

ANEXO 5. INFORMACIÓN GENERAL SOBRE QNX (Página web de Versallogic)



Installing and Running QNX Neutrino on an Embedded System

by
Anthony Chisholm

Overview

This document is a quick-start guide to the QNX Neutrino operating system, with special consideration given to VersaLogic products. It can help to familiarize new users with QNX and avoid some of the common pitfalls encountered during installation and operation.



QNX Neutrino is the latest offering from QNX Software Systems, Ltd. (QSSL), founded in 1980 and headquartered in Ottawa, Canada. It is an operating system designed from the ground up for embedded systems. It is based on a real-time microkernel, and maintains a strong focus on reliability and modularity. While it can run on very lean devices, it can also scale to large distributed or symmetric multiprocessing (SMP) systems. A QNX target is most often x86-based, but can also be based on PowerPC, MIPS, SH-4, ARM, or XScale architectures.

As its name suggests, QNX is POSIX-compliant and similar in look and feel to UNIX and Linux. However, Neutrino architecture is not directly compatible with Linux and UNIX, and is also not compatible with its predecessors, QNX4 and QNX2.

It will be worthwhile to add a note here on the potentially confusing system of QNX versions and naming conventions. QNX2 was the first major OS release from QSSL, and QNX4 was the second. When QNX6 was released, the package was logically broken down into 2 parts, the OS and the Graphical User Interface (GUI). The OS portion was dubbed QNX Neutrino, and considered separate from the Photon microGUI. "Neutrino" is also the name given to the microkernel at the heart of the OS. Later, QSSL released QNX RTP (Real-Time Platform), which was a superset of QNX6 that included tools for developers. QNX RTP was later enhanced and renamed QNX Momentics.

As of this writing, the latest version of the QNX Neutrino RTOS is 6.2.1, and the latest version of the Neutrino microkernel is 2.1.3. Unless otherwise specified, these versions are implied throughout the document.

Operation of QNX Neutrino has been verified on the following boards offered by VersaLogic:

- VSBC-8 Pentium III/Celeron EBX Single Board Computer
- Jaguar Pentium III/Celeron PC/104-Plus Single Board Computer
- Bobcat AMD 5x86 PC/104-Plus Single Board Computer
- VSBC-6 Pentium/K6 EBX Single Board Computer
- Panther Pentium/K6 PC/104-Plus Single Board Computer
- VL-686-2 Pentium/K6 STD32 Bus CPU with video, Ethernet

Installation

The QNX Momentics Development Suite is freely available for non-commercial use at <http://www.qnx.com/eval/>. Commercial users may also download a 30-day evaluation. The QNX Neutrino RTOS is on this CD, and can be installed onto an empty hard drive.

Set up your target by attaching power, a keyboard, PS/2 mouse, display, CD-ROM drive, and a hard drive (preferably empty). If your target supports CD booting (as does the VSBC-8, Jaguar, and Bobcat), then use this method. Otherwise, create a boot floppy with the Windows utilities on the installation CD. Installation is straightforward, and can take as little as 5 minutes.

After the installer reboots your target system, enter CMOS setup to make the hard drive your first boot device. If booting off the hard drive causes an "Unable to locate fs-pkg" error, then see

Gotcha #1.

After the first boot script runs, you are presented with the Video Card Setting dialog. If this dialog hangs the system, see Gotcha #2. All graphics chipsets used on VersaLogic CPU boards are supported by this installation CD. However, the hardware cursor must be disabled if your board uses the C&T 69030 video controller. Click Apply, and then Done to complete the video setup and the QNX installation. Log in as root, and explore the Photon GUI.

Helpful Tips

For those familiar with a POSIX-style environment, QNX will feel mostly familiar. For others, there can be quite a curve to learn the basic operations. There is a sea of commands, the vast majority of which you will never use, and many with several pages of command-line parameters. This section will highlight several commands that will help a new user to get started quickly.

As in Linux and Unix, many system administration and configuration functions are only available at the command line. Open a terminal window using the button bar on the right pane.

Most commands and utilities are available at /bin, /sbin, and /usr/bin. To get command-line help on a particular QNX command, type:

```
use <command>
```

You can also check the most up-to-date and detailed utility descriptions online at <http://www.qnx.com/developer/docs/>

General usage tips

Look for a file anywhere in the filesystem:

```
find / | grep <filename>
```

List all currently running processes, including process ID and arguments:

```
ps -Ao pid,args
```

Forcefully terminate a process by name:

```
slay -9 <process name>
```

Executing script files: For an often-repeated set of commands, it is helpful to build a script file, similar to a DOS batch file. To do this, make a text file (eg. commandlist) with one shell command per line. Then execute:

```
esh <commandlist>
```

This opens up a shell on top of the one you're currently using, executes the commands in this new shell, then exits. So any shell-specific variables that you set up in the script, such as present working directory, will be lost when the shell exits. To prevent the new shell from exiting upon completing the script, use interactive mode by adding the "-i" parameter.

Increase verbosity:

If a program is producing unexpected results, you may want a more detailed report of what it is doing. Many QNX commands allow you to increase their verbosity by adding v's as arguments. Add up to four v's to get the most verbosity. Example:

```
mount -vvv /dev/fd0 /fs/fd0
```

Startup sequence

It can be helpful to have a basic idea of how QNX starts up by default.

1. Boot image script runs (specified in the build file with +script directive)
2. Script runs: /etc/system/sysinit
3. Script runs: /etc/rc.d/rc.sysinit
4. <login prompt is given>
5. Script runs: ~/.profile

Prevent Photon from automatically starting at boot time:
Create the empty file:

```
/etc/system/config/nophoton
```

Connectivity tips

Display Ethernet network configuration:

```
ifconfig en0
```

Set up VSBC-6 networking:

Unlike the rest of the VersaLogic boards, Ethernet on the VSBC-6 is ISA-based and not automatically set up during installation. To start the driver and set up network services via DHCP:

```
io-net -d smc9000 -p tcpip  
dhcp.client &
```

Mount a windows share over a LAN:

```
fs-cifs //hostname:hostipaddress:/sharename /mountpoint login  
password &
```

An example:

```
fs-cifs //ANTHONY-XP:192.168.0.112:/QNXarea /fs/workstation mylogin  
mypassword &
```

The contents of your windows share will now appear as a local directory at the mount point.

Test RS-232 or RS-422 serial ports:

1. Open two terminal sessions.
2. In one, enter: `qtalk -m /dev/ser1`
3. In the other, enter: `qtalk -m /dev/ser2`
4. Add a null modem cable between the COM ports and send characters with the keyboard to verify communication in each terminal.

Working with floppy disks:

Format a floppy QNX-style, and make it bootable:

```
fdformat /dev/fd0  
dinit /dev/fd0  
cp bootable.ifs /fs/fd0/.boot
```

Note: The ".ifs" file can be created with the "mkifs" utility (see below).

Mount a QNX-formatted floppy:

```
mount /dev/fd0 /fs/fd0
```

Mount a DOS-formatted floppy:

```
mount -t dos /dev/fd0 /fs/fd0
```

Technical Notes

The mkifs Utility

In an embedded system, it is desirable to customize and build your own Neutrino image to minimize footprint, speed boot time, and ensure that all needed functionality is present. The mkifs utility can be used to generate bootable images based on a script, or build file. The build file specifies what kernel to use, what modules and shared objects to include, and how to start up and configure the system.

To form the image, files are pulled from the currently running environment, combined, and encapsulated into a header. The images generated by mkifs may be very small and can easily fit on a floppy disk. Instructions on how to create build files are beyond the scope of this document. For sample build files, look in "/x86/boot/build" or view the mkifs documentation at:

The output of mkifs is a boot image. This is the first file that gets loaded at boot time. To put these images into use, they must be copied to the boot device at /.boot or /.altboot.

To boot from a newly generated image off a floppy disk, you must initialize the disk and write the image to the .boot file in the root directory. See the Helpful Tips section for help in performing these operations.

Networking

QNX Neutrino supports all the network controllers used on VersaLogic boards.

Board	Controller	QNX Driver
VSBC-8, Jaguar, and Bobcat	Intel 82559ER	speedo
Panther, 686-2, and EPM-NET-100	AMD 79C973	pcnet
VSBC-6	SMC 91C96	smc9000

Setting up networking consists of loading and configuring the hardware driver and protocol interface(s). PCI Ethernet controllers will be automatically set up and configured to find a DHCP server at install time.

You can also use the io-net command to manually start the driver and protocol interface. Specify the type of interface and the IP address of the interface on the command line. For example:

```
io-net -dspeedo -ptcpip en0:216.116.45.244
```

In this example, "en" specifies that the interface type is Ethernet, "0" specifies the first Ethernet interface. For communication with the Internet, you may also add a static route and specify its destination, netmask, and gateway. See the tcpip documentation for syntax details. Note that you must not execute io-net more than once, so specify hardware drivers and protocols at the same time, or load them separately with the mount utility.

Alternatively, if you want to use DHCP, do not specify parameters for the tcpip protocol. Instead, after the io-net command, execute:

```
dhcp.client -i en0
```

It will look for a DHCP server and download the configuration for the specified interface. If no interface is specified, dhcp.client will configure the first unconfigured interface found.

Gotchas

Here are some issues that you may encounter while installing or getting started with QNX Neutrino.

1. When building a QNX image with the mkifs utility, make sure your build script is in the proper text format, or you will get very mysterious errors. There are two types of line terminators that QNX and Windows generally recognize, but mkifs is particular to one type. To convert to the proper type, you can use the QNX textto utility. Also be aware that QNX's GUI text editor, Notepad, will (by default) append format information to the end of your text file, and this will also interfere with mkifs.
2. CPU boards that use the ALi 1543C south bridge chip (Panther, VSBC-6, 686-2) require that DMA mode is disabled during installation and during runtime. The default boot floppy created by the Momentics CD forces DMA mode. For instructions on how to work around this, see the VersaTech KnowledgeBase article: <http://www.versalogic.com/kb/KB.asp?KBID=1282>
3. The 686-2 requires a script modification before graphics mode will work, due to a problem with the PS/2 mouse detection performed by inputtrap. Therefore, the "no graphics" safe mode must be selected prior to the first boot. After the fix has been performed, the 686-2 can boot fully into a graphical environment. The fix is detailed at the end of the VersaTech KnowledgeBase article: <http://www.versalogic.com/kb/KB.asp?KBID=1282>

4. The Bobcat requires a video card for installation. The EPM-SVGA-2 has been tested for this purpose. After installation, mkifs can be used to create a kernel image that uses a serial port console.
5. CPU boards that use the 69030 video controller (VSBC-8 rev 3, Jaguar rev 3) freeze when the hardware cursor is enabled. If the video settings dialog is used, check the "Disable Hardware Cursor" option before clicking Change Mode. If the graphics driver is started from
6. CPU boards that use the 82559ER Ethernet controller (VSBC-8, Bobcat, Jaguar) may experience a "devn-speedo: no such device" error during device initialization. This is reportedly due to a faulty pci-bios server that was a part of QNX 6.1. This issue would likely affect other on-board PCI devices as well. This has been resolved in QNX 6.2.1.

Last Revised: 1/7/04

Contact VersaLogic's Technical Support Department at info@versalogic.com for more information.

Note: Pricing and company information was deemed accurate at the time of writing of this article. VersaLogic makes no claim or warranty that the information contained in this article will not become outdated.

[Main](#) [New](#) [Products](#) [Support](#) [Partners](#) [Stories](#) [Search](#) [Company Info](#)



VersaLogic Corp. (541) 485-8575
3888 Stewart Road - Eugene, OR 97402
<http://www.VersaLogic.com>



E-mail all comments and suggestions to the [Web Administrator](#).

© 1996-2004 VersaLogic Corporation - All Rights Reserved