection is routed through a company proxy server, Subversion configuration needs to be modified to support proxy server. It is recommended to use the NTLM Authorization Proxy Server application in Linux to connect to a Windows proxy server.

If a proxy server is required to connect to the Internet, setup Subversion to use proxy server as necessary. In the user home directory there is a Subversion configuration directory called `.subversion` if Subversion has been run once beforehand. Edit the file called `servers`. Go to the **global** section at the end of the file. Assume NTLM Authorization Proxy Server is used, change http-proxy-host to “127.0.0.1” and http-proxy-port to “5865.”

3.1 Ubuntu Desktop Distribution Requirements

When using Ubuntu desktop distribution make sure the following packages are installed.

- autoconf
- bison
- flex
- gawk
- patch
- zlib-dev

3.2 Getting OpenWrt Code From the Internet

OpenWrt base source codes are retrieved from Subversion repository svn://svn.openwrt.org/openwrt. The tags subdirectory contains the snapshot of different releases. The latest release is 10.03.1. Use this command to get the code:

```
     svn co -q
svn://svn.openwrt.org/openwrt/tags/backfire_10.03.1
backfire_10.03.1
```

If hard disk space is a concern, use this command instead to skip the Subversion information:

```
     svn export -q
svn://svn.openwrt.org/openwrt/tags/backfire_10.03.1
backfire_10.03.1
```

The application packages are in the packages directory:

```
     svn co -q
svn://svn.openwrt.org/openwrt/branches/packages_10.03.1
packages_10.03.1
```

The web GUI package is from a different repository:

```
     svn co -q http://
x-wrt.googlecode.com/svn/branches/backfire_10.03 x-wrt_10.03
```

Another web GUI package is available:

```
     svn co -q http://svn.luci.subsignal.org/luci/tags/0.10.0
luci_0.10.0
```

As the OpenWrt code will be updated by developers, there may be compatibility issues with Micrel support code. So it may be necessary to get specific versions of OpenWrt code. For the records, the Micrel code work with Backfire 10.03.1 revision 29592, X-Wrt 10.03 revision 4953, and LuCI 0.10.0 revision 8131. The specific revision can be retrieved with the "`-r`" svn parameter.

OpenWrt has changed its Subversion repository URL from https to svn. Sometimes these repositories cannot be accessible because of company firewall. OpenWrt does provide the latest 10.03.1 source code in its website. Micrel also can provide a copy of the OpenWrt packages code. The Micrel OpenWrt 10.03.1 code work with Backfire 10.03.1 and its packages in revisions 29508-29638, X-Wrt 10.03 revisions 4953-4986, and LuCI 0.10.0 revisions 8086-8145.

OpenWrt creates a directory called `dl` under its working directory to store the downloaded files. These files will be removed when the working directory is cleaned completely. For efficiency it is better to create a soft link to point to a location where all the files are already downloaded. Just make sure to remove this link first before doing a distribution clean in the working directory.

3.3 Getting and Setting Up Micrel Support for OpenWrt

The Micrel updated code for OpenWrt is provided in the file archive openwrt-10.03.1.tar.bz2. Uncompress it using the command:

```
    tar jfx openwrt-10.03.1.tar.bz2
```

There is a script file called create-package in openwrt-10.03.1 directory. It is used to create soft link to Micrel code in OpenWrt directory and patch the code as necessary. Assume the OpenWrt codes are already setup in backfire_10.03.1, packages_10.03.1, luci_0.10.0, and x-wrt_10.03 directories. Run this command to setup the Micrel OpenWrt code:

```
    openwrt-10.03.1/link -d backfire_10.03.1 -p
packages_10.03.1 -x x-wrt_10.03 -w luci_0.10.0
```

Optionally the command "-l openwrt-dl" can be added to create a soft link to the openwrt-dl download directory.

Before the Micrel code is updated for a newer version, it is necessary to run this command to revert the OpenWrt code to original states:

```
    openwrt-10.03.1/unlink -d backfire_10.03.1
```

The create-package script assumes the OpenWrt codes are in original states. If they are not use the Subversion revert command to put back the original files.

4 Building OpenWrt

Go to the working OpenWrt directory, backfire_10.03.1. The configuration file is named .config. For a clean initial build make sure that file is not there, and remove it as necessary. There are only two steps to build the OpenWrt distribution. The first is configuration to select which drivers and applications to include in the distribution. The second is building the complete distribution to create image files.

4.1 Configuration

Use this command "make menuconfig" to bring up the OpenWrt Configuration utility. In **Target System** hit Return and select the proper system, which in this case can be Micrel platform. For Micrel platform the **Subtarget** can be changed to select KS8695X, KS8695V, KSZ8695P, KSZ9692P, or KSZ9692P_QUOTA. Select one of the offered profiles in **Target Profile**. One variation of the configuration can be:

Target System	Micrel
Subtarget	KSZ9692P
Target Profile	Micrel Pegasus KSZ9692P

The other options in the configuration are optional. Packages needed in the platform are already preselected. To change the GCC compiler version enable **Advanced configuration options** by typing `Y` and then go to the **Toolchain Options**. Select one of the supported compilers in **GCC compiler Version**.

The **Base system** contains application packages that are essential for running the system. The Linux kernel drivers are selected in **Kernel modules**. The web GUI packages are provided under **Administration**.

Exit the utility and save the configuration. It is recommended to copy the configuration file `.config` to some place else and rename it to something meaningful so that it can be used later.

To change kernel configuration after the toolchain has been built, use the command “`make kernel_menuconfig`.”

4.2 Build

If a proxy server is required to connect to the Internet, first setup the `http_proxy` environmental variable in case OpenWrt needs to download some files:

```
export http_proxy=”http://127.0.0.1:5865”
```

Build the complete OpenWrt distribution by typing “`make`.” For first build or debug purpose the “`V=99`” parameter can be added to provide information if something is wrong. The built toolchain is under the `staging_dir/toolchain-arm_v4t_gcc-4.3.3` directory. The Linux kernel is built under the `build_dir/linux-micrel_ksz9692p/linux-2.6.32.27` directory. The applications are built under the `build_dir/target-arm_v4t` directory. The target root filesystem is in `build_dir/target-arm_v4t/root-micrel` directory. The file images are under the `bin` directory. The file `openwrt-micrel-2.6-uImage` is the kernel image. The file `openwrt-micrel-squashfs.img` is the root filesystem image. The file `openwrt-micrel-2.6-squashfs.img` contains both kernel and root filesystem. This is likely the one to download. The file `openwrt-u-boot-micrel-2.6-squashfs.img` includes also the U-Boot bootloader. This image is used to burn a completely new flash chip. Copy the proper file image to the `tftpboot` directory to download to the target system.

4.3 Clean Up

Use the command “make clean” to do a basic cleanup. If compiler is changed in the configuration it is better to delete the old directories by typing “make dirclean.” For a complete clean use “make distclean.” Note the dl directory will be deleted and the configuration file .config removed.

5 Using OpenWrt

5.1 Downloading OpenWrt Image Using U-Boot

Under U-Boot download the OpenWrt image file into memory.

```
tftp 800000 openwrt.img
```

After the file is downloaded U-Boot will report the file size and store it in the filesize environmental variable. Assume it is 0x500000 bytes. Next copy that image to the flash memory. First erase the flash memory location. Assume the boot location is at 0x1c040000.

```
erase 1c040000 +500000
cp.b 800000 1c040000 500000
```

The Linux kernel will have its own boot command. So remove the bootargs environmental variable.

```
set bootargs
```

Boot the Linux kernel image.

```
bootd
```

5.2 Initial Boot

In the initial boot OpenWrt will setup the JFFS2 filesystem and so will take a little longer time to complete startup. It is preferred to reboot after that so that the JFFS2 filesystem can really be used.

Telnet can be used to connect to the device running OpenWrt. When the web GUI is used the first time it requires setting a password to the root account. After the password is set telnet can no longer be used. Instead secured shell utility is the only connection that OpenWrt accepts.

OpenWrt uses Squashfs for its root filesystem. Squashfs is read-only so JFFS2 filesystem is used to store updated files. OpenWrt also uses another filesystem called mini fanout overlay to put on top of Squashfs so that any change will automatically be updated in JFFS2.

The Squashfs root filesystem partition size is pre-configured in the Linux kernel. When the Squashfs image is created a marker is put at the end of the filesystem image to indicate empty JFFS2 filesystem. When the kernel starts it determines the actual size of the Squashfs partition and then uses the rest of original partition to create a new partition for the JFFS2 fileysystem. It then erases that partition to create the empty JFFS2 filesystem.

OpenWrt uses a simple way to start applications that run in background. All applications that want to start at startup have a script file created specifically for OpenWrt. These script files are copied to the `/etc/init.d` directory. Inside each of the script files there is a priority number. OpenWrt uses those numbers to create startup and terminate files stored in `/etc/rc.d` directory. OpenWrt starts those applications according to the order assigned.

Application configurations are stored in `/etc/config` directory. OpenWrt provides a UCI (User Configuration Interface) utility to manage those configurations. For how to use this utility type “`uci help`.”

Some often used commands:

```
uci show <app>
```

where `<app>` is the application which configurations you want to see.

```
uci get <app>.<parameter>
```

where `<parameter>` is a parameter defined in the configuration.

```
uci set <app>.<parameter>=<value>
```

to change the value of a configuration parameter.

```
uci commit <app>
```

to commit the changes and save the configuration in JFFS2 filesystem.

6 Using Applications

Most of the applications in OpenWrt are command line utilities. To access these applications it requires login to the root system. Secured shell utility is used for this purpose. Under Linux use `ssh`. Under Windows use `putty`.

6.1 X-Wrt

X-Wrt provides a web GUI to OpenWrt so that its configuration can be managed using the web browser. It enhances the user experience of OpenWrt. Its package name is webif.

X-Wrt requires the root account to have a password so that it can be log into. Upon the first run a password needs to be set for the root account. Type in a password twice and hit the **Set** button. The browser will ask for the user to login. Enter `root` and the new password.

X-Wrt provides the capability to backup and restore system configurations. It can also upgrade system firmware.

6.2 LuCI

LuCI (Lua Configuration Interface) is the new web GUI developed for use in OpenWrt.

6.3 Miniupnp

Miniupnp is a simple UPnP server so that clicking the UPnP device that shows up in Windows will bring up the web browser to present the web configuration management console. Miniupnp can be managed under X-Wrt. There can be only one UPnP server.

6.4 MediaTomb

MediaTomb is a UPnP Media Server. It exports media files such as photos, MP3 music files, and MPEG videos so that a compatible UPnP Client can display and play those files.

MediaTomb uses a simple database to keep track of metadata of all the media files added. This database is created on the external hard drive mounted at `/media/disk`. MediaTomb uses a web GUI to manage this database. Under Windows clicking on the MediaTomb UPnP device will bring up the web browser displaying MediaTomb GUI. If not point the browser to location http://192.168.1.1:49152 where port 49152 is the starting port address used by MediaTomb.

The left side of GUI has two selections: **Database** and **Filesystem**. Select **Filesystem** and the whole directory tree is displayed. Navigate to the media->disk to see the contents of the mounted hard drive. On the right side are files in the hard drive listed in their names. Select which file to add and click the + sign, or add all the files under the directory by clicking the + sign at the top. Select **Database** again and the added media files will be grouped accordingly into audio, photos, and video categories.

MediaTomb is known to work with PS3, Xbox 360, and other Linux-based UPnP clients. Under Windows PC the only known UPnP client is XBMC.

6.5 Quota

The disk quota support allows system administrators to put a limit of how much space users can use. It requires kernel support so that is why a specific SubTarget KSZ9692P_QUOTA is used in the OpenWrt configurations. The quota utilities—`quotacheck`, `quotaon`, `repquota`, `edquota`, and `setquota`—are all command line, so it requires a terminal to manipulate the quota settings.

6.6 E2fsprogs

This package provides utilities to format and maintain Linux ext2 and ext3 filesystems.

6.7 Samba

Samba is a file server. It allows Linux files to be accessible under Windows using SMB protocol. There are two Samba packages. The original `samba3` package builds only the smbd daemon program which is small. The disadvantage is that under Windows the shared drives can only be connected using the "`net use`" command. Micrel provides another `samba3-nmbd` package that also builds the nmbd daemon program. The resulting binary images are quite large so they may not fit in the embedded devices.

7 Compatibility

OpenWrt and its associated Web GUI applications, LuCI and X-Wrt, are in ongoing development, so getting their latest code and then applying Micrel patches may result in some conflicts. If the developers know how to correct those conflicts, then that is fine. To avoid this issue, it is better to get the exact revisions.

Micrel supports Kamikaze release 8.09 and 8.09.1, and Backfire release 10.03.1. The name of the Micrel patches for 8.09 is `openwrt-8.09.tar.gz`. It works with Kamikaze 8.09 and its packages in revision 14827, X-Wrt 8.09 revisions 4574-4721, and LuCI 0.8.6 revision 4316.

The Micrel patches for 8.09.1 is `openwrt-8.09.1.tar.gz`. It works with Kamikaze 8.09.1 revisions 16278-17513, 8.09 packages revision 17513, X-Wrt 8.09.1 revisions 4773-4819, and LuCI 0.8.7 revision 5064-5317.

The Micrel patches for 10.03.1 is `openwrt-10.03.1.tar.gz`. It works with Backfire 10.03.1 revisions 29508-29638, 10.03.1 packages revisions 29508-29638, X-Wrt 10.03 revisions 4953-4986, and LuCI 0.10.0 revisions 8086-8145.

Under the `openwrt-10.03.1` directory is a `VERSION` file that specifies which Backfire, X-Wrt, and LuCI revisions are supported.

8 Development

LuCI web management is preferred to use in OpenWrt. After a LuCI developer guide is published, new Micrel web pages can be implemented.

Alphabetical Index