Linux and the "Digital Home Technologies PCB 01070201"

By: Max Klein (maxklein@stanford.edu)
Jeremy Lujan
Tony Roberts

Overview

The board is from a multimedia console being developed in the early 2000's, apparently for audio distribution throughout a house. The board is based around the IBM PowerPC (PPC) 405GP, also code-named "Walnut." The chip has many integrated peripherals, such as two UARTs (one of which is the DB-9 on the back of the board, used for a 9600,8,n,1 RS-232 console), a PCI controller, SDRAM controller, 10/100 Ethernet MAC (connected to the National DP83843 physical interface IC), and various other items. The board also has an integrated 10/100 fast ethernet controller, a Promise IDE controller, as well as two PCI slots and one SDRAM slot.

Power

The system needs a regulated +12VDC power supply. The barrel connector at the back side of the board is wired for this use, with tip positive. We opted for a 3A supply since it will power not only the board but also a connected hard drive (not included). One can be found at Tanner's for about $8, ready to go. The 12VDC should be regulated, as it is passed directly to the hard drive power port, next to the IDE controller. The pinout of this port is standard, with +12 being supplied on the pin next to the IDE connector, ground at the middle two pins, and a regulated +5 on the pin at the outside of the board. This connector SUPPLIES power to a drive, and should not be used to back-power the motherboard. There are several LM voltage regulators on the board, which deliver the 2.5V core power to the PPC, 3.3V I/O power to the PPC and peripherals, and the 5V needed by the external drive.

ROM Monitor / Boot Loader

Lucky for us hardware hackers, the motherboard has a 4Mbit FLASH chip (the AMD AM29LV040B) which has the open source PPCBoot version 1.1.2 preloaded onto it, making our life much easier. The chip on our board was stamped with "PPCBOOT 1.00," however the actual version is 1.1.2. The PPCBoot project is not defunct, however it has been moved into the UBoot project, and UBoot images are binary compatible with PPCBoot images.

RAM

From our preliminary testing, the system recognized both single and double-sided RAM, ECC and not; however the only modules we have actually seen work are single sided non-ECC modules. The memory stick included with the board has a Column Address Select (CAS) latency of 2, which could be why our double-sided modules (all of which happened to have a CAS latency of 3) did not work past the memory remapping operation. The boot sequence shown below was done with a 128MB DIMM installed, however everything also worked fine with the included 32MB DIMM.

Linux

Our board is now running the PowerPC port of Linux, which immediately detected many of the peripherals, including the ethernet controller. The Promise IDE controller and PCI Firewire (IEEE 1394) card, however, were not detected, and will need to be included in a custom build of the kernel to work.

Getting it up

First off, our board did not have any information in the environment partition of the flash chip, such as a MAC address. To manually configure the MAC, type:

```
setenv ethaddr DE:AD:BE:EF:DE:AD
```

The above example sets the MAC address to DEADBEEFDEAD. A basic kernel, pre-compiled and packaged into an image that PPCBoot / UBoot can read, is available from:
ftp://ftp.denx.de/pub/linux/images/amcc/walnut/uImage

We decided to boot our board from a TFTP server to make life easy. This image can be copied over an ethernet network by placing it on a TFTP server and running the following commands:

```
setenv serverip 10.0.0.21           (10.0.0.21 is our TFTP server)
```

```
tftpboot 400000 uImage          (400000 is the target RAM address, uImage is the TFTP file name)
dhcp
```

NOTE: the tftpboot 400000 command will try to load the file into the physical ram, and should be executed before running the dhcp command to ensure that the target memory address is set to 0x400000, and not 0x100000 as dhcp will default to. If you try to load it into address 0x100000, the file will download but uncompressing the kernel will usually fail. At this point, you now have a bootable image in RAM at 0x400000. To boot it, simply type:

```
bootm 400000
```

The 400000 is not needed if you have already specified 400000 in the tftpboot command, but it doesn't hurt to use it. If you just do what we've said so far, the system will bootstrap and transfer execution to the kernel, which will start to load, and then panic because there is not init or other necessary userland applications.

A basic PPC userland can be found in the following image:
ftp://ftp.denx.de/pub/LinuxPPC/usr/src/SELF/images/ppc_4xx/ramdisk_image.gz

We mounted this image on the NFS server and then copied the contents to the net boot directory (/home/diskless/dingdong in our setup). To boot the kernel and have it mount this NFS userland directory, the kernel arguments must be set as follows (all on one line) before executing the bootm command:

```
setenv bootargs -s console=ttyS0,9600 root=/dev/nfs rw nfsroot=10.0.0.21:/home/diskless/dingdong
ip=10.0.0.45::10.0.0.1:255.255.255.0:dingdong:eth0:off
```

Again, to help understand the above:
10.0.0.21 is our NFS server
/home/diskless/dingdong is the location of the userland copied from the above archive
10.0.0.45 is the local IP address
10.0.0.1 is the gateway
255.255.255.0 is the local netmask
dingdong is the local hostname

After running this and executing bootm, the system should boot into a shell as shown below.

For more advanced procedures, such as building your own kernel, go to:
http://www.denx.de/wiki/DULG/BoardSelect

and select "walnut" as the board. Problems, suggestions, or neat ideas? E-mail us :)
Enjoy!

---Begin session log---

```
PPCBoot 1.1.2 (Feb 13 2002 - 15:46:54)

CPU:   IBM PowerPC 405GP Rev. E at 264 MHz (PLB=66, OPB=33, EBC=33 MHz)
       PCI async ext clock used, internal PCI arbiter enabled

Board: ### No HW ID - assuming WALNUT405
DRAM:
Reading DIMM...
Config timing register
  Raw i2c data[27] 0x14; data[29] 0x14; data[127] 0xaf
  bus_period 15
single-sided DIMM
sdram0_pmit 7c00000
sdram0_besr0 ffffffff
sdram0_besr1 ffffffff
```

```
sdram0_ecccfg 0
sdram0_eccesr ffffffff
sdram0_rtr 20880000
sdram0_tr 10a801a
sdram0_b0cr 84001
sdram0_b1cr 0
sdram0_b2cr 4084001
sdram0_b3cr 0
sdram0_cfg 80800000
128 MB
Stack Pointer before machine check 07faef70
New Stack Pointer is: 07faef68
Relocating addr_sp 07faef68; bd 07faef8c; addr 07fd0000
Now running in RAM - PPCBoot at: 07fd0000
FLASH: 512 kB
*** Warning - bad CRC, using default environment

**EJK PCI_Init
PCI:   Bus Dev VenId DevId Class Int
PCI_Scan:Device 0 is present Bus Func 0 ClassCode 600
PCI_Config_Device:Read Base Addr Reg 1 = 0x80000008
  ***PCI Memory space = 0x80000000 bytes
Device 1 not present
Device 2 not present
PCI_Scan:Device 3 is present Bus Func 1800 ClassCode c00
PCI_Config_Device:Read Base Addr Reg 0 = 0xfffff800
  ***PCI Memory space = 0x1000 bytes
PCI_Config_Device:Read Base Addr Reg 1 = 0xffffc000
  ***PCI Memory space = 0x4000 bytes
        00  03  104c  8021  0c00  1c
PCI_Scan:Device 4 is present Bus Func 2000 ClassCode 101
IDE device found:
PCI_Config_Device:Read Base Addr Reg 0 = 0x000001f1
  ***PCI I/O space = 0x10 bytes 0
IDE-Just Set CFG_ATA_BASE_ADDR to e8000000
PCI_Config_Device:Read Base Addr Reg 1 = 0x000003f5
  ***PCI I/O space = 0x4 bytes 10
PCI_Config_Device:Read Base Addr Reg 2 = 0x00000171
  ***PCI I/O space = 0x10 bytes 20
PCI_Config_Device:Read Base Addr Reg 3 = 0x00000375
  ***PCI I/O space = 0x4 bytes 30
PCI_Config_Device:Read Base Addr Reg 4 = 0xffffffc1
  ***PCI I/O space = 0x40 bytes 40
PCI_Config_Device:Read Base Addr Reg 5 = 0xfffe0000
  ***PCI Memory space = 0x20000 bytes
Hardcode address e80001f0
        00  04  105a  0d30  0101  1d
Device 5 not present
Device 6 not present
Device 7 not present
Device 8 not present
Device 9 not present
In:    serial
Out:   serial
Err:   serial
IDE:   cmd_ide:About to init PIO timing modes.
PIO Mode 0: setup=70 ns/325033 clk  len=165 ns/2300066 clk  hold=30 ns/1980000 cl
k
PIO Mode 1: setup=50 ns/3300000 clk  len=125 ns/3955033 clk  hold=20 ns/1320000 c
```

```
lk
PIO Mode 2: setup=30 ns/1980000 clk  len=101 ns/2371033 clk  hold=15 ns/990000 cl
k
PIO Mode 3: setup=30 ns/1980000 clk  len= 80 ns/985033 clk  hold=10 ns/660000 clk
PIO Mode 4: setup=25 ns/1650000 clk  len= 70 ns/325033 clk  hold=10 ns/660000 clk
cmd_ide:About to call reset.
Bus 0 dev 0: at address e80001f0not available  Status = 0x00
Hit any key to stop autoboot:  0

** Device 0 not available
## Booting image at 00100000 ...
Bad Magic Number
=> setenv ethaddr DE:AD:BE:EF:DE:AD
=> setenv serverip 10.0.0.21
=> tftpboot 400000
405GP Eth Status:
  data len error 0
  rx frames 0
  rx 0
  rx_prot_err 0
  int_err 0
  tx_err_log:
    [0] 0
    [1] 0
    [2] 0
    [3] 0
    [4] 0
    [5] 0
    [6] 0
    [7] 0
    [8] 0
    [9] 0
  rx_err_log:
    [0] 0
    [1] 0
    [2] 0
    [3] 0
    [4] 0
    [5] 0
    [6] 0
    [7] 0
    [8] 0
    [9] 0

ENET Speed is 100 Mbps - FULL duplex connection
*** ERROR: `ipaddr' not set
=> dhcp
405GP Eth Status:
  data len error 0
  rx frames 0
  rx 0
  rx_prot_err 0
  int_err 0
  tx_err_log:
    [0] 0
    [1] 0
    [2] 0
    [3] 0
    [4] 0
    [5] 0
    [6] 0
    [7] 0
    [8] 0
    [9] 0
  rx_err_log:
    [0] 0
    [1] 0
    [2] 0
    [3] 0
    [4] 0
    [5] 0
    [6] 0
    [7] 0
    [8] 0
    [9] 0

ENET Speed is 100 Mbps - FULL duplex connection
BOOTP broadcast 1
DHCPHandler: got packet: (src=67, dst=68, len=300) state: 3
Filtering pkt = 0
DHCPHandler: got DHCP packet: (src=67, dst=68, len=300) state: 3
DHCP: state=SELECTING bp_file: "/dingdong/boot/pxegrub"
TRANSITIONING TO REQUESTING STATE
Bootfile: /dingdong/boot/pxegrub
DhcpSendRequestPkt: Sending DHCPREQUEST
Transmitting DHCPREQUEST packet: len = 308
DHCPHandler: got packet: (src=67, dst=68, len=300) state: 4
Filtering pkt = 0
DHCPHandler: got DHCP packet: (src=67, dst=68, len=300) state: 4
DHCP State: REQUESTING
Bootfile: /dingdong/boot/pxegrub
DHCP client bound to address 10.0.0.45
ARP broadcast 1
Got good ARP - start TFTP
TFTP from server 10.0.0.21; our IP address is 10.0.0.45
Filename '/dingdong/boot/pxegrub'.
Load address: 0x400000
Loading: #################################################################
         #################################################################
         ###############################################################
done
Bytes transferred = 987515 (f117b hex)
=> setenv bootargs -s console=ttyS0,9600 root=/dev/nfs nfsroot=10.0.0.21:/home/di
skless/dingdong ip=10.0.0.45::10.0.0.1:255.255.255.0:dingdong:eth0:off
=> bootm
## Booting image at 00400000 ...
   Image Name:   Linux-2.6.15-rc2-g09df1d09
   Created:      2005-12-03  21:39:01 UTC
   Image Type:   PowerPC Linux Kernel Image (gzip compressed)
   Data Size:    987451 Bytes = 964 kB = 0 MB
   Load Address: 00000000
   Entry Point:  00000000
   Verifying Checksum ... OK
   Uncompressing Kernel Image ... OK
## Current stack ends at 0x07FAEC18 => set upper limit to 0x00800000
## cmdline at 0x007FFF00 ... 0x007FFF84
No initrd
memstart    = 0x00000000
```

```
memsize     = 0x08000000
flashstart  = 0xfff80000
flashsize   = 0x00080000
flashoffset = 0x00030000
sramstart   = 0x00000000
sramsize    = 0x00000000
bootflags   = 0x0000a000
procfreq    =    264 MHz
plb_busfreq =     66 MHz
pci_busfreq =     33 MHz
ethaddr     = DE:AD:BE:EF:DE:AD
IP addr     = 10.0.0.45
baudrate    =   9600 bps
getc        = 0x07fd3284
tstc        = 0x07fd32dc
putc        = 0x07fd31c8
puts        = 0x07fd3240
printf      = 0x07fe171c
install_hdlr= 0x07fd2d80
free_hdlr   = 0x07fd2e10
malloc      = 0x07fd456c
free        = 0x07fd4a70
## Transferring control to Linux (at address 00000000) ...
Linux version 2.6.15-rc2-g09df1d09 (wd@pollux) (gcc version 4.0.0 (DENX ELDK 4.0
4.0.0)) #1 Sat Dec 3 22:39:00 MET 2005
IBM Walnut port (C) 2000-2002 MontaVista Software, Inc. (source@mvista.com)
Built 1 zonelists
Kernel command line: -s console=ttyS0,9600 root=/dev/nfs nfsroot=10.0.0.21:/home/
diskless/dingdong ip=10.0.0.45::10.0.0.1:255.255.255.0:dingdong:eth0:off
PID hash table entries: 1024 (order: 10, 16384 bytes)
Warning: real time clock seems stuck!
Dentry cache hash table entries: 32768 (order: 5, 131072 bytes)
Inode-cache hash table entries: 16384 (order: 4, 65536 bytes)
Memory: 127616k available (1620k kernel code, 404k data, 124k init, 0k highmem)
Mount-cache hash table entries: 512
NET: Registered protocol family 16
PCI: Probing PCI hardware
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered
Generic RTC Driver v1.07
Serial: 8250/16550 driver $Revision: 1.90 $ 4 ports, IRQ sharing disabled
serial8250: ttyS0 at MMIO 0x0 (irq = 0) is a 16550A
serial8250: ttyS1 at MMIO 0x0 (irq = 1) is a 16550A
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
loop: loaded (max 8 devices)
PPC 4xx OCP EMAC driver, version 3.53
mal0: initialized, 1 TX channels, 1 RX channels
eth0: emac0, MAC de:ad:be:ef:de:ad
eth0: found Generic MII PHY (0x09)
Found: AMD AM29LV040B
PPC40x-flash: Found 1 x8 devices at 0x0 in 8-bit bank
number of JEDEC chips: 1
cfi_cmdset_0002: Disabling erase-suspend-program due to code brokenness.
Creating 3 MTD partitions on "PPC40x-flash":
0x00000000-0x00020000 : "reserved"
0x00020000-0x00040000 : "env"
0x00040000-0x00080000 : "u-boot"
```

```
NET: Registered protocol family 2
IP route cache hash table entries: 2048 (order: 1, 8192 bytes)
TCP established hash table entries: 8192 (order: 3, 32768 bytes)
TCP bind hash table entries: 8192 (order: 3, 32768 bytes)
TCP: Hash tables configured (established 8192 bind 8192)
TCP reno registered
TCP bic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
eth0: link is down
eth0: link is up, 100 FDX
IP-Config: Complete:
      device=eth0, addr=10.0.0.45, mask=255.255.255.0, gw=10.0.0.1,
     host=dingdong, domain=, nis-domain=(none),
     bootserver=255.255.255.255, rootserver=10.0.0.21, rootpath=
Looking up port of RPC 100003/2 on 10.0.0.21
Looking up port of RPC 100005/1 on 10.0.0.21
VFS: Mounted root (nfs filesystem) readonly.
Freeing unused kernel memory: 124k init
serial console detected.  Disabling virtual terminals.


BusyBox v0.60.1 (2002.10.24-09:26+0000) Built-in shell (msh)
Enter 'help' for a list of built-in commands.

#
```

---End session log---