WHITE PAPER

OpenWRT on the Belkin F5D7230-4

Compiling and Installing the GPL Broadcom Reference Firmware

CONTROL PAGE

Document Approvals

Approved for Publication:

Author Name:	Ian Latter 22 November 2004

Document Control

Document Name:	OpenWRT on the Belkin F5D7230-4; Compiling and Installing the GPL Broadcom Reference Firmware
Document ID:	openwrt on the belkin f5d7230-4 - broadcom firmware.doc-Release-0.1(518)
Distribution:	Unrestricted Distribution
Status:	Release
Disk File:	C:\papers\OpenWRT on the Belkin F5D7230-4 - Broadcom Firmware.doc
Copyright:	Copyright 2004, Ian Latter

Version	Date	Release Information	Author/s
0.1	*22-Nov-04*	Release / Unrestricted Distribution	Ian Latter

Distribution

Version	Release to
0.1	MidnightCode.org

Table of Contents

1 Overview

1.1 In brief

In this third paper the Belkin F5D7230-4 is further explored for its availability as a fully integrated wireless firewall router and VPN end-point, widening the search, however, to include the Broadcom reference GPL firmware.

This work must be done in order to achieve a best-practice security solution in the Small Office / Home Office (SOHO) price-point. Where, while almost all of the casual risks are equivalent to those experienced by large enterprise, no mitigation technologies are available at an appropriate cost.

The work of the first paper has been widely accepted, but successful deployment of OpenWRT did not ensue. Hidden or unknown errors in the boot process lead the author to seek a means for attaining console access so that productive debugging (rather than guesswork) could be employed in the development process. The second paper was successful in meeting its objective; a solder-less project that enabled the construction of a fully functional 115,200bps serial console for use with a Personal Computer, to debug errors in the boot sequence of custom (open source) firmware.

Belkin published the Broadcom reference firmware – a small Linux distribution, designed to act as a proof-of-concept and development environment for the Belkin engineers. To minimize the amount of experimentation required to adapt the OpenWRT and Sveasoft firmware for use on the Belkin, the published Broadcom reference firmware was compiled to see if it was functional, and able to provide driver and configuration information for the open source distributions.

By collating a mass of publicly available information, and experimenting with a single unit, the paper concludes by providing a detailed guide that will allow individuals to download and compile the Broadcom reference firmware. It is hoped that this information can be used to hasten the configuration of the OpenWRT, and Sveasoft, embedded Linux distributions for this Belkin router.

1.2 History

This paper represents a fifth week's work and a sixth week's documentation. It continues where the previous two documents (*OpenWRT on the Belkin F5D7230-4 – Understanding the Belkin extended firmware for OpenWRT development* and *OpenWRT on the Belkin F5D7230-4 – Compiling and Installing the GPL Broadcom Reference Firmware*) left off.

Sveasoft have provided access to their forums and software, to help facilitate the discovery process on this device. Access to Sveasoft and other public resources has helped to expose existing knowledge on similar devices, making it easier to evaluate this particular device.

Thank you to every contributing forum member, distant email contact, other problem solvers who've published their content, and friends alike. There wouldn't have been success without you.

2 The problem – and throwing software at it

2.1 Since the last paper

After resolving firmware data structure issues and creating a root shell on the F5D7230-4 in the first paper, then adding a serial console for interactive boot observation in the second paper, a pregnant pause was felt within the project.

With access to the complete software resource that is the Internet, including openwrt.org and sveasoft.com, it soon became clear that there was no obvious place to get a firm foot-hold on the task of modifying an existing open source project to make only the necessary modifications to “port” it to another platform (even if that platform was only marginally different to an existing embedded platform).

This issue is largely due to the limited documentation that exists in these open source communities. The approach that “the true programmer will find his way” is not without its virtue, but tends to add an unnecessary learning curve to an otherwise short-lived project.

2.2 Why play with even more software?

Early in the second paper, it was discovered that Belkin had published their open source firmware components – as a fairly complete source environment. This was not particularly appealing, as there was already an abundance of open source Broadcom-based Linux software (including kernel drivers) already available in the OpenWRT and Sveasoft projects.

It is not until one comes to the issue above (i.e. - finding no good place to start) that it becomes clear that the Broadcom reference firmware might be useful.

Thus, if it were possible to find a working kernel configuration and functional kernel drivers for the Belkin F5D7230-4 router, then this working software could be readily compared to the open source projects, and the changes ported as required.

3 Compiling the GPL Broadcom Reference Firmware

3.1 Before we begin

The following process aims to facilitate further discovery; to provide a platform for understanding the uniqueness of the Belkin firmware, versus the Linksys WRT54G.

To this end the following process does not aim to create the desired additional functionality identified in the first paper, instead, it aims to provide a Belkin compatible firmware for further firmware debugging and exploration.

Note that this documentation has been written for the F5D7230-4 and in a factory-default configuration.

Following these instructions will void your Belkin warrantee, and may render your router unusable.

No warrantee is implied or intended when you choose to follow these instructions.

Proceed at your own risk

3.2 Development server prerequisites

The following process was performed on a Virtual Machine installed with Fedora Core 2, running under Windows XP. This is a little extravagant and unnecessary.

A more reasonable development server would be a workstation configured with;

- A recent Linux distribution (A current version of RedHat, Debian, etc)
- Development utilities (A “Workstation” build under RedHat’s Fedora is ideal)
- 2Gbytes of free disk space on a single partition (/home is used in this process)

The process for building the GPL Broadcom reference firmware follows fourteen easy steps, takes about two hours to complete, and should work on any modern / current Linux installation.

The entire process is based on the assumption that the reader has a single root shell open (a console if need be), and is following the documentation as a live, step-by-step guide.

Furthermore, this process was developed rapidly due to the excellent work performed by Rick Bronson. Rick published the findings of his work on his web site and has been very supportive of the development process (see References for Rick’s URL).

3.3 Download the required software

The first two steps focus on bringing down a copy of the software source code and tools required to build to the reference firmware.

3.3.1 Step 1: Download the Broadcom reference firmware

Download the Broadcom reference firmware bundle GPL-4-00-03.tgz. This 46.5Mbyte file can be retrieved from the Belkin GPL Source Site;

```
http://web.belkin.com/support/gpl.asp
```

The Belkin GPL Source Site

Store the file GPL-4-00-03.tgz in the /tmp directory.

3.3.2 Step 2: Download the Broadcom cross-compile tool chain

Download the Broadcom (MIPSEL) cross-compile tool chain bundle wrt54g.2.02.2.tgz. This 147.5Mbyte file can be retrieved from the Linksys GPL Source Site;

```
http://www.linksys.com/support/gpl.asp
```

The Linksys GPL Source Site

Store the file wrt54g.2.02.2.tgz in the /tmp directory.

3.4 Prepare the workspace

With the software downloaded, a workspace must be prepared to make compilation straight-forward and successful.

3.4.1 Step 3: Create a working directory

The working directory is the place holder for the source code and tool chain. The entire Broadcom firmware compilation is to be performed in this directory. To fit the source code, tool chain and compiled firmware, the total space required on the file system of the working directory is 2Gbytes.

To create the working directory and move into it, perform the following as the root user, from the shell prompt;

```
cd /
mkdir –p /home/belkin
cd /home/belkin
```

3.4.2 Step 4: Configure the user environment

Shell environment variables are accessed and referenced by a number of scripts and binaries within the build process.

To configure the user environment appropriately, enter the following at the shell prompt;

```
PATH=$PATH:/home/belkin/GPL-4/tools
PATH=$PATH:/opt/brcm/hndtools-mipsel-linux/bin
PATH=$PATH:/opt/brcm/hndtools-mipsel-uclibc/bin
OPT_HOME=/opt/brcm
CUSTOMER=belkin
CROSS_COMPILE=$OPT_HOME/hndtools-mipsel-uclibc/bin/mipsel-uclibc-
export PATH OPT_HOME CUSTOMER CROSS_COMPILE
```

3.5 Create the cross-compile tool chain environment

With the software downloaded and the workspace prepared, the tool chain can be unpacked and linked appropriately, to prepare for the firmware compilation.

3.5.1 Step 5: Unpack the cross-compile tool chain bundle

The Linksys cross-compile tool chain bundle expands to 234Mbytes of files.

To extract the files in the bundle, perform the following at the shell prompt;

```
cd /home/belkin
tar xvfz /tmp/wrt54g.2.02.2.tgz
```

3.5.2 Step 6: Symlink the tools into the anticipated location

The Broadcom build scripts expect the Linksys tools to be located in the /opt/brcm directory. There is no need to copy the tools into this directory – instead, the tools will be symbolically linked to this location.

To symlink the tools, perform the following at the shell prompt;

```
cd /opt
ln –s /home/belkin/WRT54G/tools/bcrm .
cd /home/belkin
```

3.6 Create the development environment

With the software downloaded, the workspace prepared and the tool chain unpacked and linked appropriately, the Broadcom reference firmware can be unpacked and prepared for compilation.

3.6.1 Step 7: Unpack the reference firmware bundle

The Broadcom reference firmware source bundle expands to 825Mbytes of files.

To extract the files in the bundle, perform the following at the shell prompt;

```
cd /home/belkin
tar xvfz /tmp/GPL-4-00-03.tgz
```

3.6.2 Step 8: Repair Jerry Lee’s crippled software

Due to some missing #define’s, Jerry Lee’s code, from the Broadcom reference firmware source bundle, fails to compile. Fortunately, the missing pieces were published at SeattleWireless (see References).

To repair the crippled software, perform the following at the shell prompt;

```
vi GPL-4/src/router/ppp/pppoecd/sys-linux.c
```

Go to line 44, and insert the following two lines;

```
#define PPPIOCGLANIP _IOR('t', 92, int)
#define PPPIOCSLANIP _IOW('t', 91, int)
```

3.6.3 Step 9: Repair Gerald’s uncommented re-define

Due to an additional #define, Gerald’s code, from the Belkin reference firmware source bundle, fails to compile. By adding a condition to the declaration, it is possible to allow the compiler to choose whether or not to include Gerald’s modification.

To make this modification, perform the following at the shell prompt;

```
vi GPL-4/src/include/bcmnvram.h
```

Go to line 21, and make it look like the following;

```
#ifndef INLINE
#define INLINE inline            //Gerald20030113, added
#endif
```

3.7 Configure and build the firmware

With everything in place, it’s time to compile the Broadcom reference firmware.

3.7.1 Step 10: Pre-configure the Linux kernel with the Belkin defaults

Broadcom supply a default Linux kernel configuration that is appropriate for the Belkin hardware.

To copy this configuration file into place, perform the following at the shell prompt;

```
cd GPL-4/src/linux/linux
cp default_belkin .config
cd /home/belkin
```

3.7.2 Step 11: Configure the firmware

The Broadcom firmware environment is self-supporting. The developer is able to customise the entire firmware environment in the same was as he could customise the Linux kernel – using a native menu system. It is recommended that the defaults are accepted.

Launch the menu system by performing the following at the shell prompt;

```
cd GPL-4/src/router
make menuconfig
cd /home/belkin
```

Save the configuration to the default location.

3.7.3 Step 12: Prepare the firmware

The Broadcom scripts allow a “make clean” to build dependencies and remove built binaries (without removing earlier-built kernels).

To prepare the firmware, perform the following at the shell prompt;

```
cd GPL-4/src/router
make clean
cd /home/belkin
```

3.7.4 Step 13: Compile the firmware

Compiling the firmware is ultimately a simple step! However, this step could take up to two hours to complete.

To compile the firmware, perform the following at the shell prompt;

```
cd GPL-4/src
make
cd /home/belkin
```

3.7.5 Step 14: Construct the firmware image

With the firmware compiled, the final step is to assemble a single firmware image file (a TRX image file – not an extended Belkin firmware image file).

To construct the firmware image file, perform the following at the shell prompt;

```
cd GPL-4/src/router
make install
cd /home/belkin
```

3.8 Retrieving the completed firmware image and flashing it

The completed Broadcom reference firmware is in the mipsel-uclibc directory, and is called “linux.trx”. This file can be uploaded to the Belkin directly.

To retrieve the completed firmware, perform the following at the shell prompt;

```
cd GPL-4/src/router/mipsel-uclibc
```

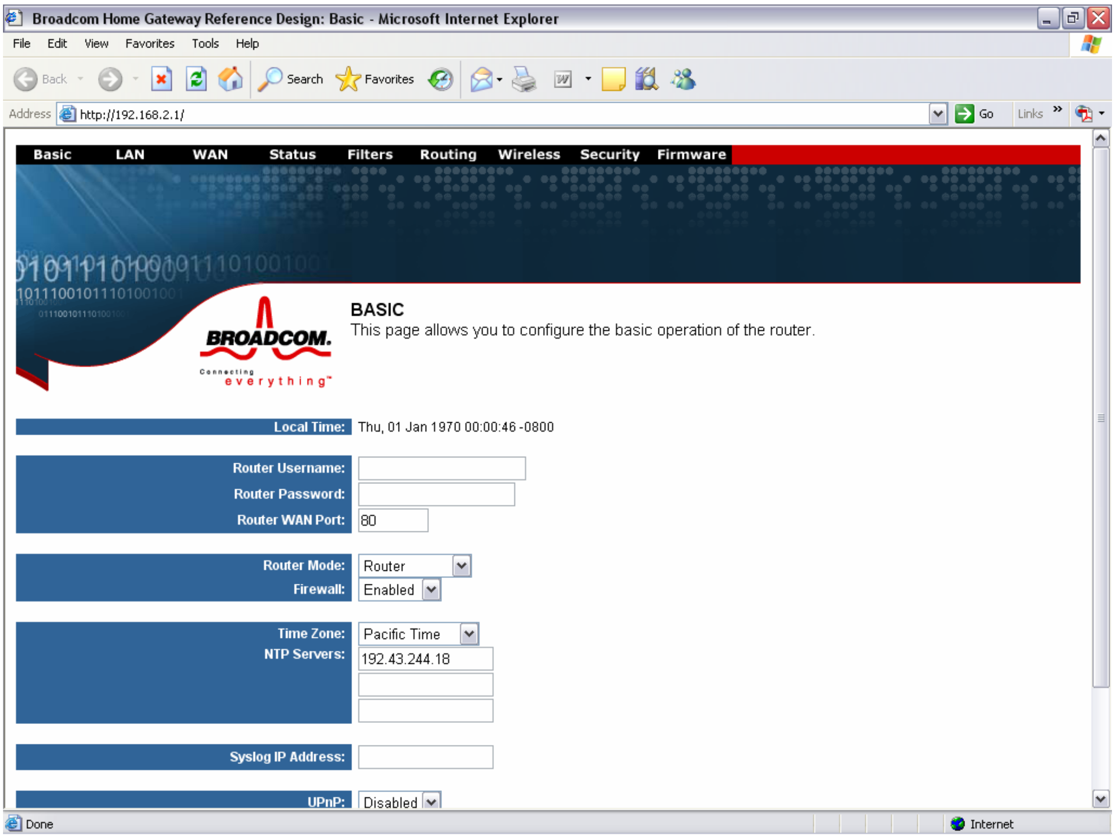
To flash the firmware image to the Belkin, see the first document in this series; *OpenWRT on the Belkin F5D7230-4 – Understanding the Belkin extended firmware for OpenWRT development*, section 3.2 Recovery – reliably flashing firmware (where *firmware.bin* is now *linux.trx*)

4 Findings

4.1 Broadcom GUI

The Broadcom reference firmware is a full (and functionally-equivalent) version of the Belkin firmware. The Broadcom reference firmware is so feature rich that it makes the stock Belkin firmware appear as though it is a re-badged / re-branded Broadcom.

Seen here, the Broadcom reference GUI (web interface), in all its glory;



BASIC

This page allows you to configure the basic operation of the router.

Local Time:	Thu, 01 Jan 1970 00:00:46 -0800
Router Username:	
Router Password:	
Router WAN Port:	80
Router Mode:	Router
Firewall:	Enabled
Time Zone:	Pacific Time
NTP Servers:	192.43.244.18
Syslog IP Address:	
UPnP:	Disabled

4.2 Broadcom console output

The console output from the Broadcom boot sequence has been posted here (note the incorrect time and date settings within the development virtual machine);

```
Here we try to capture the default reset button: None.

CFE version 1.0.37 for BCM947XX (32bit,SP,LE)
Build Date: Mon Apr 19 18:19:30 CST 2004 (denny@dnylinux)
Copyright (C) 2000,2001,2002,2003 Broadcom Corporation.

Initializing Arena.
Initializing Devices.
et0: Broadcom BCM47xx 10/100 Mbps Ethernet Controller 3.60.9.0
CPU type 0x29007: 200MHz
Total memory: 0x800000 bytes (8MB)

Total memory used by CFE:  0x80300000 - 0x80434A50 (1264208)
Initialized Data:          0x8032EB60 - 0x80330E90 (9008)
BSS Area:                  0x80330E90 - 0x80332A50 (7104)
Local Heap:                0x80332A50 - 0x80432A50 (1048576)
Stack Area:                0x80432A50 - 0x80434A50 (8192)
Text (code) segment:       0x80300000 - 0x8032EB60 (191328)
Boot area (physical):      0x00435000 - 0x00475000
Relocation Factor:         I:00000000 - D:00000000

Device eth0:  hwaddr 00-11-50-0D-DD-C4, ipaddr 192.168.2.1, mask
255.255.255.0
        gateway not set, nameserver not set
Reading :: Failed.: Timeout occured
Loader:raw Filesys:raw Dev:flash0.os File: Options:(null)
Loading: ..... 1482752 bytes read
Entry at 0x80001000
Closing network.
Starting program at 0x80001000
CPU revision is: 00029007

Primary instruction cache 8kb, linesize 16 bytes (2 ways)
Primary data cache 4kb, linesize 16 bytes (2 ways)

Linux version 2.4.20 (root@localhost.localdomain) (gcc version 3.0 20010422
(prerelease) with bcm4710a0 modifications) #11 Wed Sep 22 12:13:32 EST 2004
Determined physical RAM map:
 memory: 00800000 @ 00000000 (usable)
On node 0 totalpages: 2048
zone(0): 2048 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line: root=/dev/mtdblock2 noinitrd console=ttyS0,115200
CPU: BCM4712 rev 1 at 200 MHz
Calibrating delay loop... 199.47 BogoMIPS
Memory: 6424k/8192k available (1255k kernel code, 1768k reserved, 108k
data, 64k init, 0k highmem)
Dentry cache hash table entries: 1024 (order: 1, 8192 bytes)
Inode cache hash table entries: 512 (order: 0, 4096 bytes)
Mount-cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer-cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 2048 (order: 1, 8192 bytes)
Checking for 'wait' instruction...  unavailable.
POSIX conformance testing by UNIFIX
PCI: Fixing up bus 0
PCI: Fixing up bridge
PCI: Fixing up bus 1
Linux NET4.0 for Linux 2.4
```

```
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
Starting kswapd
devfs: v1.12c (20020818) Richard Gooch (rgooch@atnf.csiro.au)
devfs: boot_options: 0x1
pty: 256 Unix98 ptys configured
Serial driver version 5.05c (2001-07-08) with MANY_PORTS SHARE_IRQ
SERIAL_PCI enabled
ttyS00 at 0xb8000300 (irq = 3) is a 16550A
ttyS01 at 0xb8000400 (irq = 0) is a 16550A
PPP generic driver version 2.4.2
 Amd/Fujitsu Extended Query Table v1.0 at 0x0040
number of CFI chips: 1
Flash device: 0x200000 at 0x1c000000
Physically mapped flash: cramfs filesystem found at block 742
Creating 5 MTD partitions on "Physically mapped flash":
0x00000000-0x00020000 : "pmon"
0x00020000-0x001f0000 : "linux"
0x000b99e0-0x001f0000 : "rootfs"
0x00004000-0x00006000 : "profile"
0x001f0000-0x00200000 : "nvram"
sflash: found no supported devices
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 512 bind 1024)
ip_conntrack version 2.1 (64 buckets, 512 max) - 344 bytes per conntrack
ip_tables: (C) 2000-2002 Netfilter core team
ipt_time loading
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
NET4: Ethernet Bridge 008 for NET4.0
802.1Q VLAN Support v1.7 Ben Greear <greearb@candelatech.com>
All bugs added by David S. Miller <davem@redhat.com>
VFS: Mounted root (cramfs filesystem) readonly.
Mounted devfs on /dev
Freeing unused kernel memory: 64k freed
Using /lib/modules/2.4.20/kernel/drivers/net/et/et.o
Using /lib/modules/2.4.20/kernel/drivers/net/wl/wl.o
Hit enter to continue...Set name-type for VLAN subsystem. Should be visible
in /proc/net/vlan/config
Added VLAN with VID == 0 to IF -:eth0:-
Added VLAN with VID == 1 to IF -:eth0:-
WARNING:  VLAN 1 does not work with many switches,
consider another number if you have problems.
wlconf: vlan0 failed (-1)
info, udhcp server (v0.9.8) started
nas: No such file or directory
wlconf: vlan1 failed (-1)
info, udhcp client (v0.9.8) started
vlan1 dhcp
vlan1: No such process
nas: No such file or directory
connect: Network is unreachable
Hit enter to continue...debug, Sending discover...
debug, Sending discover...
debug, Sending discover...
debug, Sending discover...
debug, Sending discover...
```

5 References

5.1 Forums / Wikis / Home Pages

5.1.1 Hacking the Belkin F5D7230-4 Version 1444 router

http://www.efn.org/~rick/work/f5d7230/

Rick Bronson, 2004

5.1.2 Belkin F5D7230-4 - SeattleWireless

http://gir.seattlewireless.net/index.cgi/Belkin_20F5D7230_2d4

Wiki, July 2004

5.2 Distributions

5.2.1 Sveasoft

http://sveasoft.com/

5.2.2 OpenWRT

http://openwrt.org/

5.2.3 Belkin Firmware

http://belkin.com/support/download/download.asp?download=F5D7230-4

5.2.4 Linksys: GPL Code

http://www.linksys.com/support/gpl.asp

5.2.5 Belkin: GPL Code (Welcome to Belkin!)

http://web.belkin.com/support/gpl.asp

5.3 Miscellaneous

5.3.1 Digitally Imported Radio

http://di.fm/

5.3.2 Know Your Enemy: Statistics

http://project.honeynet.org/papers/stats/

HoneyNet Project, July 2001.

6 Contact

6.1 Additions, Modifications and Deletions

For changes to this document, please refer to the author and revision history blocks in the control page. Please report errors or omissions to the author.

6.2 Consultation

If you would like to discuss wireless router firmware or other concepts related to this paper, then please contact the author;

Ian Latter

Late night coder …

MidnightCode.org

Email: ian dot latter at midnightcode dot org

Subject: OpenWRT on the Belkin F5D7230-4