Here are the steps (and some mistakes) taken to open up the 2704n router

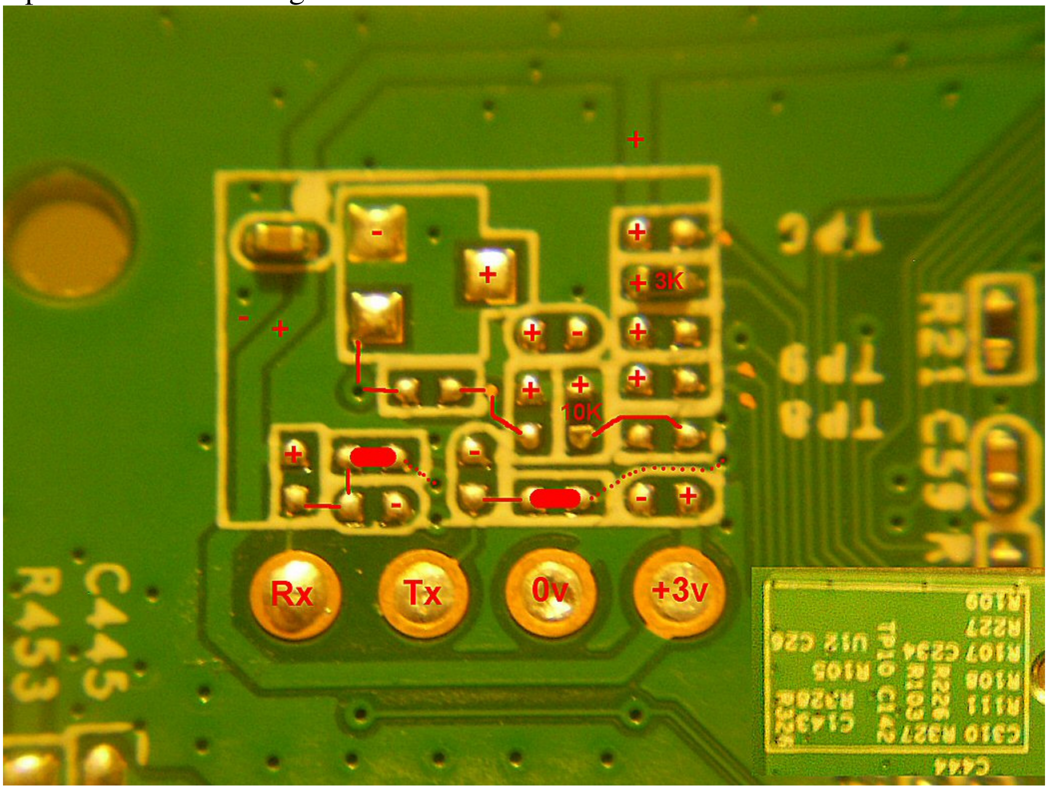
Some requirements are:

Soldering Iron
Parallel Port (or some USB gadget equivalent)
Serial Port
Hex Editor
CRC32 calculator
Linux (via a Virtual Machine) just to recompress the file-system
MIPS be disassembler
2 pairs of Glasses

You don't need all the requirements, because I can supply the information needed, it's just if you want to do it step-by-step.

Firstly find the Serial/UART Port.

In the top corner on the PCB, it looks like the Serial connection, but its missing a few components. After tracing some of the wires I can see what-is-what.



I bridged the thicker red lines (R327 & R328) and used the normal parameters: 115K, 8, N, 1. and I had the serial connection working.
Additional: I believe the connections on the right are the JTAG, but I did not peruse that avenue.

So after a power up I got this info from the Port:
(I had to fudge some of this output. I did not save the original)

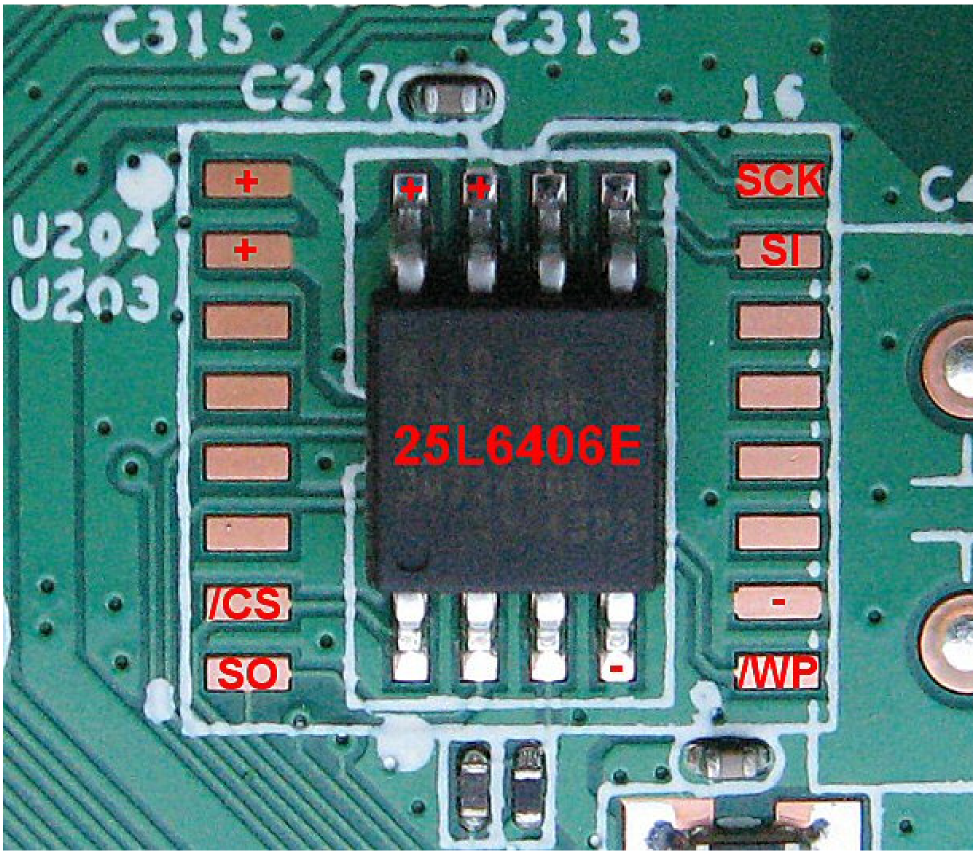
```
HELO
CPUI
L1CI
DRAM
----
PHYS
PHYE
DDR1
333H
SIZ3
SIZ2
RACE
PASS
----
ZBSS
CODE
DATA
L12F
MAIN


CFE version 7.273.1 for BCM96318 (32bit,SP,BE)
Build Date: Tue Nov 18 11:25:16 CST 2014 (cookiechen@sz01017.ads.local)
Copyright (C) 2005-2012 SAGEMCOM Corporation.

HS Serial flash device: name MX25L64, id 0xc217 size 8192KB
Total Flash size: 8192K with 2048 sectors
Chip ID: BCM6318B0, MIPS: 333MHz, DDR: 333MHz, Bus: 167MHz
Main Thread: TP0
Total Memory: 33554432 bytes (32MB)
Boot Address: 0xb8000000

Booting from only image (0xb8010000) ...
```

Then it went DEAD. The buggers disabled it!
I noticed that if I press some keys, it delayed the boot-up a bit. But I did not get any further.

There maybe a command I can type here? But the ones I tried from the internet did not work.

I swear I've seen "cookiechen" before?

Backup the SPI Flash for interrogation



www.zlgmcu.com/mxic/pdf/NOR_Flash_c/**MX25L6406E**_DS_EN.pdf

I connected the pads to a parallel port, however you will need to lift the 2 power pins from the board, to isolate the chip.
I used pin 1 from the parallel port to power the chip, it requires 20ma and that's not (normally) a problem.

I made a program to Read & Write the flash, in-order to understand the SPI Protocol better. But I will not explain it here. You will be able to find a program from the Internet to do it. (Remember it's 3.3v, 8MB)

You can download the original full flash here:
https://drive.google.com/file/d/0B4-Ln6UubyEecl9vRHhZMG9GRGs/
* I have removed my identity from it *

Looking at the Flash Layout

```
0x000000  ++++++++++++++
          + CFE        + < 0x580 - 0x97F = NVRAM
0x010000  ++++++++++++++
          + BCM TAG    +
0x010100  ++++++++++++++
          + ROOT FS    +
0x4A3100  ++++++++++++++
          + KERNEL     +
0x5C1600  ++++++++++++++
          + EMPTY      +
0x7E0000  ++++++++++++++
          + POSSIBLE   +
          + BACKUP     +
          + CONFIG ?   +
0x7F0000  ++++++++++++++
          + CONFIG     + < 40K is allocated
0x7FA000  ++++++++++++++
          + DEFAULT    +
          + PASSWORDS  +
          ++++++++++++++
```

The ROOT-FS can be extracted and opened with 7zip.
I have not been able to uncompress the LZW 'config' section. It appears that the dictionary part is missing.

CFE / NVRAM:
0x684 – stores the Board ID
0x69B – PSI size value   (Persistent storage information)
0x808 – Serial Number
0x8FD – Some unknown string, looks like its bin-hex encoded.
0x97C – CRC32 of NVRAM (0x580 to 0x97F)

If you corrupt the nvram section and write it back. The next power on asks you for some details (via the Serial Port). So the serial port works when it needs to !

I can't remember… but… if you remove (zero out) the unknown string, the serial will start outputting the normal Kernel log while it's booting-up.

Changing the ROOT File System

In the directory /webs-EN/ you'll see:
menuBcm.js – which has lots of disabled features called 'bugs' ?
so, first swap "menuBcm.js" for "menuBcm_withdect.js" which saves a lot of time. Then I added a couple of extra options in the new menuBcm.js. Like view the config in text, not the AES-CBC encrypted one.

A program like WinMerge shows the difference in a really nice way.

For the missing html files I used this GPL source code:
http://oss.sky.com/SkyHD/SKY-IHR-2-1-s-3761-R-consumer-release.tar.gz
Because it uses the same Kernel version 2.6.30 (and even the same magic number)

This Kernel appears to have some missing options like UNIX98_PTY, so I compiled a Static Telnet Binary from here. Using the Tool-chain from the Source above.

"HTTPD" binary adjustments:

Some of web pages are locked from editing, it says something like:
"You are not allow to access this page"

When I looked at the binary I saw these pages inside of it:

```
192.168.1.254/rtroutecfg.cmd?action=view
192.168.1.254/arpview.cmd?action=view
192.168.1.254/backupsettings.cmd?action=view
192.168.1.254/seclogreset.cmd?action=view
192.168.1.254/security_log.cmd?action=view
192.168.1.254/seclogview.cmd
192.168.1.254/voicelogview.cmd
192.168.1.254/logview.cmd
192.168.1.254/scvrtsrv.cmd?action=view
192.168.1.254/devtoapp.cmd?action=view
192.168.1.254/addscvrtentry.cmd?action=view
192.168.1.254/firewallcfg.cmd?action=view
192.168.1.254/wancfgplusnet.cmd?action=view
192.168.1.254/scprttrg.cmd?action=view
192.168.1.254/scoutflt.cmd?action=view
192.168.1.254/scinflt.cmd?action=view
192.168.1.254/scmacflt.cmd?action=view
192.168.1.254/qoscls.cmd?action=view
192.168.1.254/scdmz.cmd?action=view
192.168.1.254/dslatm.cmd?action=view
192.168.1.254/ethwan.cmd?action=view
192.168.1.254/l2tpacwan.cmd?action=view
192.168.1.254/storageservicecfg.cmd?action=view
192.168.1.254/wancfg.cmd?action=view
192.168.1.254/wanifc.cmd?action=view
192.168.1.254/wansrvc.cmd?action=view
192.168.1.254/wanL3Edit.cmd?action=view
192.168.1.254/statsxtm.cmd?action=view
192.168.1.254/statswan.cmd?action=view
192.168.1.254/adslcfgadv.cmd?action=view
192.168.1.254/adslcfgtone.cmd?action=view
```

```
192.168.1.254/engdebug.cmd?action=view
192.168.1.254/dumpcfgdynamic.cmd?action=view
192.168.1.254/dumpcfg.cmd?action=view
192.168.1.254/dumpmdm.cmd?action=view
192.168.1.254/dumpmsg.cmd?action=view
192.168.1.254/qospolicer.cmd?action=view
192.168.1.254/qosqueue.cmd?action=view
192.168.1.254/qosmgmt.cmd?action=view
192.168.1.254/dhcpdstaticlease.cmd?action=view
192.168.1.254/prmngr.cmd?action=view
192.168.1.254/urlfilter.cmd?action=view
192.168.1.254/portmap.cmd?action=view
192.168.1.254/ripcfg.cmd?action=view
192.168.1.254/wlmacflt.cmd?action=view
192.168.1.254/wlwds.cmd?action=view
192.168.1.254/wlstationlist.cmd?action=view
192.168.1.254/ddnsmngr.cmd?action=view
192.168.1.254/certlocal.cmd?action=view
192.168.1.254/certca.cmd?action=view
192.168.1.254/ipv6lancfg.cmd?action=view
192.168.1.254/tunnelcfg.cmd?action=view
192.168.1.254/ippcfg.cmd?action=view
192.168.1.254/sysinfo.cmd?action=view
192.168.1.254/vstatus.cmd?action=view
192.168.1.254/LanguageIdSet.cmd?action=view
192.168.1.254/LanguageIdDisplaySet.cmd?action=view
192.168.1.254/modconn.cmd?action=view

192.168.1.254/lanvlancfg.html
192.168.1.254/mocacfg.html
192.168.1.254/qosqmgmt.html
192.168.1.254/rtdefaultcfg.html
192.168.1.254/adslcfgc.html
192.168.1.254/xdslcfg.html
192.168.1.254/dslbondingcfg.html
192.168.1.254/upnpcfg.html
192.168.1.254/dnsproxycfg.html
192.168.1.254/standby.html
192.168.1.254/bmu.html
192.168.1.254/wlcfg.html
192.168.1.254/wlsecurity.html
192.168.1.254/wlcfgadv.html
192.168.1.254/wlses.html
192.168.1.254/wlwapias.html
192.168.1.254/wlfon.html
192.168.1.254/voicemgcp_basic.html
192.168.1.254/voicentr.html
192.168.1.254/voicesip_basic.html
192.168.1.254/voicesip_advanced.html
192.168.1.254/voicesip_debug.html
192.168.1.254/voicedect.html
192.168.1.254/updatesettings.html
192.168.1.254/defaultsettings.html
192.168.1.254/seclogintro.html
192.168.1.254/sntpcfg.html
192.168.1.254/resetrouter.html
192.168.1.254/qsmain.html
192.168.1.254/tr69cfg.html
192.168.1.254/logout.html
192.168.1.254/logintro.html
192.168.1.254/logconfig.html
```

(I added on the ‘action=view’ to see them before adjusting any binaries)

When I disassembled the ELF binary, I saw a function called “isPageAllowed”.

```
loc_40B374:
li      $s2, 0xC5FC
la      $t9, isPageAllowed
addu    $s2, $sp, $s2
move    $a0, $s2
jalr    $t9 # isPageAllowed
li      $a1, 0xA
b       loc_40B3F4       # Adjusted Here, (at first)
lw      $gp, 0xED40 + var_0xED20($sp)
la      $t9, log_log
lui     $s0, 0x47
la      $a3, strNotAllowed  # "Not allowed to load..."
li      $a2, 0xF86
li      $a0, 3
addiu   $a1, $s0, strHandle_request
jalr    $t9 # log_log
```

I changed the instruction in the middle (via a Hex editor) and it ‘appeared’ to work. But… after a short while I noticed that things stops responding and the Log was complaining about file locking ???

So…
I looked in the function “isPageAllowed” and it makes reference to some lookup authentication tables. At 0x8DD40 and 0x8E028 (binary file location).

The CMD table format is (0x8DD40):
WORD: Address of cmd string, e.g. “rtroutecfg”
WORD: Address of function to execute
WORD: Auth Level, e.g. 0xB

The HTML table format is (0x8E028):
WORD: Address to html string, e.g. “lanvlancfg”
WORD: Auth Level, e.g. 0xB

When you login as ‘Admin’. You are Auth-Level 10 (0xA). I don’t know how to increase the Admin’s auth-level. So I decreased the pages Auth levels instead.

Yahoo, it worked.

Limited config storage
This is a bit confusing…

The PSI value 0x28 (40K) is passed from the CFE/Bootloader to the Linux Kernel. The binary programs httpd & libcms_core can store about 160K of config in memory.

The library program compressed the config from e.g. 37K to about 12K and passed it to the Kernel to write to flash at 0x7F0000.

But at some point, it cuts the text config to 40K before its compressed?. This is not the case if you upload the config as a 'restore' file.

I don't really understand it, but if I change the CFE PSI value to 0x40 (64K) its helps a lot, and it does not overwrite 'Default Passwords' part of the Flash (0x7FA000).

I have added about a dozen static MAC addresses, and 2 dozen simple outbound firewall rules and it works OK.

By the way… why does my Korean TV need to talk to the Netherlands, Germany and Microsoft HQ !. I only pressed 1 buton !

Recreating the new File System / Image:

On a Linux PC or VM: use ‘unsquashfs’ to extract the Root FS into a directory. You can use the ‘hostTools’ in the sky source code.

After the additions & changes. Use ‘mksquashfs’ to get it back again:

```
../hostTools/mksquashfs ./plusnet-root-fs/ new-squashfs.bin -b 65536
-be -noappend -all-root
```

It will produce a file like this:
https://drive.google.com/file/d/0B4-Ln6UubyEeNnBqWURXTFBmMnc/

Then I (mostly) copied-n-pasted the file onto the original image (carful of the increased size). But it might be better to use a Broadcom image creator like imagetag3 or OpenWRT is going to have something.

So, at this point you have a whole image ready to burn. However it needs an additional tag of 20 bytes at the bottom for the web GUI to accept it.

```
The first 4 bytes is the NOT CRC32 of the whole image (excluding this tag)
The next  4 bytes is the string “6318” - the CPU id
The next  4 bytes might not be needed, but put the CPU id in a byte format
        (0x00 0x00 0x63 0x18)
The next  8 bytes are best left empty 0x00
```

I have tried to upload the Broadcom image (without the CFE or 20 byte footer tag). But I can not get it to work.
The log identifies the 0x100 byte BCM tag at the top of the image, but it complains that another tag is not found?

TIP: never write a whole-image without the CFE. Or you will defiantly need a soldering iron, glasses and a steady hand.

Download the whole-image-with-no-ids:
https://drive.google.com/file/d/0B4-Ln6UubyEeb2l0RUFUekFFOFU/
This can be uploaded via the GUI web interface. But this will overwrite your Serial number & Mac address.

Matt Goring – April 2015

Serial output of the adjusted firmware

```
HELO
CPUI
L1CI
DRAM
----
PHYS
PHYE
DDR1
333H
SIZ3
SIZ2
RACE
PASS
----
ZBSS
CODE
DATA
L12F
MAIN


CFE version 1.0.38-114.185 for BCM96318 (32bit,SP,BE)
Build Date: Wed Dec  5 16:41:36 PST 2012 (williamz@bcacpe-sqa)
Copyright (C) 2000-2011 Broadcom Corporation.

HS Serial flash device: name MX25L64, id 0xc217 size 8192KB
Total Flash size: 8192K with 2048 sectors
Flash split 32 : AuxFS[2686976]
Chip ID: BCM6318B0, MIPS: 333MHz, DDR: 333MHz, Bus: 167MHz
Main Thread: TP0
Total Memory: 33554432 bytes (32MB)
Boot Address: 0xb8000000

Board IP address                  : 192.168.1.1:ffffff00
Host IP address                   : 192.168.1.100
Gateway IP address                :
Run from flash/host (f/h)         : f
Default host run file name        : vmlinux
Default host flash file name      : bcm963xx_fs_kernel
Boot delay (0-9 seconds)          : 1
Board Id (0-4)                    : 96318REF
Primary AFE ID OVERRIDE           : 0x55555555
Bonding AFE ID OVERRIDE           : 0x55555555
Number of MAC Addresses (1-32)    : 11
Base MAC Address                  : 44:e9:dd:01:01:01
PSI Size (1-64) KBytes            : 64
Enable Backup PSI [0|1]           : 0
System Log Size (0-256) KBytes    : 0
Auxillary File System Size Percent: 32
Main Thread Number [0|1]          : 0

*** Press any key to stop auto run (1 seconds) ***
Auto run second count down: 0
Booting from only image (0xb8010000) ...
Code Address: 0x80010000, Entry Address: 0x802760b0
Decompression OK!
Entry at 0x802760b0
Closing network.
Disabling Switch ports.
Flushing Receive Buffers...
0 buffers found.
Closing DMA Channels.
Starting program at 0x802760b0
Dentry cache hash table entries: 4096 (order: 2, 16384 bytes)
Inode-cache hash table entries: 2048 (order: 1, 8192 bytes)
Memory: 27828k/31488k available (2480k kernel code, 3660k
reserved, 484k data, 124k init, 0k highmem)
Calibrating delay loop... 331.77 BogoMIPS (lpj=165888)
Mount-cache hash table entries: 512
--Kernel Config--
  SMP=0
  PREEMPT=0
  DEBUG_SPINLOCK=0
  DEBUG_MUTEXES=0
Broadcom Logger v0.1 Nov 18 2014 11:33:08
net_namespace: 1128 bytes
NET: Registered protocol family 16
Total Flash size: 8192K with 2048 sectors
registering PCI controller with io_map_base unset
registering PCI controller with io_map_base unset
bio: create slab <bio-0> at 0
pci 0000:01:00.0: PME# supported from D0 D3hot
pci 0000:01:00.0: PME# disabled
pci 0000:02:00.0: reg 10 64bit mmio: [0x000000-0x007fff]
pci 0000:02:00.0: supports D1 D2
pci 0000:01:00.0: PCI bridge, secondary bus 0000:02
pci 0000:01:00.0:   IO window: disabled
pci 0000:01:00.0:   MEM window: 0x10200000-0x102fffff
pci 0000:01:00.0:   PREFETCH window: disabled
PCI: Enabling device 0000:01:00.0 (0000 -> 0002)
PCI: Setting latency timer of device 0000:01:00.0 to 64
BLOG v3.0 Initialized
BLOG Rule v1.0 Initialized
Broadcom IQoS v0.1 Nov 18 2014 11:35:23 initialized
Broadcom GBPM v0.1 Nov 18 2014 11:35:23 initialized
NET: Registered protocol family 8
NET: Registered protocol family 20
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 1024 (order: 1, 8192 bytes)
TCP bind hash table entries: 1024 (order: 0, 4096 bytes)
TCP: Hash tables configured (established 1024 bind 1024)
TCP reno registered
NET: Registered protocol family 1
squashfs: version 4.0 (2009/01/31) Phillip Lougher
squashfs: version 4.0 with LZMA457 ported by BRCM
msgmni has been set to 54
io scheduler noop registered (default)
pcieport-driver 0000:01:00.0: device [14e4:6318] has invalid
IRQ; check vendor BIOS
PCI: Setting latency timer of device 0000:01:00.0 to 64
PPP generic driver version 2.4.2
PPP Deflate Compression module registered
PPP BSD Compression module registered
NET: Registered protocol family 24
PPPoL2TP kernel driver, V1.0
bcm963xx_mtd driver v1.0
File system address: 0xb8010100
```

```
brcmboard: brcm_board_init entry
SES: Button Interrupt 0x0 is enabled
Serial: BCM63XX driver $Revision: 3.00 $
Magic SysRq enabled (type ^ h for list of supported commands)
ttyS0 at MMIO 0xb0000100 (irq = 36) is a BCM63XX
Total # RxBds=1507
bcmPktDmaBds_init: Broadcom Packet DMA BDs initialized

bcmxtmrt: Broadcom BCM6318B0 ATM/PTM Network Device v0.5 Nov 18
2014 11:35:13
GACT probability NOT on
Mirror/redirect action on
u32 classifier
    input device check on
    Actions configured
GRE over IPv4 tunneling driver
TCP cubic registered
Initializing XFRM netlink socket
NET: Registered protocol family 10
IPv6 over IPv4 tunneling driver
NET: Registered protocol family 17
NET: Registered protocol family 15
Initializing MCPD Module
Ebtables v2.0 registered
ebt_time registered
ebt_ftos registered
ebt_wmm_mark registered
802.1Q VLAN Support v1.8 Ben Greear <greearb@candelatech.com>
All bugs added by David S. Miller <davem@redhat.com>
VFS: Mounted root (squashfs filesystem) readonly on device 31:0.
Freeing unused kernel memory: 124k freed
init started: BusyBox v1.17.2 (2014-11-18 11:38:02 CST)
starting pid 143, tty '': '/etc/init.d/rcS'
mount: mounting none on /proc/bus/usb failed: No such file or
directory
starting pid 147, tty '': '-/bin/sh'


BusyBox v1.17.2 (2014-11-18 11:38:02 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.


Loading drivers and kernel modules...

chipinfo: module license 'proprietary' taints kernel.
Disabling lock debugging due to kernel taint
brcmchipinfo: brcm_chipinfo_init entry
Broadcom Ingress QoS Module  Char Driver v0.1 Nov 18 2014
11:33:26 Registered<243>

Broadcom Ingress QoS ver 0.1 initialized
BPM: tot_mem_size=33554432B (32MB), buf_mem_size=5033160B (4MB),
num of buffers=2383, buf size=2112
Broadcom BPM Module Char Driver v0.1 Nov 18 2014 11:33:25
Registered<244>
[NTC bpm] bpm_set_status: BPM status : enabled

NBUFF v1.0 Initialized
Initialized fcache state
Broadcom Packet Flow Cache  Char Driver v2.2 Nov 18 2014
11:33:27 Registered<242>
Created Proc FS /procfs/fcache
Broadcom Packet Flow Cache registered with netdev chain
Broadcom Packet Flow Cache learning via BLOG enabled.
Constructed Broadcom Packet Flow Cache v2.2 Nov 18 2014
11:33:27
bcmxtmcfg: bcmxtmcfg_init entry
adsl: adsl_init entry
Broadcom BCM6318B0 Ethernet Network Device v0.1 Nov 18 2014
11:35:06
ETH Init: Ch:0 - 200 tx BDs at 0xa19d9000
ETH Init: Ch:1 - 200 tx BDs at 0xa1223000
ETH Init: Ch:0 - 953 rx BDs at 0xa1224000
ETH Init: Ch:1 - 100 rx BDs at 0xa12e0800
Error: kerSysRegisterDyingGaspHandler: list head is null
eth0: MAC Address: 44:E9:DD:01:01:01
eth1: MAC Address: 44:E9:DD:01:01:01
eth2: MAC Address: 44:E9:DD:01:01:01
eth3: MAC Address: 44:E9:DD:01:01:01
BCM63XX_USB USB Device not present
insmod: can't insert '/lib/modules/2.6.30/extra/bcm_usb.ko': No
such device
wl: high_wmark_tot=1548
PCI: Enabling device 0000:02:00.0 (0000 -> 0002)
PCI: Setting latency timer of device 0000:02:00.0 to 64
wl: passivemode=1
wl: napimode=0
wl0: allocskbmode=1 currallocskbsz=256
otp_read_pci: bad crc
Neither SPROM nor OTP has valid image
wl:srom/otp not programmed, using main memory mapped srom
info(wombo board)
wl:loading /etc/wlan/bcm43217_vars.bin
Failed to open srom image from '/etc/wlan/bcm43217_vars.bin'.
wl:loading /etc/wlan/bcm43217_map.bin
eth2 Link UP 100 mbps full duplex
wl0: Broadcom BCMa8db 802.11 Wireless Controller
5.100.138.2008.cpe4.12L06B.4
Error: kerSysRegisterDyingGaspHandler: list head is null
message received before monitor task is initialized
kerSysSendtoMonitorTask
Broadcom 802.1Q VLAN Interface, v0.1

===== Release Version 7.273.1 (build timestamp 141118_1136)
=====

Got primary config file from flash (len=56235), validating....
wait instruction: enabled
Netfilter messages via NETLINK v0.30.
device eth0 entered promiscuous mode
ADDRCONF(NETDEV_UP): eth0: link is not ready
ip_tables: (C) 2000-2006 Netfilter Core Team
ip6_tables: (C) 2000-2006 Netfilter Core Team
device eth1 entered promiscuous mode
ADDRCONF(NETDEV_UP): eth1: link is not ready
device eth2 entered promiscuous mode
br0: port 3(eth2) entering forwarding state
device eth3 entered promiscuous mode
```

```
ADDRCONF(NETDEV_UP): eth3: link is not ready
device wl0 entered promiscuous mode
WLmngr Daemon is running
optarg=0 shmId=0
wlevt is ready for new msg...
br0: port 5(wl0) entering forwarding state
*** dslThread dslPid=601
BcmAdsl_Initialize=0xC0110188, g_pFnNotifyCallback=0xC013EE94
open error...
AdslFileLoadImage name /etc/adsl/adsl_phy.bin
pSdramPHY=0xA1FFFFF8, 0x23EE2 0xDEADBEEF
*** XfaceOffset: 0x1C790 => 0x1C790 ***
*** PhySdramSize got adjusted: 0x91DFC => 0xA7ECC ***
AdslCoreSharedMemInit: shareMemSize=98576(98576)
AdslCoreHwReset:  pLocSbSta=81d20000 bkupThreshold=1600
AdslCoreHwReset:  AdslOemDataAddr = 0xA1FC671C
***BcmDiagsMgrRegisterClient: 0 ***
Error: kerSysRegisterDyingGaspHandler: list head is null
dsl kLedStateFail
bcmxtmrt: PTM/ATM Non-Bonding Mode configured in system
nf_conntrack version 0.5.0 (492 buckets, 1968 max)
ssk:error:18.796:rcl_sagemObject:434:init firewall class

rmmod: can't unload 'nf_nat_sip': unknown symbol in module, or
unknown parameter
rmmod: can't unload 'nf_conntrack_sip': unknown symbol in
module, or unknown parameter
monitor task is initialized pid= 244
br0: port 5(wl0) entering disabled state
device wl0 left promiscuous mode
br0: port 5(wl0) entering disabled state
device wl0 entered promiscuous mode
br0: port 5(wl0) entering forwarding state
kerSysBackupPsiGet, strLen = 40960, offset = 0
br0: port 5(wl0) entering disabled state
telnetd: starting
  port: 23; interface: br0; login program: /bin/sh
Setting SSID: "PLUSNET-XXXXX"
Setting SSID: "wl0_Guest1"
Setting SSID: "wl0_Guest2"
Setting SSID: "wl0_Guest3"
wlmngr_wlIfcDown: reset wifi_up_time
wlmngr_wlIfcDown: reset wifi_up_time
wlmngr_wlIfcDown: reset wifi_up_time
wlmngr_wlIfcDown: reset wifi_up_time
device wl0 left promiscuous mode
br0: port 5(wl0) entering disabled state
device wl0 entered promiscuous mode
br0: port 5(wl0) entering forwarding state
acsd: scan in progress ...
acsd: scan in progress ...
acsd: scan in progress ...
acsd: scan in progress ...
acsd: scan in progress ...
acsd: scan in progress ...
acsd: scan in progress ...
acsd: scan in progress ...
acsd: scan in progress ...
acsd: scan in progress ...
acsd: scan in progress ...
acsd: scan in progress ...
acsd: scan in progress ...
acsd: selected channel spec: 0x2b01
wlmngr_wlIfcUp: wifi_up_time=30
```