User's Manual for the Boundary Devices Neon® board

Boundary Devices

March 21, 2005

Contents

1 Intended Audience

This document aims to provide the information needed to integrate the Neon® board into your application. As such, it addresses both hardware and software integration.

2 Overview of features

The following are highlights of the Neon® board.

- Available with Windows Ce or Linux Operating Systems
- Full featured Boot Loader for custom startup
- 400 MHz Intel PXA-255 CPU
- 32 or 64MB SDRAM
- 8 or 32MB Intel StrataFlash (tm) EEPROM
- Silicon Motion SM-501 Graphics Controller
- Active Matrix LCD Support,
- Including Full-Motion Video
- STN Passive LCD Display Support
- 4 or 5-Wire Resistive Touch-Screen Support
- 44KHz Stereo 16-Bit Audio Output, for Headphones or Speakers
- 44KHz Monaural Audio Input (microphone)
- 3 RS-232 or TTL Serial Ports
- 1 USB 1.1 Slave Port
- 1 USB 1.1 Master Port
- Built-in 10/100 Ethernet Controller,
- Including Power-Over-Ethernet (PoE) Capability
- Built-in Interface for Magnetic Stripe Readers and Printers
- MMC Slot for Expanded Storage
- General Purpose I/O for Device Control
- Built-in Switching Power Supply for 12V DC Input
- JTAG Interface
- Customized Versions Available

3 Hardware feature

3.1 Layout

As shown in Figure 1, the Neon® board contains a wide variety of I/O options for use in your application. Note that some of these may not be populated on an evaluation or production board.

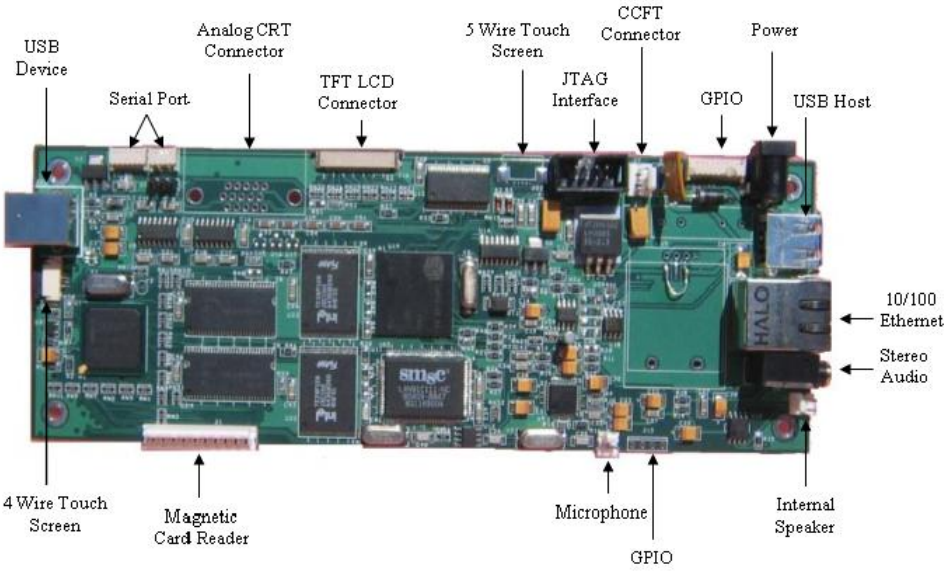

Figure 1: Neon board

3.2 Mounting

The Neon® board measures 2.75" by 6.75", slightly larger than the Hitachi® 6.2" display, to allow for easy mounting.

There are four mounting holes 1/4" from each edge in each of the four corners, and the holes are 1/8" in diameter.

3.3 Connector reference

The following is a list of all connector part numbers used on the Neon® platform for use in identifying mating parts for your application. Note that

Boundary Devices will periodically switch vendors for these parts, but will notify you of any changes that require a new mating part.

Description	Manufacturer	Part
USB Master	FCI	87520-0010B
USB Slave	SINGATRON	KS-001-BNW
I2C	FCI	68897-001
Ethernet	Halo	HFJ11-2450E
Stereo Audio	Singatron	2SJ-43723N13
Backlight inverter	Molex	53048-0210
MMC/SD	AVX	14 5638 009 511 862
TFT Display		
Touch Screen	Molex	52207-0590
Serial Port	FCI	68897-001
JTAG	Molex	53048-0810

3.4 Electrical characteristics

4 Software features

As provided by Boundary Devices , the Neon® board supports either Windows CE 5® or Linux.

To simplify the installation of either, the Das U-Bootboot loader is installed on our evaluation boards, and two MMC cards are shipped to allow the use of either operating system.

4.1 Das U-Boot

The Das U-Boot Boot Loader is a full-featured loader for either Linux or Windows CE that supports a wide variety of options for loading your Operating System and application.

Boundary Devices ships U-Boot both as a binary image and as source code in the form of a patch that adds support for either Neon or BD-2003 devices.

The binary image may be burned directly to sector zero of the on-board flash.

The source code will require a set of Linux or Cygwin(Windows) tools for cross-compilation. The following section will detail the requirements and steps for building.

4.1.1 Requirements for building under Linux

Since the Das U-Boot project uses GNU tools, most of the required components will generally be available on a GNU/Linux system.

The three pieces which may not commonly be installed are the bzip2 and wget packages and an ARM cross compiler.

Boundary Devices typically uses GCC-2.95.3 to create U-Boot images, since that matches what we use to build the Linux image to run on the Neon itself, but the binary distribution of GCC-3.4.3 from GNUARM is a nice alternative.

4.1.2 Requirements for building under Windows with Cygwin

There are two primary requirements for building under Windows.

The first, Cygwin, provides a set of Unix utilities under the Windows operating system. Since the Cygwin installer allows components to be selected individually, the following list shows the requirements for building a Das U-Boot image with Neon® support. Note that this list is probably incomplete, but these should be the only required items which differ from the Cygwin defaults.

Base/diffutils
Devel/binutils
Devel/gcc
Devel/make
Devel/patchutils
Utils/bzip2
Web/wget

The second requirement for building is the X-Scale cross-compiler itself. The GNUARM project provides a wealth of information needed to build a cross-compiler for ARM processors. Thankfully, it also provides an installer. As of this writing, Boundary Devices currently uses the GCC-3.4.3 package for Cygwin.

4.1.3 General build steps

Quick start:

```
wget http://easynews.dl.sourceforge.net/sourceforge/u-boot/u-boot-1.1.2.tar.bz2
bzcat u-boot-1.1.2.tar.bz2 | tar -xvf -
wget http://boundarydevices.com/uboot_neon_bd2003.diff
patch -p0 < uboot_neon_bd2003.diff
cd u-boot-1.1.2
CROSS_COMPILE=arm-elf- make neon_config
CROSS_COMPILE=arm-elf- make
```

Explanation.

The first four lines retrieve and extract the Das U-Boot sources and add support for the Neon® and BD-2003 devices.

The last two lines configure for the Neon® board itself, and finally, build a U-Boot binary. When complete, you'll find a file named `u-boot.bin` in your u-boot-1.1.2 directory.

4.1.4 Tailoring U-Boot for your application

The Boundary Devices patches (uboot_neon_bd2003.diff) make a variety of decisions about the boot process which may not match with the needs of your application.

In general, the file `u-boot-1.1.2/include/configs/neon.h` defines these choices.

In particular, the distributed copy currently expects a Windows BMP file named `bdlogo.bmp` to be present on the MMC card and writes it to the display, then loads an operating system image from a file named `nk.nb0` to

RAM address 0xa0030000 and executes it.

Both of these are defined by the lines which resemble this:

```
#define CONFIG_BOOTCOMMAND "mmcinit; " \
                           "fatload mmc 0 a0008000 bdlogo.bmp ; " \
                           "bmp display a0008000 ; " \
                           "fatload mmc 0 A0030000 nk.nb0 ; " \
                           "g A0030000"
```

As mentioned previously, the Das U-Boot Boot Loader is a very capable loader with support for USB and network boot, including BOOTP/DHCP, and NFS mounting support. Please refer to the Das U-Boot website for details.

4.2 Windows CE

As mentioned earlier, the Neon® board ships with a runnable Windows CE 5.0 image on MMC card. A Board Support Package is also available and necessary to tailor the operating system for a given application.

The following sections describe the process of producing an image matching the one shipped with the Neon® board.

4.2.1 Prerequisites and components

Most of the tools needed to create a bootable Windows CE 5® application for the Neon® board are provided by Microsoft. The following is a complete list of components and where they may be obtained.

Windows CE 5®	Microsoft
Embedded Visual C++ 4.0	Microsoft
Embedded Visual C++ Service Pack	Microsoft
Neon® Board Support Package	Boundary Devices

4.2.2 BSP Installation

The Neon BSP is made available as a Windows installer file on the Boundary Devices website. This file defines a single BSP for the BD2003 and SM501-supporting variants. Installation consists of running the `.msi` file.

```
c:\> wget http://www.boundarydevices.com/bspNeon320x240.msi
c:\> .\bspNeon320x240.msi
```

As a reference tool for the content of the BSP, you should consider using MSI2XML to view the content.

4.2.3 Building the demo

The Platform Builder project used to construct our sample image may be found on the Boundary Devices web site [1]. After installation of the BSP, this project may be copied to a new directory within the WINCE500 PBWorkspaces directory and built using Platform Builder.

```
C:\WINCE500\PBWorkspaces>md bdWeb
C:\WINCE500\PBWorkspaces>cd bdWeb
C:\WINCE500\PBWorkspaces\bdWeb>wget http://boundarydevices.com/bdWeb.pbxml
--17:37:40--  http://boundarydevices.com/bdWeb.pbxml
           => 'bdWeb.pbxml'
Resolving boundarydevices.com... 66.113.228.134
Connecting to boundarydevices.com[66.113.228.134]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 45,478 [text/plain]
100%[=====================================================================>]
17:37:40 (58.90 KB/s) - 'bdWeb.pbxml' saved [45478/45478]

C:\WINCE500\PBWorkspaces\bdWeb>.\bdWeb.pbxml
C:\WINCE500\PBWorkspaces\bdWeb>
```

After this is done, you should be able to build the sample application through the `Build OS|Sysgen` and `Build OS|Build and Sysgen Current BSP` menu options.

[1] We found out how to do this after the painful construction of our 4.2. build notes.

4.3 Linux Support

The Linux Environment for Boundary Devices boards consists of three primary pieces, a toolchain, the kernel and device drivers, and a set of libraries and applications.

In order to allow configuration of each, a modified version of the PTXdist tool is used to configure and build each.

4.3.1 Toolchain

Before the kernel and applications can be built, it is first necessary to have a cross-compiler toolchain. The following package contains a build script for building a GCC-2.95.3 toolchain under Linux.

The following is a short list of the usage steps.

```
$ wget http://boundarydevices.com/boundary-2.4.19-rmk7-pxa2.patch.bz2
$ bzcat boundary-2.4.19-rmk7-pxa2.patch.bz2 | tar -xvf -
$ cd linux-2.4.19
$ make menuconfig
$ make all
```

4.3.2 Kernel

Arm-Linux kernel version 2.4.19	Linux kernel patches for ARM processors
PXA Patches	Intel PXA support for ARM-Linux
Boundary Devices patches	Boundary Devices support

4.3.3 Libraries and Applications

Place links and references as in this page here.

5 Development Tools

5.1 JTAG system-level debugger

Place links and references as in this page here.

5.2 TeraTerm blast extensions

Describe TeraTerm extensions and the Blast® protocol here. Also add download link.