WHITE PAPER

OpenWRT on the Belkin F5D7230-4

Attaining console access via the native Belkin serial port for OpenWRT development

CONTROL PAGE

Document Approvals

Approved for Publication:

Author Name: Ian Latter
9 November 2004

Document Control

Document Name: OpenWRT on the Belkin F5D7230-4; Attaining console access via the native Belkin serial port for OpenWRT development
Document ID: openwrt on the belkin f5d7230-4 - serial console.doc-Release-0.2(250)
Distribution: Unrestricted Distribution
Status: Release
Disk File: C:\papers\OpenWRT on the Belkin F5D7230-4 - Serial Console.doc

Version	Date	Release Information	Author/s
0.2	*09-Nov-04*	Release / Unrestricted Distribution	Ian Latter
0.1	*07-Nov-04*	Release / Unrestricted Distribution	Ian Latter

Distribution

Version	Release to
0.2	MidnightCode.org (Julian updated circuit; added Wolf serial note)
0.1	MidnightCode.org

Table of Contents

1 Overview

1.1 In brief

In this second 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 locate the device's hardware serial console.

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 has not ensued. Hidden or unknown errors in the boot process have 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.

By collating a mass of publicly available information, and experimenting with a single unit, the paper concludes by providing a physical console into the device, providing local root user level access, and a schematic diagram for a solder-less project that will allow individuals to try this experiment for themselves. It is hoped that this information can be used to debug open source firmware and to then adapt the OpenWRT, and Sveasoft, embedded Linux distributions for this Belkin router.

1.2 History

This paper represents a third week's work and a fourth week's documentation. It continues where the previous document (*OpenWRT on the Belkin F5D7230-4 – Understanding the Belkin extended firmware for OpenWRT development*) 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.

This experiment was conducted in three sessions; one long Friday evening shared with Ray Poularas, Julian Bale and Ian Latter laying down the ground work and getting the entire solution built, but with only minor success; the following Sunday with Ian Latter rebuilding the project from scratch in parallel to the Friday night work, but with no further success; and the following Tuesday night shared with Julian Bale and Ian Latter, where Julian diligently installed and tested each piece of the puzzle until the only possible, logical outcome, could have been success.

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 – why a serial console is needed

2.1 Since the last paper

After resolving firmware data structure issues and creating a root shell on the F5D7230-4 in the previous paper, development continued in an effort to load an OpenWRT firmware into the device.

This was not successful.

The result is a device that has booted (or at least, is not continually boot cycling) but refused to respond to network requests of any description - from ICMP “ping” through to TCP web interface access – the test PC couldn’t even ARP for the Belkin MAC address.

2.2 Why, then, a serial console?

From the failure above, there is no indication as to what the fault could be. I.e. the fault could be anything, like;

- The OpenWRT software didn’t configure the network interface using NVRAM settings (putting it on an alternate default network)?
- OpenWRT doesn’t have drivers for the F5D7230-4 ethernet hardware?
- The kernel was unable to find the root file system, as maybe it was in a different location in the Belkin, to where it would have been in the Linksys?
- The unit is continually soft rebooting, but as the OpenWRT software doesn’t have LED bank drivers, no external indication is available?

Etc.

A serial console is fundamental to the ability to reliably debug new, open source, firmware.

3 Locating and interfacing with the native serial console

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 serial console root shell for further firmware debugging and exploration.

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

If you choose to follow these instructions then you're going to have to get your hands dirty. You will need to pull your router apart, you may break it – the decision to go ahead with this project is yours and yours alone.

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

<u>No warrantee is implied or intended when you choose to follow these instructions</u>.

Proceed at your own risk

3.2 Getting under the hood

So we move forward with the wise words of Professor Bill (William) Caelli, AO, of Queensland University of Technology;

```
Imagine you bought a Holden and the bonnet was welded down to stop you
looking at the motor – there'd be an outcry … but in the IT game they get
away with it, and one more freedom is lost.
```

See; *Computerworld* **|** *Microsoft and Hollywood hit the control button*

And, with the wit of Tim "the Tool Time" Taylor in Home Improvement episode 168;

```
Ian:     Ooo, what do you have under the hood?
Tim:     [Mockingly] What do I have under the hood?
         [Tim and Ian go to garage]
```

See; *Script 7.17 – Taking Jill for Granite*

The F5D7230-4 is made up of a three-part pre-molded plastic casing. The top (silver) and the bottom / base (charcoal) halves make up the bulk of the casing, but the rear of the unit – the flat part with the antenna and the ethernet ports, etc - is a third piece that slots in between the other two.

In order to find the serial port, the case must be dismantled.

There are two screws in the base of the device that are screwed into the top / lid through the mother-board inside. To remove these screws, lift the two lower corners of the sticker on the base, and use a small phillips-head screwdriver, as shown in this picture;

With a flat-head screwdriver, gently wedge in between the top and bottom halves of the case and twist slightly. Repeat this process along the front of the case and then down the sides, as demonstrated in this picture;

Be gentle but firm with this process, as there are eight plastic tabs inside the lid of the device, and they can snap off (there are two on the front and three on either side).

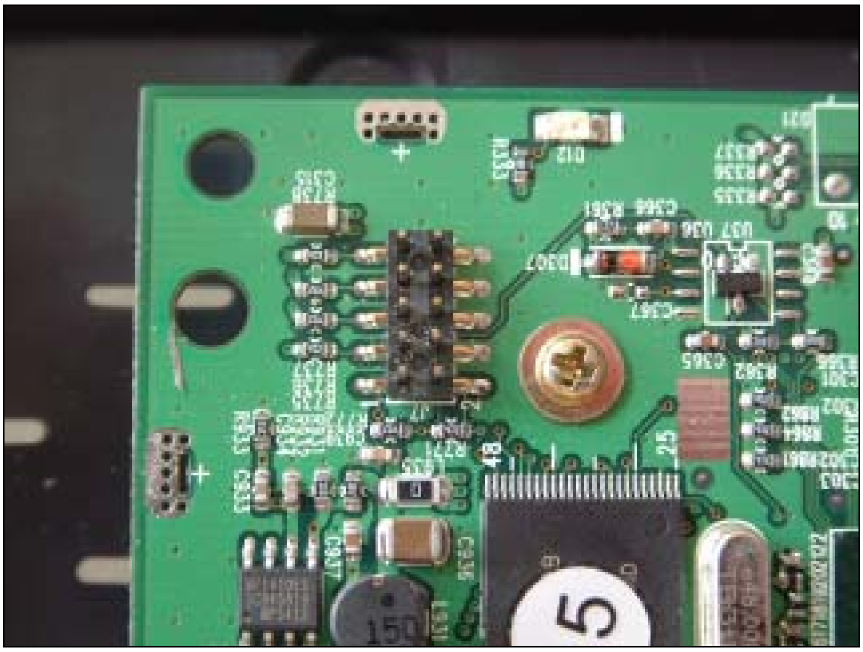
The inside of lid is shown here, with the two screw posts clearly visible. See, also, that the LEDs are not connected to the lid – that clear plastic channels "forward" the light from the micro-LEDs that are surface mounted on the mother-board;

With the casing dismantled, you can now see the inner beauty of Belkin router, model number F5D6230-4;

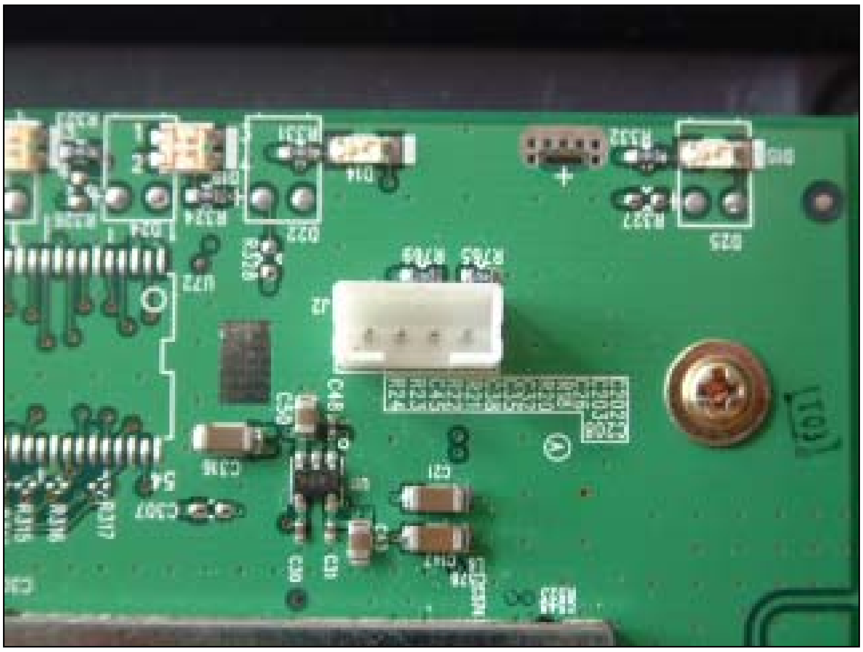
3.3 Searching for the serial port

Like the Belkin, different versions of the Linksys hardware appear to have different physical characteristics (pinouts).

The Linksys' seem to come with at least one large header block – of 10pins – that includes two serial interfaces. The F5D7230-4 also comes with a large, 10pin, header block (labeled J7), pictured here;



Unlike the Linksys devices, the Belkin also comes with a smaller, white 4pin block, labeled J2 and pictured here;



Confident that somewhere in either J7 or J2 there was a serial port, testing began on each pin to evaluate them for suitability.

The first round of testing was performed with a logic probe. However, as the serial interface was running at 115,200 bits per second (known due to the kernel command-line parameter passed by the boot loader – found via the output of /proc/kmsg in the previous paper), and as the logic probe was unable to compensate for different data rates (i.e. we had bought the cheapest logic probe available), all that this revealed was which pins were logically on or logically off, as an average over the sampling period of the probe – ultimately, not particularly useful.

The second round of testing was performed with a multi-meter. The meter was able to provide information on ports and their voltages. We now knew which ports were live and which were ground – and of the live ones, what voltages were registering.

3.4 Accessing more information

Not surprisingly there are a number of similarities between the Belkin 54g and the Linksys 54g hardware specifications – due to the common designs and chipsets.

Taking advantage, then, of the vast amount of public Linksys WRT54G(S) information, it was possible to get a running start into the discovery process.

Two key pieces of information were made available during the discovery process;

- A cross post by HWA, of TWrecks' Sveasoft forum contribution, revealed that J2 (also labeled as CKN16) had, according to TWrecks' oscilloscope, one pin giving out the 115,200 bits per second data rate.
- Two circuits published by Linksys serial port discoverers contributed the TTL to RS232 converter that we didn't know we needed. The primary was from Rod Whitby, but Rod's MAX233 wasn't available locally – the secondary was from Melbourne Wireless where they used the MAX232 (we could get an equivalent to this – the HIN232)

However, while the remaining work was largely trial and error, two pieces of advice were gleaned from the discovery process (one was offered by Rod Whitby directly);

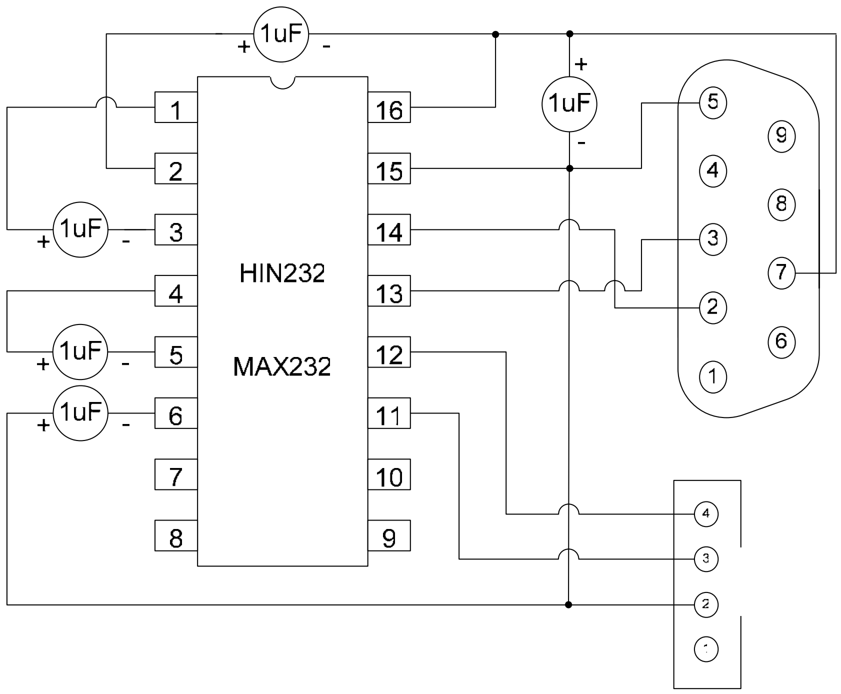
- Look for +3 to +5volt pins as these are likely to be TTL logic circuits. (All live pins showed +3.3volts which was very promising).
- Look for a voltage / logic change across the data (transmit) pins during a power cycle. (three live pins showed a one-second logic change on power cycle, which was also very promising)

It was this mix of information, combined with common sense and patience, which eventually lead to our success.

3.5 Building the TTL to RS232 circuit

As the IO interface on the Belkin is 3.3volts, it is insufficient to drive the RS232 interface on a personal computer. This meant building a converter in order to probe for the serial console.

The MAX232 / HIN232 is a common line driver for interfacing low voltage TTL circuits with standard RS232 circuits. Included, here, is the circuit diagram that we developed after fixing out mistake (below) – based on the manufacturer's reference circuit it was designed to avoid using an external 5volt power source by drawing +5volts from the connected computer's serial port;



For easy experimentation, the prototype was constructed on a solder-less bread-board system.

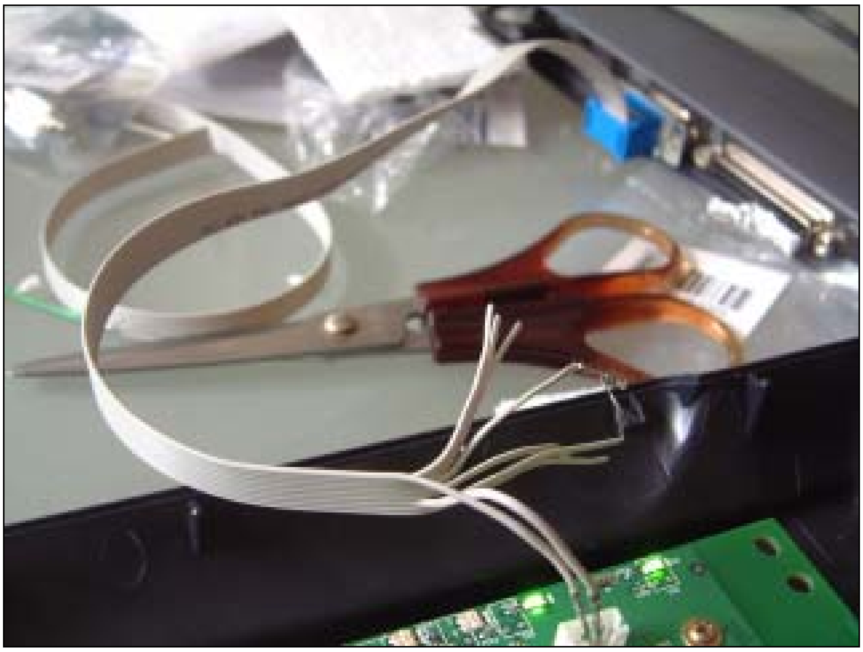
Other than the 232 chip, the circuit requires five 1uF electrolytic capacitors, one 9pin female serial adapter to connect to the PC, and some sort of pin-headers for connecting to the Belkin (Sveasoft's Wolf recommends "KK-connectors").

The reason why the Linksys experimenters use the 233 chips is because they are 3.3volt driven, allowing them to be powered from the Linksys board directly.

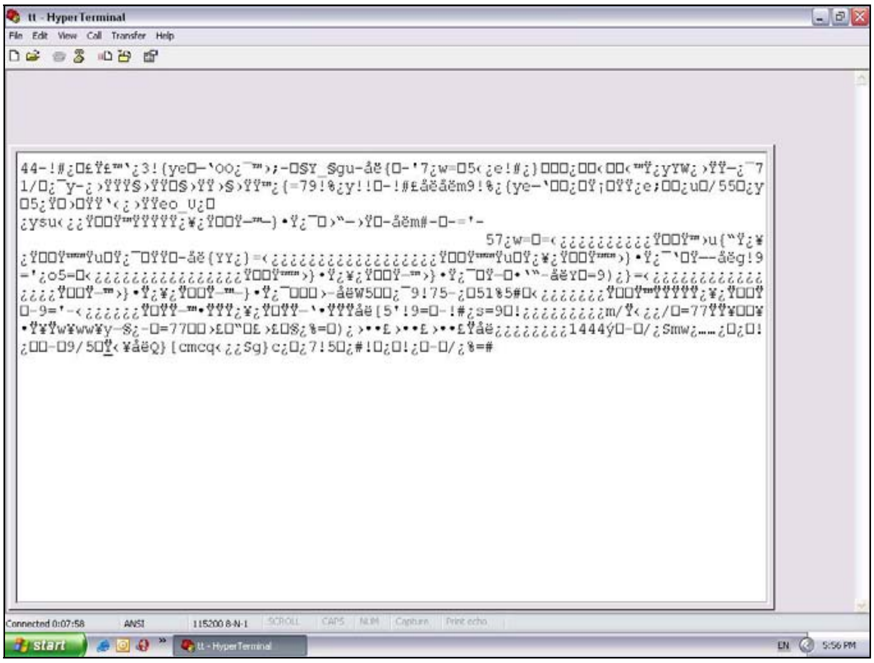
An alternative to the home-grown solutions are any of a number of pre-built solutions that can be found online. See the A232DBHv3 from the Comp Sys web site (reference) at the end of this document.

3.6 Direct connection mistake

Unfortunately the first prototype converter circuit we constructed was faulty, so we jinxed our serial port probing. In disgust, we tried the live pins on J2 connected directly to the serial port on the PC, following the hint that oscilloscope found - the 115,200 bits per second line rate;



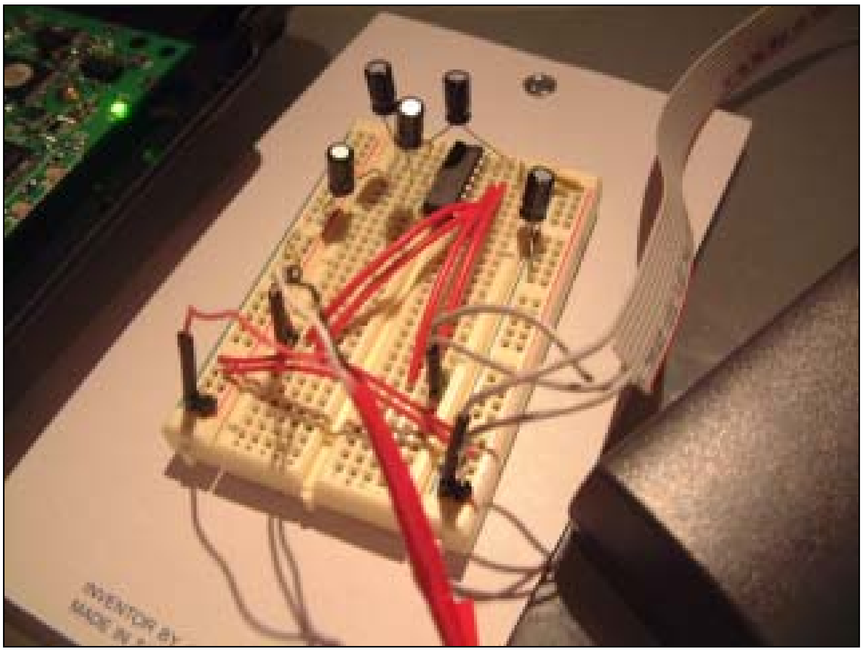
If it hadn't have been for the remarkably well timed output (that appeared to run in sync with a 2-3 second kernel boot sequence) we would have gone back to repairing the TTL to RS232 circuit as a priority. Shown here – the garbage output coming from the Belkin, with a direct connection to the PC serial port;



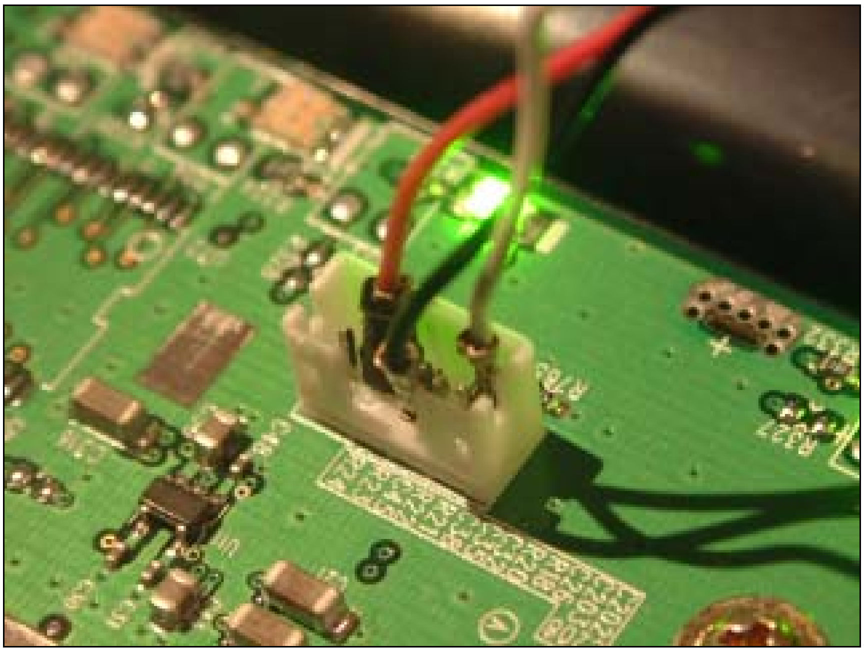
However, as the output through HyperTerminal was optimal at 115,200 bits per second with eight data bits, no parity bits and one stop bit (8n1) we were drawn to the possibility of software misconfiguration, over a hardware fault. The garbage output (above) is now understood to be the sign of a missing TTL to RS232 converter.

3.7 Using the TTL to RS232 circuit correctly

We were confident that we had the found the right pins. So, the converter circuit was rebuilt from scratch, with each component tested for its affect on the overall circuit;

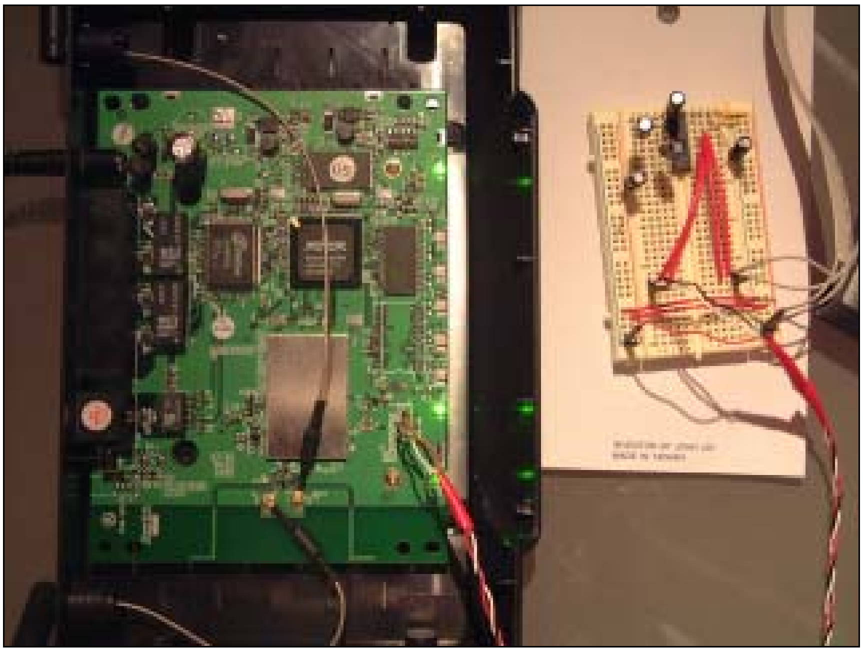


Unsure of the final J2 pin configuration, socket header pins were used – some with heat-shrink for shielding;



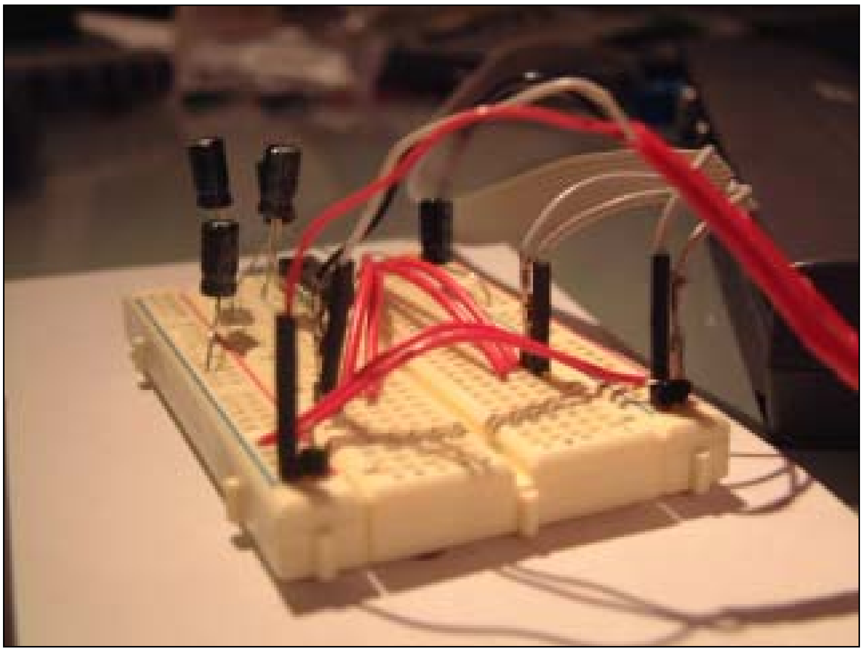
The testing was driven now by a strict logical process. Starting with two PCs connected via serial ports and communicating via HyperTerminal, we moved through a full cycle of tests – each bringing the solution closer to the ultimate “Belkin to bread-board to PC” outcome.

Seen here, the F5D7230-4 wired to the bread-board;



We entered into the final test, uncertain of some of the previous measurements, but unable to prove their accuracy to ourselves. Finally, with the circuit connected, and the Belkin rebooted, output came through perfectly (as readable ASCII).

Shown here, the fully laid out bread board with all components in their final (functional) configuration;



Once this was achieved, a further 10minutes was spent enabling the receive capability on the Belkin (transmit on the PC – console input). An overview of the final work-bench layout – with the unseen experimenters giddy from the lack of food, sleep and serial access;

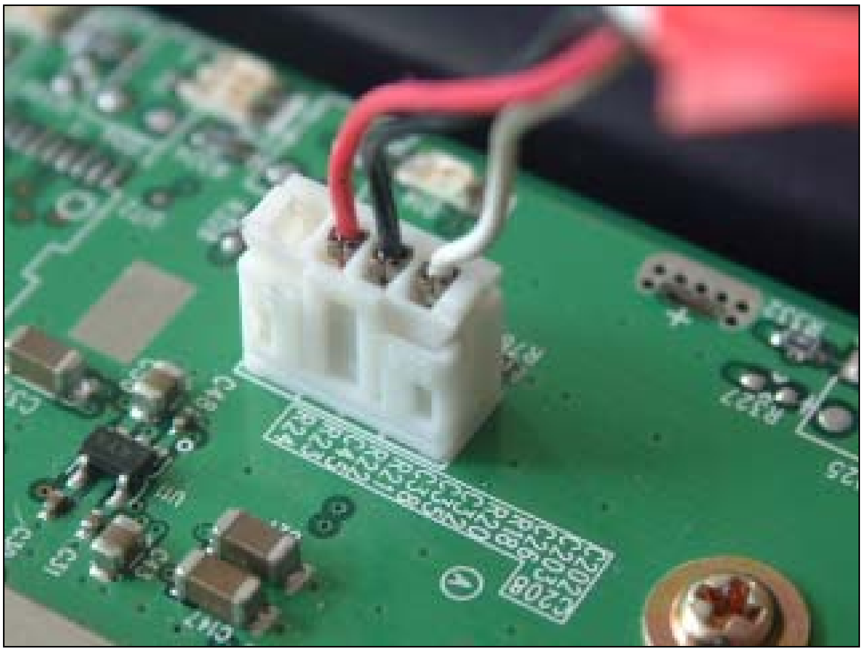
3.8 The surprise of success

With much love and admiration, even the flowers smiled and sighed at the sign of a successful root serial console – operating at 115,200 bps (8n1);

3.9 Finishing touches

After the event, the J2 header was replaced with an internal PC audio connector (used to connect the analogue CDROM audio output signal to the PC sound card). This provided a better connection and some sort of reasonable electrical isolation.

Now it looks like a bought one;



4 Findings

4.1 Pinouts revealed

This table shows the known pinouts of the four pins on the J2 connector for the different versions of the F5D7230-4 hardware that have been found locally;

Version Number	1	2	3	4	Pinout Description
version 1000au	pwr	gnd	txd	rxd	Power; Ground; Transmit Data; Receive Data

Table 1 - Pinouts for known tested F5D7230-4 hardware labeled versions

Note; Pin 1 is on the left hand end of the J2 connector, when the indent / key is facing you, as depicted in "Finishing touches" above.

4.2 A second serial port?

We were confident that there was a serial console on this device, not purely due to the reports of Linksys users. In the first paper, the /proc/kmsg output (section 6.2) indicated that there was a serial interface already configured via a 16550A on IRQ3, at 0xb8000300. However, that same output also indicates that there is a second serial interface – a 16550A on IRQ0, at 0xb8000400.

It is common knowledge that the Broadcom chipset supports two serial interfaces (from the Linksys arenas). If it existed it would be most useful to find that second interface. Sveasoft's Wolf has indicated that the IRQ0 configuration is common amongst the Linksys devices, and that a "setserial" configuration may be all that is required to make it operational – that just leaves finding the pins on the board ...

4.3 Root serial console

The Linux distribution that Belkin have embedded, accepts a carriage return at any stage during the boot process, as the initiator for an interactive root shell. It is believed that this is simply the visual result of the BusyBox terminal start process (Press Enter to use this terminal) combined with the console / boot-process output running out of the same TTY.

With an exploratory platform available, it is now possible to interact directly with the native Linux distribution, and to monitor the progress of any alternative firmware consoles.

4.4 Apres-ski

Not long (within minutes) after first details of the working serial console were published, we were informed of a new link just published – "Hacking the Belkin F5D7230-4 Version 1444 router", by Rick Bronson. To date, Rick has done a good deal more work at the console of a Belkin than anyone else (outside of Belkin Inc) that we are aware of.

4.5 Console output dump

Included here is the dump of a complete boot for a factory default Belkin F5D7230-4 (version 1000au);

```
Decompressing..........done
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 (lchen@penguin.askey.com) (gcc version 3.0 20010422
(prerelease) with bcm4710a0 modifications) #8 Mon Dec 1 20:51:49 PST 2003
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"
0x000b99d8-0x001f0000 : "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
Using /lib/modules/2.4.20/kernel/drivers/net/led/led.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.
killall: iappd: no process killed
iappd: No such file or directory
No interface specified. Quitting...
info, Moreton Bay DHCP Server (v0.9.5) started
===wan_ifname=vlan1, ifname=vlan1
killall: upnp: no process killed
info, Moreton Bay DHCP Client (v0.9.5) started
Hit enter to continue...debug, Sending discover...
No interface specified. Quitting...
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

5.1.3 broadband >> Forums >> Belkin >> Belkin Firmware Image Header

http://www.clanspace.com/forum/remark,10837191~mode=flat~days=9999~start=22

hwa, October 2004

5.1.4 Belkin F5D7230-4 Circuit Board and Info

http://www.linux-hacker.net/misc/F5D7230/

hardware1.

5.2 Serial interface information

5.2.1 Linksys WRT54G / WRT54GS Dual Serial Port Mod

http://www.rwhitby.net/wrt54gs/serial.html

Rod Whitby.

5.2.2 OLD Melbourne Wireless site – see http://melbourne.wireless.org.au/

http://melbourne.wireless.vic.asn.au/wiki/?minitar

Melbourne Wireless Inc, 2002.

5.2.3 Comp Sys MAX232 Adapters

http://www.compsys1.com/workbench/On_top_of_the_Bench/Max233_Adapter/max233_adapter.html

Comp Sys.

5.2.4 WRT54G & WRT54GS Serial Adaptor

http://www.theforestnetworking.com/sveasoft/serial.php

Sveasoft's Wolf.

5.3 Hardware leads (due to PCB markings)

5.3.1 Broadcom Corporation – BM94712 – 802.11b/54g™ Integrated Wireless AP/Router Reference Design

http://www.broadcom.com/products/product.php?product_id=BCM94712&category_id=

Broadcom Corporation.

5.3.2 Broadcom Intros New Integrated 11g Processor

http://www.wi-fiplanet.com/news/article.php/2238881

WiFi Planet, July 2003.

5.3.3 MoiMeme : LinksysWRT54G

http://rage.against.org/LinksysWRT54G

MoiMeme, October 2004.

5.3.4 EM638165TS7

http://www.capitalelect.co.il/all_stock/E/EM638165TS7_ETRONTEC__il.htm

Capital Electronics Ltd.

5.3.5 Re(3): was is das für ein Motherboard

http://forum.geizhals.at/t243853,1442999.html?sort=time

Geizhals Preisvergleich, April 2004.

5.3.6 LCD inverter reference

“D-YING Maxdata Eton Pro 001-00975-C20 E150630 94V-0”

http://laptops.home.att.net/Laptop-LCD-inverters-ref.htm

Laptop Repair.

5.3.7 d-ying Electronics, Inc. professional manufacturer of PrintedCircuit Board (PCB)

http://www.iyuyu.com/web/dying/tw/index.php

TSU D-ying Electronic Inc.

5.4 Distributions

5.4.1 Sveasoft

http://sveasoft.com/

5.4.2 OpenWRT

http://openwrt.org/

5.4.3 Belkin Firmware

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

5.4.4 Linksys: GPL Code

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

5.4.5 Belkin: GPL Code (Welcome to Belkin!)

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

5.5 Articles / Media

5.5.1 Script 7.17 – Taking Jill for granite

http://www.hiarchive.co.uk/script.php?s=7&e=17

Home Improvement Archive, November 2003.

5.5.2 Computerworld | Microsoft and Hollywood hit the control button

http://www.computerworld.com.au/nindex.php?id=1074796437&eid=-255

Computerworld, May 2003.

5.6 Miscellaneous

5.6.1 Digitally Imported Radio

http://di.fm/

5.6.2 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

