Getting started with OpenWode

Date: 2011-05-20

OpenWode version: openwode-v1.0b1

Author: The Doctor/ The Mentor

1 What is OpenWode?

OpenWode is a firmware package that allows Wode users to build and run their own home-cooked Embedded Linux OS on Wode.

OpenWode consists of the following items:

1. an update.bin for installing the OpenWode firmware. While this is installed, you cannot run any normal Wode software.
2. An update.bin to restore the firmware back to the Wode v2.8a. After installation of this update, your Wode will function again like a normal wode and can be used to play your favorites ISO files.
3. The source code for building a minimal embedded Linux runtime.

OpenWode is entirely based on OpenWrt so when you would like to know more about the build environment, you can always go to http://www.openwrt.org and browse for more information. So in many cases where you read OpenWode, you could also read OpenWrt. We have chosen to stick to one name for clarity, but it is absolutely not our intention to pretend that we wrote the OpenWrt build environment and runtime. These are the people who wrote OpenWrt: https://dev.openwrt.org/wiki/people

OpenWode contains the following changes w.r.t. OpenWrt:

1. OpenWrt does not support the LPC313x series. Therefore support has been added.
2. Some configuration has been done for the Wode hardware.
3. The `build.sh` script was added to minimize the interaction with the command line for dummies.

When you unpack the ZIP file, there are a number of directories. We will briefly discuss them in the next subsections.

updates

In the `updates` directory are two subdirectories:

- `openwode` : this directory contains the `update.bin` to install the OpenWode firmware
- `retail` : this directory contains the `update.bin` to revert back to the regular Wode firmware (rev. 2.8a)

images
The `images` directory contains the `uImage` file. It contains the prebuilt OpenWode embedded Linux image and was build from the sources included in the sources directory.

Sources
The `sources` directory contains the file `openwode-v1.0b1-19052011.bundle`. You can use it to build your own OpenWode image and make modifications or add your own applications.
How you can use the `openwode-v1.0b1-19052011.bundle` file is described in subsequent sections.

This document will describe the quick steps to take to build an image and deploy it on Wode hardware.

2 prepare the build host

It is assumed here that the build host is running a Debian based Linux distribution (like Ubuntu). It is not tested on other Linux distributions, but since we have used standard functionality of OpenWrt, chances are that your platform is supported as well... Please check this page to read more about supported platforms:
http://wiki.openwrt.org/doc/howto/buildroot.exigence

A build will NOT work on (any) M$ Windows natively for sure.

If you don't have such a machine, please download VirtualBox from http://www.virtualbox.org/wiki/Downloads and install it.

Then you can create a virtual machine and boot it with an Ubuntu ISO install image. These can be obtained from: http://www.ubuntu.com/download/ubuntu/download.

We suggest you take the 10.04 LTS version. It is not important if you are using your machine in 32 or 64 bits mode.

In order to be able to execute the next steps, you will need to install some packages that are needed by programs we will run later. You can install these packages by entering the following command on a command line on the build host (make sure there is nothing after the '\' on the fore-last line!):

```
sudo apt-get install build-essential subversion \
libncurses5-dev zlib1g-dev gawk flex git-core
```

or in case you have a 64-bit machine install the following packages:

```
sudo apt-get install build-essential subversion gettext \
libncurses5-dev zlib1g-dev gawk flex gcc-multilib git-core
```

If you are not running a Debian based distribution, then please check http://wiki.openwrt.org/doc/howto/buildroot to see how to install the prerequisites for your

distribution. Please note that the binaries for the GIT source control system are added to the list of packages found on this page.

If you are not familiar with the concepts of git yet, please read this great on-line book first: http://progit.org/book/

It is very likely worth the time and will pay off when trying to understand the seemingly git magic in the next sections.

3 Create an OpenWode working directory

The OpenWode release consists of three parts:

1. the OpenWode sources, based on OpenWrt
2. openwode-update.bin. This wode update file contains new firmware that will allow you to load your custom software on the Wode.
3. documentation (this document)

Item 1 consists of so-called *git bundle*. A git bundle is single file representation of a complete repository of the git source control system. Please have these files ready on your system since we will need them in the next steps!

4 Create the OpenWode working directory

```
export OPENWODE_ROOT=${HOME}/openwode
mkdir -p ${OPENWODE_ROOT}
```

Then descent into the fresh new working directory and create an empty git working directory:

```
cd ${OPENWODE_ROOT}
git init
```

The output from the git init command will tell you that git Initialized an empty Git repository in `/home/<username>/openwode/.git/`

Then we need to get the stuff from the release into our working directory:

```
git pull <path to the openwode git bundle>
git fetch --tags <path to the openwode git bundle>
```

This will copy all changes contained in the git bundle file into our working directory.

The last thing we need to do is make sure we have the correct version. Our local git working directory contains the full history and we must ensure that we select the right version. Like any source control system, git uses *tags* to track versions. We will now checkout the correct version:

```
git checkout openwode-v1.0b1
```

Please note that chances are that you have selected a revision that is not the latest version in the repository. Since you cannot alter history, this means that you should not

modify the sources. If you want to experiment and change some of the sources, you should create a *branch* with the following command:

```
git checkout -b my_openwode_branch
```

5 Setup the OpenWode working directory

Now we need to prepare OpenWode for building.

All we need to do is copy the configuration file into the right place:

```
cd ${OPENWODE_ROOT}
./build.sh prepare
```

Because OpenWode is using

6 Build the OpenWode image

To start building the image, please issue the following command:

```
./build.sh build
```

Since it can be handy to inspect the output from the build process afterwards, the build.sh scripts also stores the output of the build process in a file called `compile.log`. Before each build this file is emptied so if you want to keep the output but want to start another build, you will have to save a copy of it.

After the build process has finished successfully, you can find the produced binaries in the `${OPENWODE_ROOT}/bin/lpc313x` directory.

It will contain a binary called `openwrt-lpc313x-2.6-uImage`. This file contains our software and we will use it in the following steps.

7 Prepare your Wode device for use with OpenWode

To be able to load your software on the Wode, you need to install a special update called `update-openwode-vf01.bin`.

PLEASE NOTE THAT THIS UPDATE WILL REMOVE ALL SOFTWARE FROM THE WODE. YOU WILL NOT BE ABLE TO USE YOUR WODE AS USUAL AS LONG AS THE OPENWODE FIRMWARE IS INSTALLED!

Put the file `update-openwode-vf01.bin` in the root of an SD card (vfat formatted) and rename it to `update.bin`.

Now insert the SD card into the slot on the Wode and power up the wode.
The display on the Wode will indicate that the firmware is being updated.

After the upgrade is done, your Wode is ready to run custom firmware.

Now put the `openwrt-lpc313x-2.6-uImage` you build in the previous steps and put it also in the root of your SD card and rename it to `uimage`.

But before we proceed with trying to start our home-grown OpenWode image, we need to setup the network first. Because the console is not accessible, we need to log in on the Wode using an SSH client. SSH is an acronym for Secure Shell.
To be able to log in with SSH, we need to have a network connection and know the IP address of the Wode. Therefore we will first setup the network in the next section.

8 Network configuration on the Wode

In this section you will read how you can configure the network capabilities of your wode. The following subsections will learn you the details for each specific use-case.

1. Wireless configuration with DHCP
2. Wireless configuration with static IP address
3. Wired configuration with DHCP
4. Wired configuration with static IP address

WIRELESS CONFIGURATION with DHCP

Configuration of the wireless network is done by editing a file called 'wifi' and must be placed in the root directory of the SD Card.

It contains parameters and their values as name-value pairs, for example "param1=value1".

2 or 3 name-value pairs must be present in your 'wifi' config file:

ssid
: The broadcasted SSID of the wireless network

encryption
: Wireless encryption method. `none` for an open network (you don't need to specify *key* in this case), `wep` for WEP, `psk` for WPA-PSK, or `psk2` for WPA2-PSK. See the WPA modes table for additional possible values.

key
: In any WPA-PSK mode, this is a string that specifies the pre-shared passphrase from which the pre-shared key will be derived. If a 64-character hexadecimal string is supplied, it will be used directly as the pre-shared key instead.
In WEP mode, it contains a string specifying a pass-phrase or key directly.

Example for WPA-PSK config:

```
ssid=Backfire
encryption=psk
key=WoDan???
```

Example for WPA2 Personal (PSK):

```
ssid=Backfire
encryption=psk2
key=WoDan???
```

In the table at the bottom of this page are all possible values for the *encryption* parameter. Please look there for more details.

WIRELESS CONFIGURATION with static IP address

It is strongly suggested to use DHCP and add a static DHCP address to your wireless router for your wode. This way you don't need to configure your Wode but it will still always get a predefined IP address. There might be situations though when you would want to config your wode with a static IP address.

[Configuration of the wireless network is done by editing a file called 'wifi' and must be placed in the root directory of the SD Card.

It contains parameters and their values as name-value pairs, for example "param1=value1".

To configure wireless with a static IP address, only a few parameters must be added to the 'wifi' file:

proto
: For a static ip address, the *proto* parameter is always `static`

ipaddr
: As you might have guessed already, this parameter contains the static ip address that must be used. A value for example could be "192.168.0.123"

netmask
: This contains the netmask for the network that you want to connect to.
It depends on your IP address range, but this is usually set to '255.255.255.0' for home applications.

gateway
: This tells the wode where to send network packets that are destined outside the current network. it is usually the address of your wireless router and in case of our previous example, it will have a value of '192.168.0.1'

dns
: This tells the wode where to ask for host name lookups. It is usually the address of your wireless router. Example: '192.168.0.1'

Optional are:

hostname
: You can set the hostname of your wode if you like. The host name cannot contain spaces.

Example for WPA-PSK config with static IP:

```
ssid=Backfire
encryption=psk
key=WoDan???

proto=static
ipaddr=192.168.0.123
netmask=255.255.255.0
gateway=192.168.0.1
dns=192.168.0.1
```

WIRED CONFIGURATION with DHCP

For wired network configuration with DHCP, all the defaults will suffice and you don't need to have a special config file on your SD Card.

If you wish you can still have a file called 'wired' in the root of your SD Card.
It contains parameters and their values as name-value pairs, for example "param1=value1".

It can contain the following parameters:

proto
: For use with DHCP, the proto parameter is always `dhcp`

hostname
: You can set the hostname of your wode if you like. The host name cannot contain spaces. The default host name is 'wode' so you will recognize your device just using the default value.

WIRED CONFIGURATION with static IP address

Configuration of the wired network is done by editing a file called 'wired' and must be placed in the root directory of the SD Card. It contains parameters and their values as name-value pairs, for example "param1=value1".

It is best to configure your wired dongle to use DHCP for configuration. You can specify a fixed IP address at your router so that your Wode is assigned a previously known IP

address. There might be situations though when you would want to config your wode with a static IP address.

The following name-value pairs must be present in your 'wired' config file:

proto
: For a static ip address, the *proto* parameter is always `static`

ipaddr
: As you might have guessed already, this parameter contains the static ip address that must be used. A value for example could be "192.168.0.123"

netmask
: This contains the netmask for the network that you want to connect to.
It depends on your IP address range, but this is usually set to '255.255.255.0' for home applications.

gateway
: This tells the wode where to send network packets that are destined outside the current network. it is usually the address of your wireless router and in case of our previous example, it will have a value of '192.168.0.1'

dns
: This tells the wode where to ask for host name lookups. It is usually the address of your wireless router. Example: '192.168.0.1'

Optional are:
hostname
: You can set the hostname of your wode if you like. The host name cannot contain spaces.

Example for wired config with static IP:

```
proto=static
ipaddr=192.168.0.123
netmask=255.255.255.0
gateway=192.168.0.1
dns=192.168.0.1
```

WPA modes

The table at this link shows all possible values of the [i]encryption[/i] parameter. Please check the 'maintenance' web page of your wireless access point to see what encryption method you must use.

http://wiki.openwrt.org/doc/uci/wireless#wpa.modes

9 Installing an SSH client

Windows

We advice you use the puTTY SSH client. It is freeware and can be downloaded from here: http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe

Save it to for example your Desktop or create a folder. You can start it up straigt away.

Debian/ Ubuntu Linux

Just run the following command line in a terminal window:

```
sudo apt-get install openssh-client
```

10 Accessing the OpenWode

Now you can log in to your OpenWode installation using SSH. First you need to find out what IP address has been assigned to your Wode. How this was done depends totally on the choices you made in section “Network configuration on the Wode“, starting on page 5.

The username and password are

Username	root
Password	Openwode

Via a terminal window, you can issue the following command:

```
ssh root@<IP ADDRESS>
```

Below you can see what the result could look like. Only the first time you will be asked for confirmation. Since it is the first time you connect to the Wode on this IP address, SSH cannot authenticate the Wode. Therefore you are asked if this machine can be trusted. It is absolute safe here to answer 'yes'.

Then you will be asked to enter your password (like every time you try to log in on your Wode). Enter the password en press <ENTER>.

```
TheDoctor@CountyGeneral:~$ ssh root@192.168.0.123
The authenticity of host '192.168.0.123 (192.168.0.123)' can't be
established.
RSA key fingerprint is
e1:aa:19:e3:20:9b:43:1f:24:3f:ef:71:83:c3:85:c9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.123' (RSA) to the list of
known hosts.
root@192.168.0.123's password:


BusyBox v1.15.3 (2011-05-19 00:25:34 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.


  _________                 _
 |   |   |   | _____   ___| | _____          Wii
 |   |   |   ||  _  ||  _  ||  -__|   Optical Drive
 |_____ ___||_____||_____||_____|     Emulator
  Backfire Latest r26317 - Kernel 2.6.32
  ------------------------------------------------------
root@Wode:~#
```

At this point you have build, booted and logged in on your wode successfully.

It is now up to you to do with it whatever you like!

----- END OF DOCUMENT -------