

Collection of some hints about Debian installation on my D-Link DSM-G600:

0x87e3f28 [0, 0]

Contents: [Dobrica Pavlinu's random unstructured stuff]

- [Dobrica Pavlinu's random unstructured stuff \(Resources\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(firmware \(kernel+busybox\)\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(usbfs\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(USB printer\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(Kernel 2.4\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(Kernel 2.6\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(New round based on powerpc kernels\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(Compilation notes\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(Emulation\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(Source code\)](#)
- [Dobrica Pavlinu's random unstructured stuff \(Board specification\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(Memory map\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(MTD\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(Kernel configuration options\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(IRQ\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(mpc10x\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(IO block\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(Serial\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(CPU\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(Led control strings\)](#)
  - ◆ [Dobrica Pavlinu's random unstructured stuff \(Chassis Status Strings\)](#)

## Resources

- wiki at <http://dsmg600.info/>
- forum at <http://forum.dsmg600.info/>

## firmware (kernel+busybox)

It will fix various problems, including, but not limited to, problems with web interface in firefox.

<http://download.dsmg600.info/people/sala/fwimage-04-sala-20070128>

I tried to build firmware from source, but had various problems with building gcc. That are mostly related to gdb, but after a few random patches it passed compilation (but is broken, because I just commented offending lines).

## usbfs

I also want usbfs which generate compilation errors because of missing files in

`include/linux/usbdevice_fs.h`

Based on [instructions on wiki](#) and [post on forum](#)

`cd /mnt/HD_a2`

```
export DEBOOTSTRAP_DIR=`pwd`/usr/lib/debootstrap
export PATH=$PATH:/mnt/HD_a2/bin:/mnt/HD_a2/sbin
./usr/sbin/debootstrap --arch powerpc etch /mnt/HD_a2/debian http://ftp.de.debian.org/debian
cp /etc/resolv.conf /mnt/HD_a2/debian/etc/resolv.conf
# prepare chroot jail
mount -t proc proc /mnt/HD_a2/debian/proc
cd /mnt/HD_a2/
./chroot /mnt/HD_a2/debian /bin/bash
# install additional packages
apt-get update
apt-get install locales
dpkg-reconfigure locales
passwd root
apt-get install dropbear hotplug
```

## USB printer

Kernel module for USB printers for 2.4.21-pre4 kernel: [printer.o](#)

After installation of `cupsys` and `foomatic` all went smoothly, USB printer ([Samsung ML-2510](#)) was found and configured automatically. There are a few more details in [this forum thread](#).

## Kernel 2.4

As a try to understand this ppc board, I tried to port changes from MontaVista (D-Link used that tree) to current upstream 2.4 kernel and got to the point where SCSI controller times out when reading partition table. This means that board does init, it has serial port which works and has some idea about PCI space (which seems somewhat fishy if you ask me). [linux-2.4.34.4-dsmg600.diff.gz](#)

Few more details are available in [this forum post](#).

## Kernel 2.6

I have a crazy idea: this device **should** run 2.6! So, here is my work in progress...

According to [patches for kurobox](#) which is quite similar to our hardware, there isn't any changes in current upstream version except for missing drivers. For DSM G600 that boils down to [IP1000](#) network driver, which still isn't in mainland, but seems to be referenced on LKLM as possible inclusion.

- patch against upstream 2.6.21.1 [linux-2.6.21.1-dsmg600.diff.gz](#) (adds IP1000A driver for LAN, wifi missing)

Expirince with 2.4 kernel shows that I have to do more porting to make 2.6 port usable. As a start, code for board initialization is specific for DSM-G600 as well as serial port driver. When we have those two things working we can see if `loader.o` could boot our kernel. I think that shouldn't be problem, because `loader.o` origin shows that it's possible to boot 2.6 kernel from 2.4 kernel using `loader.o` module.

# New round based on powerpc kernels

After review of current upstream kernel (2.6.22) and state of [powerpc kernel.org repository](#) I decided to base my efforts on this branch.

For now here are few useful links:

- dtc compiler: <git://www.jdl.com/software/dtc.git>
- [Kurobox support in stock 2.6 kernels](#)

## Compilation notes

Compile kernel and convert it to binary format for [loader.o](#):

```
. env.sh
make
powerpc-linux-objcopy -O binary vmlinux
scp vmlinux disk:/tmp/
```

Then try out your kernel on DSM:

```
cd /mnt/HD_a2 && sync && insmod /mnt/HD_a2/loader.o kernel=/mnt/HD_a2/debian/tmp/vmlinux
```

Please note that **you must use full path** to kernel binary. Also note that all paths are customized for my particular device (to help with copy/paste :-)

## Emulation

It's useful to have development environment for DSM on another machine, so I tried to use [QEMU](#) to do so.

- patch which adds support for [platforms based on MPC82xx](#)
- [Installing Debian Sarge for the PowerPC under the QEMU](#) which I updated to etch to be in sync with DSM

This didn't work quite well as described first in [this blog post](#).

My efforts right now are into making [GXemu](#) emulate enough of DSM-G600 to boot [original D-Link kernel](#).

## Source code

All source code is now available [in git repository](#)

## Board specification

Here is information that I collected while working on 2.4 port in hope that it will assist me in porting

## 2.6 kernel

# Memory map

Part of information extracted from u-boot loader, part from kernel source

from	to	size	
00000000	02000000	02000000	SDRAM 32Mb
80000000	f0000000	70000000	pci mem space?
fc000000	fcc00000	00c00000	EUMB (PCI I/O space)
fcc00000	fcf00000	00300000	pci cfg regs
fcf00000	fd000000	00100000	pci iack
fe000000	febfffff	00c00000	PCI host bridge
ffc00000	ffffffff	00400000	Flash 4Mb

## Important addresses:

```
10000000      CFG_MAX_RAM_SIZE, CFG_BANK0_END (u-boot)
40000000      CFG_INIT_RAM_ADDR (u-boot)

c0000000      start of kernel 2.4.21-pre4 VM

80000000 --- PCI memory space ---
bffffd00      PCI 1033:0035 Non-prefetchable memory
bffffe00      PCI 1814:0201 Non-prefetchable memory
c3029f00      PCI 1033:00e0 00:0e.2 irq 2 ciehci_hcd
c3032000      PCI 1033:0035 00:0e.0 irq 1 usb-ohci
c3034000      PCI 1033:0035 00:0e.1 irq -1 usb-ohci disabled
c30ab000      PCI 1814:0201 irq 0 wirel, /sys/cra0
f0000000 --- PCI memory space ---

fc000000 --- EUMB ---
fc040000      OpenPIC_Addr (mpc1)
fc004500      ttyS0
fc004600      ttyS1
fd000000 --- EUMB ---

febffe00-febffe7f : PCI device 1191:0008
febffe00-febffe0f : atp86x
febffee4-febffee7 : PCI device 1191:0008
febffee8-febffeef : PCI device 1191:0008
febffee8-febffeef : atp86x
febffef4-febffef7 : PCI device 1191:0008
febffef8-febffeef : PCI device 1191:0008
febffef8-febffeef : atp86x IDE, irq 4
febfff00-febfffff : PCI device 13f0:1023
febfff00-febffffe : Sundance Technology IPG Triple-Speed Ethernet

ff000000      ROM_CS1_START (on soc?), FLASH_BASE1_PRELIM (u-boot)
ff800000      ROM_CS0_START
ffc00000      FLASH_BASE0_PRELIM (u-boot)
ffc20000      ramdisk load address
fff00000      TEXT_BASE (u-boot)
fff00100      CFG_RESET_ADDRESS
fff10000      boot image load address
```

## Important constants:

```
CFG_NS16550_CLK          100000000
CONFIG_SYS_CLK_FREQ      100000000
```

## MTD

Addresses are relative to start of mtd at 0xffc00000

```
0x00000000-0x00010000 : "Linux mtd1"
0x00010000-0x00020000 : "Linux mtd2"
0x00020000-0x00300000 : "Linux Ramdisk"
0x00300000-0x00310000 : "U-BOOT BOOT LOADER"
0x00310000-0x00400000 : "Linux Kernel"
```

## Kernel configuration options

From old 2.4 kernel, just something to keep eye on while configuring 2.6 kernels...

- CONFIG\_6xx
- CONFIG\_SANDPOINT
- CONFIG\_PPC\_ISATIMER
  
- CONFIG\_MTD\_PARTITIONS
- CONFIG\_MTD\_CHAR
- CONFIG\_MTD\_BLOCK
  
- CONFIG\_MTD\_CFI
- CONFIG\_MTD\_GEN\_PROBE
- CONFIG\_MTD\_CFI\_AMDSTD

## IRQ

```
sandpoint_map_irq(struct pci_dev *dev, unsigned char idsel, unsigned char pin)
{
    static char pci_irq_table[][4] =
    /*
     *      PCI IDSEL/INTPIN->INTLINE
     *          A   B   C   D
     */
    {
        { 0, 0, 0, 0 }, /* IDSEL 13 - mini-PCI */
        { 1, -1, 2, 0 }, /* IDSEL 14 - NEC USB2.0 */
        { 3, 0, 0, 0 }, /* IDSEL 15 - ADM983 */
        { 4, 0, 0, 0 }
    };

    const long min_idsel = 13, max_idsel = 16, irqs_per_slot = 4;
    return PCI_IRQ_TABLE_LOOKUP;
}
```

```

static u_char sandpoint_openpic_initsenses[] __initdata = {
    (IRQ_SENSE_LEVEL | IRQ_POLARITY_NEGATIVE), /* 17, EPIC IRQ 1 - PCI1 - flash*/
    (IRQ_SENSE_LEVEL | IRQ_POLARITY_NEGATIVE), /* 18, EPIC IRQ 2 - LAN*/
    (IRQ_SENSE_LEVEL | IRQ_POLARITY_NEGATIVE), /* 19, EPIC IRQ 3 - Not used*/
    (IRQ_SENSE_LEVEL | IRQ_POLARITY_NEGATIVE), /* 20, EPIC IRQ 4 - Not used*/
    1
}

```

```

sandpoint_init_IRQ(void)
    /* Map EPIC IRQs 0-3 */
    openpic_set_sources(0, 5, OpenPIC_Addr + 0x10200);
    /* Skip reserved space and map i2c and DMA Ch[01] */
    openpic_set_sources(113, 3, OpenPIC_Addr + 0x11020);
    /* Skip reserved space and map Message Unit Interrupt (I20) */
    openpic_set_sources(118, 1, OpenPIC_Addr + 0x110C0);
    //REX: UART
    openpic_set_sources(121, 1, OpenPIC_Addr + 0x11120); //ttyS0

    openpic_set_sources(122, 1, OpenPIC_Addr + 0x11140); //ttyS1 jack1

```

## mpc10x

```

mpc10x_bridge_init(hose,
    MPC10X_MEM_MAP_B,
    MPC10X_MEM_MAP_B, MPC10X_MAPB_EUMB_BASE)

```

should be changed to:

```

mpc10x_bridge_init(hose,
    MPC10X_MEM_MAP_B,
    MPC10X_MEM_MAP_B,
    0xfc000000) == 0)

```

## IO block

consistent with following:

```

- io_block_mapping(0xfe000000, 0xfe000000, 0x02000000, _PAGE_IO);
+ io_block_mapping(0xfc000000, 0xfc000000, 0x04000000, _PAGE_IO);

```

## Serial

```

#define UART0_INT      121
#define UART1_INT      122

#define SANDPOINT_SERIAL_0 0xFC004500

```

```
#define SANDPOINT_SERIAL_1 0xFC004600
```

## CPU

```
#define CPU_200 1  
#define BASE_BAUD (100000000/16)
```

## Led control strings

Taken from [Beattie's page about DSM-G600](#)

String	Function
SYN	Power led flash, HDD, HDD-Full, USB, WLAN leds off
ZWC	Turn Power Off
ZWO	Power led solid
ZBO	Power led flash
WLO	WLAN led green
WLC	WLAN led off
WBO	WLAN led flash green then off
HDE	HDD led yellow solid
HDC	HDD-Full led off
HBO	HDD-Full led flash yellow
HDN	HDD led off
MMK	USB led green
MMF	USB led yellow
MMC	USB led off
MMI	USB led blink green
MUI	USB led blink yellow
MMN	USB led off
AKO	Unknown
TSO	Power, HDD, USB, WLAN leds green, HDD-Full yellow
TSR	Power, HDD-Full, WLAN leds off, HDD, USB leds yellow
TSC	Power, HDD, HDD-Full, USB, WLAN leds off

## Chassis Status Strings

String	Bit	Function
RKO	0x0001	
UKO	0x0002	
CKO	0x0004	
EKO	0x0008	
PKO	0x0100	
1KO	0x0200	
TS1	0x0010	
TS2	0x0020	
RKR	0x0040	
UKR	0x0080	
IOK	0x1000	
LOK	0x2000	
NOK	0x4000	
AKI	0x8000	