DNS-320

This page will share some more information about the ShareCenter Pulse (DNS-320).

Hardware

Additional to the product specifications.

CPU	800 MHz Marvell 88F6281 (Kirkwood)
RAM	128 MB
USB	1 USB2.0 Port (front)
LAN	Marvell 88E1118R-NNC1
Cooling	ADDA AD045HB-G73
Kernel	2.6.22.18
Samba	3.2.8
C library	gcc-4.2-glibc-2.5

Reviews & Tutorials

- DNS-320-Page on NAS-Tweaks with further Pictures
- Replacing firmware with Debian Linux

/proc/cpuinfo

```
Processor       : ARM926EJ-S rev 1 (v5l)
BogoMIPS        : 791.34
Features        : swp half thumb fastmult edsp
CPU implementer : 0x56
CPU architecture: 5TE
CPU variant     : 0x2
CPU part        : 0x131
CPU revision    : 1
Cache type      : write-back
Cache clean     : cp15 c7 ops
Cache lockdown  : format C
Cache format    : Harvard
I size          : 16384
I assoc         : 4
I line length   : 32
I sets          : 128
D size          : 16384
D assoc         : 4
D line length   : 32
D sets          : 128

Hardware        : Feroceon-KW
Revision        : 0000
Serial          : 0000000000000000
```

Serial PIN Layout

BusyBox break-in code is 5784468 for Firmware before v2.03. Starting with v2.03 it is set by the ui [http://forums.dlink.com/index.php?topic=42483.msg149905#msg149905].

Jumper: JP3
TX [5]
GND [4]
VD33 [3]

RX [1]

Serial console boot messages

Below the messages sent to the serial console during the system startup, till init is started.

```
 ** MARVELL BOARD: DB-88F6281A-BP LE

U-Boot 1.1.4 (Jun  7 2010 - 10:33:37) Marvell version:
3.4.14.DNS320_02
U-Boot code: 00600000 -> 0067FFF0  BSS: -> 006CEE80
Soc: MV88F6281 Rev 3 (DDR2)
CPU running @ 800Mhz L2 running @ 400Mhz
SysClock = 200Mhz , TClock = 166Mhz
DRAM CAS Latency = 3 tRP = 3 tRAS = 8 tRCD=3
DRAM CS[0] base 0x00000000   size 128MB
DRAM Total size 128MB  16bit width
Flash:  0 kB
Addresses 8M - 0M are saved for the U-Boot usage.
Mem malloc Initialization (8M - 7M): Done
NAND:128 MB
*** Warning - bad CRC or NAND, using default environment
CPU : Marvell Feroceon (Rev 1)
Streaming disabled
Write allocate disabled
USB 0: host mode
PEX 0: interface detected no Link.
Net:   egiga0 [PRIME]
Hit any key to stop autoboot:  0
NAND read: device 0 offset 0x100000, size 0x300000
load addr ....  =a00000
 3145728 bytes read: OK
NAND read: device 0 offset 0x600000, size 0x300000
load addr ....  =f00000
 3145728 bytes read: OK
## Booting image at 00a00000 ...
   Image Name:   Linux-2.6.22.18
   Created:      2010-11-15   9:49:59 UTC
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    2196588 Bytes =  2.1 MB
   Load Address: 00008000
   Entry Point:  00008000
   Verifying Checksum ... OK
OK
## Loading Ramdisk Image at 00f00000 ...
   Image Name:   Ramdisk
   Created:      2010-12-06   7:00:10 UTC
   Image Type:   ARM Linux RAMDisk Image (gzip compressed)
   Data Size:    1563868 Bytes =  1.5 MB
   Load Address: 00e00000
   Entry Point:  00e00000
   Verifying Checksum ... OK
Starting kernel ...

Uncompressing Linux.....................................................................
Linux version 2.6.22.18 (bing@SWTEST1) (gcc version 4.2.1) #22 Mon Nov 15 17:49:27 CST 2010
CPU: ARM926EJ-S [56251311] revision 1 (ARMv5TE), cr=00053977
Machine: Feroceon-KW
Using UBoot passing parameters structure
```

```
Memory policy: ECC disabled, Data cache writeback
CPU0: D VIVT write-back cache
CPU0: I cache: 16384 bytes, associativity 4, 32 byte lines, 128 sets
CPU0: D cache: 16384 bytes, associativity 4, 32 byte lines, 128 sets
Built 1 zonelists.  Total pages: 32512
Kernel command line: root=/dev/ram console=ttyS0,115200 :::DB88FXX81:egiga0:none
PID hash table entries: 512 (order: 9, 2048 bytes)
Console: colour dummy device 80x30
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 128MB 0MB 0MB 0MB = 128MB total
Memory: 123776KB available (4120K code, 251K data, 124K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
NET: Registered protocol family 16

CPU Interface
-------------
SDRAM_CS0 ....base 00000000, size 128MB
SDRAM_CS1 ....disable
SDRAM_CS2 ....disable
SDRAM_CS3 ....disable
PEX0_MEM ....base e8000000, size 128MB
PEX0_IO ....base f2000000, size   1MB
INTER_REGS ....base f1000000, size   1MB
NFLASH_CS ....base fa000000, size   2MB
SPI_CS ....base f4000000, size  16MB
BOOT_ROM_CS ....no such
DEV_BOOTCS ....no such
CRYPT_ENG ....base f0000000, size   2MB

  Marvell Development Board (LSP Version KW_LSP_4.3.4_patch30)-- DB-88F6281A-BP  Soc: 88F6281 A1 LE

 Detected Tclk 166666667 and SysClk 200000000
MV Buttons Device Load
Marvell USB EHCI Host controller #0: c05bb600
PEX0 interface detected no Link.
PCI: bus0: Fast back to back transfers enabled
SCSI subsystem initialized
usbcore: registered new interface driver usbfs
usbcore: registered new interface driver hub
usbcore: registered new device driver usb
NET: Registered protocol family 2
Time: kw_clocksource clocksource has been installed.
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 3, 32768 bytes)
TCP bind hash table entries: 4096 (order: 2, 16384 bytes)
TCP: Hash tables configured (established 4096 bind 4096)
TCP reno registered
checking if image is initramfs...it isn't (no cpio magic); looks like an initrd
Freeing initrd memory: 1527K
cpufreq: Init kirkwood cpufreq driver
XOR registered 1 NET_DMA over 4 channels
XOR 2nd invalidate WA enabled
cesadev_init(c00119d8)
mvCesaInit: sessions=640, queue=64, pSram=f0000000
Warning: TS unit is powered off.
MV Buttons Driver Load
VFS: Disk quotas dquot_6.5.1
Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
squashfs: version 3.3 (2007/10/31) Phillip Lougher
squashfs: LZMA suppport for slax.org by jro
Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
JFFS2 version 2.2. (NAND) ©Â©© 2001-2006 Red Hat, Inc.
SGI XFS with large block numbers, no debug enabled
io scheduler noop registered
io scheduler anticipatory registered (default)
Serial: 8250/16550 driver $Revision: 1.7 $ 4 ports, IRQ sharing disabled
serial8250.0: ttyS0 at MMIO 0xf1012000 (irq = 33) is a 16550A
serial8250.0: ttyS1 at MMIO 0xf1012100 (irq = 34) is a 16550A
RAMDISK driver initialized: 16 RAM disks of 10240K size 1024 blocksize
loop: module loaded
Loading Marvell Ethernet Driver:
  o Cached descriptors in DRAM
  o DRAM SW cache-coherency
  o Single RX Queue support - ETH_DEF_RXQ=0
  o Single TX Queue support - ETH_DEF_TXQ=0
```

```
  o TCP segmentation offload enabled
  o LRO support supported
  o Receive checksum offload enabled
  o Transmit checksum offload enabled
  o Network Fast Processing (Routing) supported
  o Driver ERROR statistics enabled
  o Driver INFO statistics enabled
  o Proc tool API enabled
  o SKB Reuse supported
  o SKB Recycle supported
  o Rx descripors: q0=128
  o Tx descripors: q0=532
  o Loading network interface(s):
    o  register under egiga0 platform
    o egiga0, ifindex = 1, GbE port = 0

Warning: Giga 1 is Powered Off

mvFpRuleDb (c7dd0000): 1024 entries, 4096 bytes
Integrated Sata device found
scsi0 : Marvell SCSI to SATA adapter
scsi1 : Marvell SCSI to SATA adapter
scsi 0:0:0:0: Direct-Access     SAMSUNG  HD502IJ          1AA0 PQ: 0 ANSI: 5
scsi 1:0:0:0: Direct-Access     SAMSUNG  HD502IJ          1AA0 PQ: 0 ANSI: 5
scsi 0:0:0:0: Attached scsi generic sg0 type 0
scsi 1:0:0:0: Attached scsi generic sg1 type 0
NFTL driver: nftlcore.c $Revision: 1.1.1.1 $, nftlmount.c $Revision: 1.1.1.1 $
NAND device: Manufacturer ID: 0xec, Chip ID: 0xf1 (Samsung NAND 128MiB 3,3V 8-bit)
Scanning device for bad blocks
Bad eraseblock 98 at 0x00c40000
Using static partition definition
Creating 6 MTD partitions on "nand_mtd":
0x00000000-0x00100000 : "u-boot"
0x00100000-0x00600000 : "uImage"
0x00600000-0x00b00000 : "ramdisk"
0x00b00000-0x07100000 : "image"
0x07100000-0x07b00000 : "mini firmware"
0x07b00000-0x08000000 : "config"
ehci_marvell ehci_marvell.70059: Marvell Orion EHCI
ehci_marvell ehci_marvell.70059: new USB bus registered, assigned bus number 1
ehci_marvell ehci_marvell.70059: irq 19, io base 0xf1050100
ehci_marvell ehci_marvell.70059: USB 2.0 started, EHCI 1.00, driver 10 Dec 2004
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 1 port detected
USB Universal Host Controller Interface driver v3.0
mice: PS/2 mouse device common for all mice
i2c /dev entries driver
md: linear personality registered for level -1
md: raid0 personality registered for level 0
md: raid1 personality registered for level 1
device-mapper: ioctl: 4.11.0-ioctl (2006-10-12) initialised: dm-devel@redhat.com
dm_crypt using the OCF package.
usbcore: registered new interface driver usbhid
drivers/hid/usbhid/hid-core.c: v2.6:USB HID core driver
TCP cubic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
md: Autodetecting RAID arrays.
md: autorun ...
md: ... autorun DONE.
RAMDISK: Compressed image found at block 0
VFS: Mounted root (ext2 filesystem).
Freeing init memory: 124K
init started: BusyBox v1.11.2 (2010-11-17 11:44:30 CST)
[...]
```

Setting up ffp

There is a complete Tutorial on the Installation of ffp available on NAS-Tweaks: Installation of fun_plug 0.5. If you want to do everything yourself instead, just follow the instructions below.

To get the fonz fun_plug running some small changes to the available ffp `fun_plug` file are needed.

Search and replace all entries of `/mnt/HD_a2/` with `/mnt/HD/HD_a2/`.

Then follow the normal instructions of how to install ffp.

shadowandy has provided modified ffp package for DNS-320 here

Another article in German on installing fun_plug 0.5 can be found here

Once installed and running you might want to set-up ssh and change the root password. (for example this)

But to need to change `store-passwd.sh` also because that is functioning differently on our device. It can be found at `/ffp/sbin/store-passwd.sh`

And make it look like this:

```
#!/bin/sh

echo "Copying files to /usr/local/config/..."
cp -f /etc/passwd /usr/local/config/.
cp -f /etc/group /usr/local/config/.
cp -f /etc/shadow /usr/local/config/.
cp -f /etc/samba/smbpasswd /usr/local/config/.
```

Installation and Configuration of Transmission with addons

Contributed by Zane Chua on 10th June 2011.

First of all, you need Funplug 0.5 above. SSH into the NAS by using PuTTY. Installation of Transmission:

```
wget http://kylek.is-a-geek.org:31337/files/curl-7.18.1.tgz
wget http://kylek.is-a-geek.org:31337/files/Transmission-2.42-1.tgz
funpkg -i curl-7.18.1.tgz
funpkg -i Transmission-2.42-1.tgz
```

Once installed, you need to edit the transmission.sh in /ffp/start SSH into the NAS by using PuTTY again. Type these commands:

```
wget http://ffp.wolf-u.li/additional/app-editors/nano-2.0.9-2.tgz
funpkg -i nano-2.0.9-2.tgz
cd /ffp/start/
nano transmission.sh
```

Edit the line from

```
TRANSMISSION_HOME=/mnt/HD_a2/.transmission-daemon
```

to

```
TRANSMISSION_HOME=/mnt/HD/HD_a2/.transmission-daemon
```

Press Ctrl + O and Ctrl + X

After that, i believe that you would want to access the Transmission Web Interface.
SSH into the NAS by using PuTTY again.

```
/ffp/start/transmission.sh stop
su nobody -c "transmission-daemon -f -g /mnt/HD/HD_a2/.transmission-daemon -w /mnt/HD/HD_a2/Downloads
```

I downloaded Transmission Torrent Client Utilities v0.2 from http://dns323.thecorewithin.net/
Run this code via SSH again

```
wget http://dns323.thecorewithin.net/trans_utils-0.2.tgz
mv trans_utils-0.2.tgz /mnt/HD/HD_a2/trans_utils-0.2.tgz
cd /mnt/HD/HD_a2/
tar xvzf trans_utils-0.2.tgz
cd trans_utils-0.2
nano install.sh
```

Change the line from

```
BASE_SHARE=/mnt/HD_a2
```

to

```
BASE_SHARE=/mnt/HD/HD_a2
```

Press Ctrl + O and Ctrl + X

Continue with the code:

```
cd scripts
nano torrent_utils.sh
```

Change the line from

```
BASE_SHARE=/mnt/HD_a2
```

to

```
BASE_SHARE=/mnt/HD/HD_a2
```

and the line from

```
TRANSMISSION_HOME=/mnt/HD_a2/.transmission-daemon
```

to

```
TRANSMISSION_HOME=/mnt/HD/HD_a2/.transmission-daemon
```

and the line from

```
WATCH_DIR=$BASE_SHARE/watch
```

to

```
WATCH_DIR=$BASE_SHARE/Downloads/watch
```

and the line from

```
COMPLETE_DIR=$BASE_SHARE/complete
```

to

```
COMPLETE_DIR=$BASE_SHARE/Downloads
```

Search for the line

```
mv "$COMPLETE_DIR/${SEEDING_FILENAME}.moving" "$COMPLETE_DIR/${SEEDING_FILENAME}"
```

Add this line below it

```
chown -R *USERNAME* "$COMPLETE_DIR/${SEEDING_FILENAME}"
```

Replace *USERNAME* with a User you created in the Web Configuration so anyone can edit the files
Press Ctrl + O and Ctrl + X

Continue with the code:

```
cd ..
cd start
nano trans_utils.sh
```

Change the line from

```
TRAN_UTILS=/mnt/HD_a2/.transmission-daemon/scripts
```

to

```
TRAN_UTILS=/mnt/HD/HD_a2/.transmission-daemon/scripts
```

Continue with the code:

```
cd ..
./install.sh
```

It shouldn't have any issues running but it should make a watch and complete folder in /mnt/HD/HD_a2 which

is not even used so we shall just remove them and create them in the proper directories
Continue with the code:

```
cd ..
rm -rf watch
rm -rf complete
cd Downloads
mkdir watch
cd ..
/ffp/start/transmission.sh stop   (wait a while after this)
chown -R nobody /mnt/HD/HD_a2/.transmission-daemon
chown -R nobody /mnt/HD/HD_a2/Downloads
```

Extras
If you're a paranoid freak like me and you want to be able to remote access your DNS-320 not from your home and want to use a different port.
You have to change the line in torrent_utils.sh from

```
TRANS_REMOTE=$BASE_SYS/bin/transmission-remote
```

to

```
TRANS_REMOTE="$BASE_SYS/bin/transmission-remote Port"
```

and then if you're intending to use authentication by username and password you have to add the username and password in the respective fields in the Settings.json File located in /mnt/HD/HD_a2/.transmission-daemon.
Then you have to change the line in torrent_utils.sh from

```
TRANS_REMOTE=$BASE_SYS/bin/transmission-remote
```

to

```
TRANS_REMOTE="$BASE_SYS/bin/transmission-remote port --auth username:password"
```

That should work. Theoretically. If something is wrong with my commands, can someone please edit.
Configuration of Automatic.
I just recently decided to test out Automatic again, i had issues the last time but it seems everything works perfectly now so i'm including this here.
SSH into your NAS via PuTTY and make sure you're in /mnt/HD/HD_a2

```
wget http://kylek.is-a-geek.org:31337/files/Automatic-0.7.1-1.tgz
wget http://www.inreto.de/dns323/fun-plug/0.5/packages/libxml2-2.6.31-2.tgz
wget http://www.inreto.de/dns323/fun-plug/0.5/packages/pcre-7.7-1.tgz
funpkg -i libxml2-2.6.31-2.tgz
funpkg -i pcre-7.7-1.tgz
funpkg -i Automatic-0.7.1-1.tgz
```

You then need to edit the automatic.sh located in /ffp/start.

```
cd /ffp/start/
nano automatic.sh
```

Edit the line from

```
automatic_flags="-c /ffp/etc/automatic.conf -l /mnt/HD_a2/.transmission-daemon/automatic.log"
```

to

```
automatic_flags="-c /ffp/etc/automatic.conf -l /mnt/HD/HD_a2/.transmission-daemon/automatic.log -v 2"
```

Press Ctrl + O and Ctrl + X

Now you need to create a automatic configuration file. We'll use the sample file.

```
cd /ffp/etc/
nano automatic.conf-sample
```

Remove the # from rpc-host and rpc-port.
If you need #rpc-auth means you've configured a username and password for Transmission.
Change the line from

```
torrent-folder = "/tmp"
```

to

```
torrent-folder = "/mnt/HD/HD_a2/.transmission-daemon/autotorrents"
```

and the line from

```
statefile = "~/.config/automatic.state"
```

to

```
statefile = "/mnt/HD/HD_a2/.transmission-daemon/automatic.state"
```

Edit in your RSS Feeds and filters
Press Ctrl + O when you're done and save it as automatic.conf then Press Ctrl + X
Now we need to go create the folders and state file for automatic. Fire up PuTTY again and do these commands:

```
cd /mnt/HD/HD_a2/.transmission-daemon
mkdir autotorrents
touch automatic.state
chmod 777 automatic.state
```

Now you can start automatic.sh by typing

```
/ffp/start/automatic.sh start
```

If you want to stop automatic

```
/ffp/start/automatic.sh stop
```

Just some tips for the Filters. My RSS Feeds are basically sorted out so I used this section for filters

```
filter = {  pattern => "(?!.*.)"
         }
```

It basically means that it applies to all RSS Feeds.
All my commands should be correct, someone edit em if they're wrong. Have fun!

Changing the port of Web Configuration Utility

Contributed by Anand Kothapeta on 5/20/2011.

I was running two NAS servers (one D-Link DNS-320 and my old PC with FreeNAS) and a web server on a different PC. These are all behind my router. My router and FreeNAS allows me to change the default port for the web configuration utilities. So, I can change them from port 80 and port forward them, and can access the web configuration utilities from internet. However I always have an issue with DNS-320, since it doesn't allow me to change the port and it conflicts with web server. There are some articles on how to change it by running lighttpd. However, I found out that DNS-320 also uses the lighttpd to run the webconfig utility. Also those articles change the port to 81 (not much flexibility). I finally figured out a way to change the port to whatever I wish it to be. Here are the steps. You need to first install the ffp using above articles and links.

1) telnet/ssh login to the DNS-320 box. (see ffp tutorials) and run the following command

```
cp /etc/lighttpd/lighttpd.conf /ffp/lighttpd_portmod.conf
```

2) Now edit the /ffp/lighttpd_portmod.conf file and change the following line to lets say 8888 and save the file.

```
server.port = 8888
```

3) edit the fun_plug script and add the following lines between fun_plug.local and FFP_RC if conditions.

```
# run fun_plug.local, if present
if [ -x /ffp/etc/fun_plug.local ]; then
    echo "* Running /ffp/etc/fun_plug.local ..."
    /ffp/etc/fun_plug.local
fi

#*** start Mods for different http port for webadmin tool
echo "killing  lighttpd-angel"
kill -9 `pidof lighttpd-angel`
echo "killing  lighttpd"
```

```
kill -9 `pidof lighttpd`

echo "Restart  lighttpd-angel with different port"
/usr/sbin/lighttpd-angel -D -m /usr/local/lib -f /ffp/lighttpd_portmod.conf &
#*** end Mods different http port

# run commands
if [ -x $FFP_RC ]; then
    echo "* Running $FFP_RC ..."
    $FFP_RC
    echo "*  OK"
else
    echo "$FFP_RC: Not found or not executable"
fi
```

4) Now reboot it.

5) Look at ffp.log to make sure you see the following lines.

```
killing  lighttpd-angel
killing  lighttpd
Restart  lighttpd-angel with different port
```

6) Now open the browser and try http://<ipaddress>:8888/ should take you to the web config tool.

7) Now you can port forward 8888 in your router, and port forward 80 to web server.

Setting up Debian in chrooted enviroment

D-Link has recently put a prebuilt Debian Squeeze package on their ftp site. The package size is approximately 150 MB. Read the instruction inside. You may have to modify `—bind` (dash-dash-bind) to `—rbind` (dash-dash-rbind) for the `/proc` (and possibly `/dev`) mounts in fun_plug.debian to get some usb devices (e.g. the DS9490R USB-1wire adapter) to work properly in squeeze.

the prebuilt Debian Squeeze package is NO LONGER available (anywhere?)as of july 2014, with no alternative build options ,compiling new apps are left to only the semi pro devs that can create their own dev systems.

Source code for firmware

Dlink have release source code for firmware 1.0 on their ftp site. The source code size is approximate 25 MB

U-BOOT Configuration

You can access the boot console (for trying to run your own kernel) via a netconsole. To interrupt the U-Boot startup, press SPACE and 1 before the one-second timeout. You should get a Marvell_DNS320» prompt immediately. If the kernel starts to boot, you missed the timeout window. Starting ahead of the timeout and pressing SPACE-1-SPACE-1, etc. will increase you chances of hitting the window.

The full U-Boot environment (MAC addresses hidden) is:

```
Marvell_DNS320>> printenv
```

```
bootargs=root=/dev/ram console=ttyS0,115200 :::DB88FXX81:egiga0:none
bootcmd=nand read.e 0xa00000 0x100000 0x300000;nand read.e 0xf00000 0x600000 0x300000;bootm 0xa00000
baudrate=115200
loads_echo=0
ipaddr=2.76.88.201
serverip=2.76.88.2
rootpath=/mnt/ARM_FS/
netmask=255.255.255.0
stdin=serial
stdout=serial
stderr=serial
console=console=ttyS0,115200 mtdparts=nand_mtd:0xc0000@0(uboot)ro,0x7f00000@0x100000(root)
mainlineLinux=no
CASset=min
enaMonExt=no
enaCpuStream=no
enaWrAllo=no
pexMode=RC
disL2Cache=no
setL2CacheWT=yes
disL2Prefetch=yes
enaICPref=yes
enaDCPref=yes
sata_dma_mode=yes
MALLOC_len=1
ethprime=egiga0
netbsd_en=no
vxworks_en=no
bootargs_root=root=/dev/nfs rw
bootargs_end=:::DB88FXX81:eth0:none
image_name=uImage
standalone=fsload 0x2000000 $(image_name);setenv bootargs $(console) root=/dev/mtdblock0 rw ip=$(ipad
bootdelay=1
disaMvPnp=no
ethaddr=00:50:XX:XX:XX:XX
ethmtu=1500
mvPhoneConfig=mv_phone_config=dev0:fxs,dev1:fxs
mvNetConfig=mv_net_config=(00:11:XX:XX:XX:XX,0:1:2:3),mtu=1500
usb0Mode=host
yuk_ethaddr=00:00:00:XX:XX:XX
nandEcc=1bit
netretry=no
rcvrip=169.254.100.100
loadaddr=0x02000000
autoload=no
enaAutoRecovery=yes
ethact=egiga0

Environment size: 1238/131068 bytes
```

Building a Custom Firmware ("ROM")

D-Link provides a complete build environment on its ftp server that lets you create your own firmware images easily. Untar it and run

```
source build_fw
```

this will result in a file “DNS320-firmware” in the merge folder, which can be flashed to the unit via the webinterface (firmware update). Right now the build_fw script does not do a full rebuild of the complete firmware, it rather puts together precompiled binaries that come with the built environment. Here is an attempt at rebuilding the build script. This blog post describes how to assemble a firmware image for the DNS-323 and will probably helpful for the DNS-320 as well.

Use this custom build_fw to make a custom settings for the kernel

```
# Dlink build envorment it only works on x86 arch (32Bits)
```

```
if [ ! -d "arm-none-linux-gnueabi-6xxx" ]; then
   # directory did not exists unpack it
   tar zxvf arm-none-linux-gnueabi-6xxx.tgz
fi

WORKDIR=$(pwd)
export WORKDIR
PATH=${WORKDIR}/arm-none-linux-gnueabi-6xxx/bin:$PATH
export PATH
CC=arm-none-linux-gnueabi-gcc
export CC
export CROSS_COMPILE=arm-none-linux-gnueabi-
CXX=${CROSS_COMPILE}g++
export CXX
RANLIB=arm-none-linux-gnueabi-ranlib
export RANLIB
LD=arm-none-linux-gnueabi-ld
export LD
STRIP=arm-none-linux-gnueabi-strip
export STRIP

# gl setup path to mkimage
if [ ! -d "ramdisk" ]; then
  tar -xzf ramdisk.tgz
fi
PATH=${WORKDIR}/ramdisk:$PATH

# gl extract kernel
if [ ! -d "linux-2.5.22.18" ]; then
  tar -xzf linux-2.6.22.18.tgz
fi

# config our kernel and build it
cd ./linux-2.6.22.18
sh cv.sh
make clean
make menuconfig
make
make modules
make uImage
cd ../

# Dlink run merge now
if [ ! -d "merge" ]; then
 tar zxvf merge.tgz
fi

if [ ! -f "merge/uImage.org" ]; then
  mv merge/uImage merge/uImage.org
fi

if [ -f "merge/uImage" ]; then
 rm merge/uImage
fi

cp linux-2.6.22.18/arch/arm/boot/uImage merge/uImage
cd merge
sudo cp ../ramdisk/uRamdisk .
./merge
```

This Program works simlure as Dlink 320 merge does. It have been create by Magnus Olsen and release under GPL 2.0 The header for firmware are lite diffent againts DNS-323 to use it as merge u need compile it gcc dns320flash.c -o dns320flash copy where merge is and type dns320flash -mf myDNS320firmware now u will have a firmware call myDNS320firmware

```


#include <stdio.h>
#include <stdlib.h>

#define UINT32 unsigned int
#define BYTE unsigned char
#define ALGIN4(_x) (_x + (4 - (_x % 4)))
typedef struct {
```

```
   UINT32 uImageOffset;
   UINT32 uImageLenght;
   UINT32 uRamDiskOffset;
   UINT32 uRamDiskLenght;
   UINT32 ImageOffset;
   UINT32 ImageLenght;
   UINT32 DefaultOffset;
   UINT32 DefaultLenght;
   UINT32 uImageChecksum;
   UINT32 uRamDiskChecksum;
   UINT32 ImageChecksum;
   UINT32 DefaultChecksum;
   BYTE magic_0;
   BYTE magic_1;
   BYTE strModelName[9];
   BYTE magic_2;
   BYTE productId;
   BYTE customId;
   BYTE modelId;
   BYTE hardwareId;
   BYTE subId;
   BYTE Reseverd[59];
   UINT32 NextOffset;
} DLINK_DNS320_FIRMWARE_HEAD_128BYTE;

void showFirmware(char *flashfilename, int extract_frimware, char *kernelfilename, char *ramdiskfilen
int pack_firmware(char *flashfilename, char *kernelfilename, char *ramdiskfilename, char *imagefilena

void firmware_unpack_extract(char *inFileName, char *outFileName, UINT32 inFileOffset, UINT32 dataLen
UINT32 firmware_calc_checksum(char *inFileName);
int calc_checksum(char *filename);
int check_size(char *filename, UINT32 *file_size, UINT32 max_size);
char * strtolower(const char *ret);
int compare(char *filename1, char *filename2);


char *strtolower(const char *ret)
{
    char *s = ret;
    for ( s = ret;*s != 0; s++)
    {
        if (*s != 0)
        { *s = tolower(*s); }
    }
    return ret;
 }

int main(int argc, char **argv)
{
    int showHelp = 1;

    if (argc == 3)
    {
        strtolower(argv[1]);
        if ( strcmp("-s",argv[1]) == 0)
        { showFirmware(argv[2], 0, "", "", "", "" ); showHelp--; }

        if ( strcmp("-x",argv[1]) == 0)
        { showFirmware(argv[2], 1, "uImage", "uRamDisk", "image.cfs", "default.tar.gz" ); showHelp--;

        if ( strcmp("-mf",argv[1]) == 0)
        { pack_firmware(argv[2],"uImage", "uRamDisk", "image.cfs", "default.tar.gz"); showHelp--; }

    }

    if (argc == 4)
    {
        strtolower(argv[1]);
        if ( strcmp("-cmp",argv[1]) == 0)
        { compare(argv[2], argv[3]); showHelp--; }
    }

    if (argc == 8)
    {
        strtolower(argv[1]);
        if ( strcmp("-m",argv[1]) == 0)
        {
            strtolower(argv[6]);
```

```
            if ( strcmp("-f",argv[6]) == 0)
            { pack_firmware(argv[7],argv[2], argv[3], argv[4], argv[5]); showHelp--; }
        }

        if ( strcmp("-x",argv[1]) == 0)
        {
            strtolower(argv[3]);
            if ( strcmp("-o",argv[3]) == 0)
            { showFirmware(argv[2], 1, argv[4], argv[5], argv[6], argv[7] ); showHelp--; }
        }
    }

    if (showHelp)
    {
        printf("Copyright 2011 by Magnus Olsen (magnus@greatlord.com)\n");
        printf("This program are under licen GPL 2.0\n\n");
        printf("Merge kernel (uImage) + ramdisk (uRamDisk) + image (image.cfs) and\nconfig (default.t
        printf("dns320flash -m kernel ramdisk image config -f dlink320firmware\n\n\n");
        printf("dns320flash -mf dlink320firmware dlink320firmware\n\n\n");
        printf("Extract flash image to kernel (uImage) + ramdisk (uRamDisk) + \nimage (image.cfs) and
        printf("dns320flash -x dlink320firmware -o kernel ramdisk image config\n");
        printf("dns320flash -x dlink320firmware \n\n\n");
        printf("Show flash image contains for dlink DNS320\n\n");
        printf("dns320flash -s dlink320firmware\n\n\n");
    }

    return 0;
}

void showFirmware(char *flashfilename, int extract_frimware, char *kernelfilename, char *ramdiskfilen
{
    int read_byte;
    FILE *fp;
    char str[10];
    DLINK_DNS320_FIRMWARE_HEAD_128BYTE dnsHeader;


    fp = fopen(flashfilename,"rb");
    if (fp)
    {
        memset( &dnsHeader,0,sizeof(DLINK_DNS320_FIRMWARE_HEAD_128BYTE));
        read_byte = fread(&dnsHeader, 1, sizeof(DLINK_DNS320_FIRMWARE_HEAD_128BYTE),fp);
        fclose(fp);

        if (read_byte != sizeof(DLINK_DNS320_FIRMWARE_HEAD_128BYTE))
        {
            printf("not vaild firmware for DNS320");
            exit(0);
        }

        memcpy(&str,dnsHeader.strModelName,9);
        str[9] = 0;

        if  ( (dnsHeader.magic_0 != 0x55) ||
              (dnsHeader.magic_1 != 0xAA) ||
              (dnsHeader.magic_2 != 0xAA) ||
              (dnsHeader.productId != 0x00) ||
              (dnsHeader.customId != 0x08) ||
              (dnsHeader.modelId != 0x07) ||
              (dnsHeader.hardwareId != 0x01) ||
              (dnsHeader.subId != 0x00) ||
              (strcmp(str,"DNS323D1U") != 0) )
        {
            printf("not vaild firmware for DNS320");
            exit(0);
        }

        printf("magic_0     : %02x\n",dnsHeader.magic_0);
        printf("magic_1     : %02x\n",dnsHeader.magic_1);
        printf("string      : %s\n",str);
        printf("magic_2     : %02x\n",dnsHeader.magic_2);
        printf("productId   : %02x\n",dnsHeader.productId);
        printf("customId    : %02x\n",dnsHeader.customId);
        printf("modelId     : %02x\n",dnsHeader.modelId);
        printf("hardwareId  : %02x\n",dnsHeader.hardwareId);
        printf("subId       : %02x\n",dnsHeader.subId);
        printf("Next_offset : %08x\n\n",dnsHeader.NextOffset);
```

```
        printf("Kernel (uImage) Offset   head : %08x\n",dnsHeader.uImageOffset);
        printf("Kernel (uImage) Lenght   head : %08x\n",dnsHeader.uImageLenght);
        printf("Kernel (uImage) Checksum head : %08x\n",dnsHeader.uImageChecksum);


        if (extract_frimware)
        {
            firmware_unpack_extract(flashfilename, kernelfilename, dnsHeader.uImageOffset, dnsHeader.
            printf("Kernel (uImage) Checksum file : %08x\n\n",calc_checksum(kernelfilename));
        }
        else
        { printf("\n"); }

        printf("uRamDisk        Offset   head : %08x\n",dnsHeader.uRamDiskOffset);
        printf("uRamDisk        Lenght   head : %08x\n",dnsHeader.uRamDiskLenght);
        printf("uRamDisk        Checksum head : %08x\n",dnsHeader.uRamDiskChecksum);
        if (extract_frimware)
        {

            firmware_unpack_extract(flashfilename, ramdiskfilename, dnsHeader.uRamDiskOffset, dnsHead
            printf("uRamDisk        Checksum file : %08x\n\n",calc_checksum(ramdiskfilename));
        }
        else
        { printf("\n"); }


        printf("image           Offset   head : %08x\n",dnsHeader.ImageOffset);
        printf("image           Lenght   head : %08x\n",dnsHeader.ImageLenght);
        printf("image           Checksum head : %08x\n",dnsHeader.ImageChecksum);

        if (extract_frimware)
        {
            firmware_unpack_extract(flashfilename, imagefilename, dnsHeader.ImageOffset, dnsHeader.Im
            printf("image           Checksum file : %08x\n\n",calc_checksum(imagefilename));
        }
        else
        { printf("\n"); }

        printf("Default         Offset   head : %08x\n",dnsHeader.DefaultOffset);
        printf("Default         Lenght   head : %08x\n",dnsHeader.DefaultLenght);
        printf("Default         Checksum head : %08x\n",dnsHeader.DefaultChecksum);


        if (extract_frimware)
        {
            firmware_unpack_extract(flashfilename, configfilename, dnsHeader.DefaultOffset, dnsHeader
            printf("Default         Checksum file : %08x\n\n",calc_checksum(configfilename));
        }
        else
        { printf("\n"); }




    }

}
void firmware_unpack_extract(char *inFileName, char *outFileName, UINT32 inFileOffset, UINT32 dataLen
{
    FILE *read_fp;
    FILE *save_fp;
    BYTE *ch;
    UINT32 lenght = 0;
    printf("Extracting %s from %s\n",outFileName, inFileName);
    read_fp = fopen(inFileName,"rb");
    if (read_fp)
    {
        fseek(read_fp,inFileOffset, SEEK_SET);
        save_fp = fopen(outFileName,"wb+");
        if (save_fp)
        {
            while(lenght<dataLenght)
            {
                fread(&ch,1,1,read_fp);
                fwrite(&ch,1,1,save_fp);
                lenght++;
            }
```

```
            printf("writing done\n");
            fclose(save_fp);
        }
        fclose(read_fp);
    }
}

int pack_firmware(char *flashfilename, char *kernelfilename, char *ramdiskfilename, char *imagefilena
{
    FILE *fp_read;
    FILE *fp_save;
    int i, kernelSize,ramdiskSize, imageSize, configSize;
    UINT32 byte;
    DLINK_DNS320_FIRMWARE_HEAD_128BYTE dnsHeader;

    memset(&dnsHeader,0,sizeof(DLINK_DNS320_FIRMWARE_HEAD_128BYTE));


    if ( check_size(kernelfilename,         &dnsHeader.uImageLenght,    0x00300000) == -1)
    { return -1; }

    if ( check_size(ramdiskfilename,       &dnsHeader.uRamDiskLenght,  0x00300000) == -1)
    { return -1; }

    if ( check_size(configfilename, &dnsHeader.DefaultLenght,   0x00300000) == -1)
    { return -1; }

    if ( check_size(imagefilename,      &dnsHeader.ImageLenght,     0x04000000) == -1)
    { return -1; }


    dnsHeader.magic_0 = 0x55;
    dnsHeader.magic_1 = 0xAA;
    dnsHeader.magic_2 = 0xAA;
    dnsHeader.productId = 0x00;
    dnsHeader.customId = 0x08;
    dnsHeader.modelId = 0x07;
    dnsHeader.hardwareId = 0x01;
    dnsHeader.subId = 0x00;
    memcpy(dnsHeader.strModelName,"DNS323D1U",9);


    dnsHeader.uImageChecksum = calc_checksum(kernelfilename);
    dnsHeader.uRamDiskChecksum = calc_checksum(ramdiskfilename);
    dnsHeader.DefaultChecksum = calc_checksum(configfilename);
    dnsHeader.ImageChecksum = calc_checksum(imagefilename);


    kernelSize = dnsHeader.uImageLenght;
    ramdiskSize = dnsHeader.uRamDiskLenght;
    imageSize = dnsHeader.ImageLenght;
    configSize = dnsHeader.DefaultLenght;

    if ((dnsHeader.uImageLenght % 4) != 0)
    { dnsHeader.uImageLenght = ALGIN4(dnsHeader.uImageLenght); }

    if ((dnsHeader.uRamDiskLenght % 4) != 0)
    { dnsHeader.uRamDiskLenght = ALGIN4(dnsHeader.uRamDiskLenght); }

    if ((dnsHeader.ImageLenght % 4) != 0)
    { dnsHeader.ImageLenght = ALGIN4(dnsHeader.ImageLenght); }

    if ((dnsHeader.DefaultLenght % 4) != 0)
    { dnsHeader.DefaultLenght = ALGIN4(dnsHeader.DefaultLenght); }


    dnsHeader.uImageOffset = sizeof(DLINK_DNS320_FIRMWARE_HEAD_128BYTE);
    dnsHeader.uRamDiskOffset = sizeof(DLINK_DNS320_FIRMWARE_HEAD_128BYTE) + dnsHeader.uImageLenght;
    dnsHeader.ImageOffset = sizeof(DLINK_DNS320_FIRMWARE_HEAD_128BYTE) + dnsHeader.uImageLenght + dns

    dnsHeader.DefaultOffset = sizeof(DLINK_DNS320_FIRMWARE_HEAD_128BYTE) + dnsHeader.uImageLenght + d

    printf("Kernel (uImage) Offset    : %08x\n",dnsHeader.uImageOffset);
    printf("Kernel (uImage) Lenght    : %08x\n",dnsHeader.uImageLenght);
    printf("Kernel (uImage) Checksum  : %08x\n\n",dnsHeader.uImageChecksum);
    printf("uRamDisk        Offset    : %08x\n",dnsHeader.uRamDiskOffset);
    printf("uRamDisk        Lenght    : %08x\n",dnsHeader.uRamDiskLenght);
    printf("uRamDisk        Checksum  : %08x\n\n",dnsHeader.uRamDiskChecksum);
```

```
    printf("image           Offset    : %08x\n",dnsHeader.ImageOffset);
    printf("image           Lenght    : %08x\n",dnsHeader.ImageLenght);
    printf("image           Checksum  : %08x\n\n",dnsHeader.ImageChecksum);
    printf("Default         Offset    : %08x\n",dnsHeader.DefaultOffset);
    printf("Default         Lenght    : %08x\n",dnsHeader.DefaultLenght);
    printf("Default         Checksum  : %08x\n\n",dnsHeader.DefaultChecksum);

    if ( (fp_save = fopen(flashfilename,"wb+")) == 0)
    {
        printf("Error : Can not merge firmware\n");
        return -1;
    }

    fwrite(&dnsHeader,1,sizeof(DLINK_DNS320_FIRMWARE_HEAD_128BYTE),fp_save);

    if ( (fp_read = fopen(kernelfilename,"rb")) == 0)
    {
        printf("Error : Can not merge firmware\n");
        return -1;
    }
    for(i=0;i<dnsHeader.uImageLenght/4;i++)
    {
        fread(&byte,1,4,fp_read);
        fwrite(&byte,1,4,fp_save);
    }
    fclose(fp_read);

    if ( (fp_read = fopen(ramdiskfilename,"rb")) == 0)
    {
        printf("Error : Can not merge firmware\n");
        return -1;
    }
    for(i=0;i<dnsHeader.uRamDiskLenght/4;i++)
    {
        fread(&byte,1,4,fp_read);
        fwrite(&byte,1,4,fp_save);
    }
    fclose(fp_read);

    if ( (fp_read = fopen(imagefilename,"rb")) == 0)
    {
        printf("Error : Can not merge firmware\n");
        return -1;
    }
    for(i=0;i<dnsHeader.ImageLenght/4;i++)
    {
        fread(&byte,1,4,fp_read);
        fwrite(&byte,1,4,fp_save);
    }

    fclose(fp_read);

    if ( (fp_read = fopen(configfilename,"rb")) == 0)
    {
        printf("Error : Can not merge firmware\n");
        return -1;
    }

    for(i=0;i<dnsHeader.DefaultLenght/4;i++)
    {
        fread(&byte,1,4,fp_read);
        fwrite(&byte,1,4,fp_save);
    }
    fclose(fp_read);


    fclose(fp_save);

    return 0;
}


int check_size(char *filename, UINT32 *file_size, UINT32 max_size)
{
    FILE *fp_read;

    if ((fp_read = fopen(filename,"rb")) )
    {
```

```
        fseek(fp_read,0,SEEK_END);
        *file_size = ftell(fp_read);
        if ( *file_size >= max_size )
        {
            printf("\nMake firmware error\n");
            printf("Ramdisk sizes can not more than %d byte",max_size);
            return -1;
        }
        fclose(fp_read);
    }
    else
    {
        printf("Can not open file uImage\n");
    }
    return 0;
}



int calc_checksum(char *filename)
{
    FILE *fp_read;
    UINT32 chk_sum = 0;
    int byte,size,filesize;

    if (!(fp_read = fopen(filename,"rb")))
    {
        printf("Error: Can't open %s file !\n", filename);
        return 0;
    }

    fseek(fp_read, 0, SEEK_END);
    filesize = ftell(fp_read);
    fseek(fp_read, 0, SEEK_SET);

    for(size=0;size<filesize;size+=4)
    {
        byte = 0;
        fread(&byte, 1, 4, fp_read);
        chk_sum ^= byte;
    }

    fclose(fp_read);
    return chk_sum;
}

int compare(char *filename1, char *filename2)
{
    FILE *fp_read1;
    FILE *fp_read2;
    int file1_size;
    int file2_size;
    int i;

    if ((fp_read1 = fopen(filename1,"rb")) == 0)
    { printf("Error : Can not open file %s\n",filename1); return -1; }

    if ((fp_read2 = fopen(filename2,"rb")) == 0)
    { printf("Error : Can not open file %s\n",filename2); fclose(fp_read1); return -1; }

    fseek(fp_read1,0,SEEK_END);
    fseek(fp_read2,0,SEEK_END);

    file1_size = ftell(fp_read1);
    file2_size = ftell(fp_read2);

    fseek(fp_read1,0,SEEK_SET);
    fseek(fp_read2,0,SEEK_SET);

    if (file1_size != file2_size)
    {
        printf("Error : File %s and file %s does not have same filesize\n",filename1,filename2);
        fclose(fp_read1);
        fclose(fp_read2);
        return -1;
    }
```

```
    for (i=0;i<file1_size;i++)
    {
        if ( fgetc(fp_read1) != fgetc(fp_read2) )
        {
            printf("Error : Byte are not same\n");
            printf("FileOffset : %08lx\n",ftell(fp_read1));
            fclose(fp_read1);
            fclose(fp_read2);
            return -1;
        }

    }
    fclose(fp_read1);
    fclose(fp_read2);
    return 0;
}
```

Other info

The following thread handles some DNS-320 initial conversation, request there if you want some specific info here, or add yourself.

Fan Speed settings

With ffp is enabled, login to the box via telnet and run FT_testing

ex:

```
/ # FT_testing -H
Fan and Temperature Testing Tools
Usage
FT_testing      -H      ...Help message
FT_testing      -T      ...Get Temperature
FT_testing      -F      ...Set Fan in Full Speed
FT_testing      -L      ...Set Fan in Low Speed
FT_testing      -S      ...Set Fan Stop
FT_testing      -A      ...Print Receive UART Packet.
FT_testing      -G      ...Get Fan Status.
FT_testing      -P      ...Power off system.
FT_testing      -R      ...Reboot system.
```

if you need the fan is always off, type:

```
ps uax |grep fan_control
```

and kill this process

or

```
killall fan_control
```

, and run

```
FT_testing -S
```

/code

SSH Access

This is just a quick n' dirty way to get root access to the device. This configuration is *not* recommended as permanent.

1. Create fun_plug
2. Create sshd_config
3. reboot dns-320

File 1 fun_plug *

```
#!/bin/sh
# credit to
# http://webapps.itcs.umich.edu/radmind/index.php/Generate_ssh_host_keys
# for the scripted ssh-keygen

ssh-keygen -q -t dsa -f /mnt/HD/HD_a2/ssh_host_dsa_key -N ""\
    -C "" < /dev/null > /dev/null 2> /dev/null

/usr/sbin/sshd -f /mnt/HD/HD_a2/sshd_config
```

File 2 sshd_config *

```
Port 22
Protocol 2
HostKey /mnt/HD/HD_a2/ssh_host_dsa_key
PermitEmptyPasswords yes
PasswordAuthentication yes
ChallengeResponseAuthentication no
TCPKeepAlive yes
```

* Create the files with a Linux compatible text editor (Like Notepad++)

Connect: `ssh root@dns_ip`

Detect USB button pressure

I wanted to replace the 8 seconds pressure on the power button by a simple pressure on the usb button. So, i've analysed the chk_io program while it seemed to be the one which does the job for the usb copy. The chk_io is not gpl, so only the binary is available. I used strace and LD_PRELOAD on ioctl to analyse it, and finally found a similar program for the dns-323.

I'm not sure it's required to open /dev/REG each time, but it's what the official program does. Idem for the open attributes.

I've created :

* ffp/extra/shutdown/chkbutton (gcc -static -s chkbutton.c -o chkbutton)

* /ffp/extra/shutdown/shutdown_ffp.sh

* /ffp/extra/shutdown/usbOff.sh

* /ffp/etc/fun_plug.local

Just copy these files or feel free to modify them.

Files are available here : http://nadenislamarre.free.fr/dns320

```
# chkbutton.c

#define _GNU_SOURCE 1
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

#define DEVICE "/dev/REG"
#define REQUEST 0x65
#define REGISTER 0x10110
#define BTN_NOTUSB 0x08000000

int main (int argc, char *argv[]) {
  int fd;
  unsigned long data[2];

  while(1) {
    if ((fd = open (DEVICE, O_RDWR|O_LARGEFILE)) == -1) {
      exit(1);
    }

    data[0] = REGISTER;
    data[1] = 0;

    if(ioctl(fd, REQUEST, &data) == 0) {
      if(! ((data[1] & BTN_NOTUSB) == BTN_NOTUSB)) {
        printf("USB\n");
        fflush(stdout);
        usleep(1000 * 1000 * 3);
      }
    }

    close(fd);
    usleep(1000*30);
  }

  return 0;
}
```

The following script called in the /ffp/etc/fun_plug.local make the dns-320 reboot just by pressing on the usb button. The redirection to /dev/null is required because the shutdown.sh script kill processes on some conditions.

```
#!/bin/sh

# /ffp/extra/shutdown/usbOff.sh

export PATH=$PATH:/bin:/sbin

chkbutton |
while read BTN
do
  if test "$BTN" = "USB"
  then
    led usb blue blinking
    shutdown_ffp.sh >/dev/null 2>/dev/null
    exit 0
  fi
done
```

```
#!/bin/sh

# /ffp/etc/fun_plug.local

# replacement for shutdown ; must not be on a mounted device
cp /ffp/extra/shutdown/* /sbin
usbOff.sh&
```

The shutdown script is the same as the official one except that i've added a proper shutdown for ffp : /ffp/etc/rc stop

```
#!/bin/sh

# /ffp/extra/shutdown/shutdown_ffp.sh

source /usr/local/modules/files/project_features

/ffp/etc/rc stop

kill_running_process
#kill -9 -1

# remove link file
rm /usr/local/upload
sync

umount_dev.sh all

umount /usr/local/config

sleep 1
# shutdown device

MODEL=$(cat "/usr/local/modules/files/model")
echo "MODEL:$MODEL"

if [ "$PROJECT_FEATURE_MCU_CHIP_WT693P" = "1" ]; then
        echo "device shutdown"
        up_send_ctl DeviceShutdown 0

elif [ "$PROJECT_FEATURE_MCU_CHIP_69P803" = "1" ]; then
        echo "send cmd to micro-p to shutdown"
        up_send_ctl DeviceShutdown 1
else
        poweroff
fi
```