

COMPARISON BETWEEN QNX® RTOS V6.1 AND WINDOWS CE 3.0

© Copyright Dedicated Systems Experts NV. All rights reserved, no part of the contents of this document may be reproduced or transmitted in any form or by any means without the written permission of Dedicated Systems Experts NV, Gerrechtsstraat 21, 1070 Brussels, Belgium.

Disclaimer

Although all care has been taken to obtain correct information and accurate test results, Dedicated Systems Experts and Dedicated Systems Magazine cannot be liable for any incidental or consequential damages (including damages for loss of business, profits or the like) arising out of the use of the information provided in this report, even if Dedicated Systems Experts and Dedicated Systems Magazine have been advised of the possibility of such damages.

<http://www.dedicated-systems.com>

E-mail: info@dedicated-systems.com

Doc no.: **DSE-RTOS-EVA-014**

Issue: **2.0**

Date: **May 13, 2002**

1	Introduction.....	4
2	Installation and Configuration.....	5
2.1	QNX RTOS v6.1	5
2.2	Windows CE 3.0	5
3	RTOS Architecture	6
3.1	System Architecture	6
3.1.1	QNX RTOS v6.1	6
3.1.2	Windows CE 3.0	6
3.2	Basic System Facilities.....	7
3.2.1	Task Handling Method.....	7
3.2.2	Memory Management Method.....	7
3.2.3	Interrupt Handling Method	8
4	API Richness.....	10
5	Internet support.....	11
5.1	QNX RTOS v6.1	11
5.2	Windows CE 3.0	11
6	Tools.....	12
6.1	QNX RTOS v6.1	12
6.2	Windows CE 3.0	12
7	Documentation and Support	13
7.1	QNX RTOS v6.1	13
7.2	Windows CE 3.0	13
8	Development methodology	14
8.1	QNX RTOS v6.1	14
8.2	Windows CE 3.0	14
9	Test results.....	15
10	Conclusion.....	18
11	References.....	19
12	Appendix A: Product ratings.....	20
12.1	QNX RTOS v6.1	21
12.2	Windows CE 3.0	22
13	Appendix B: QNX RTOS v6.1 report excerpts	23
14	Appendix C: Windows CE 3.0 report excerpts.....	26
15	Appendix D : Summary test identification codes.....	28

2 Installation and Configuration

QNX RTOS v6.1	0		10
Windows CE 3.0	0		10

2.1 QNX RTOS v6.1

The QNX RTOS v6.1 is quick and easy to install. After only a few minutes the basic modules are installed i.e., the kernel and a user interface (Photon Windowing System). Additional packages like compilers etc, can be installed by means of the package manager.

Configuring the QNX RTOS v6.1 is reasonably straightforward. When installing the full environment, the most critical components like storage devices and network cards are detected automatically. If further configuration is necessary, it can be done through the graphical user interface.

Building a custom QNX image is done through build files. Modules can be added, removed and configured by manually editing these text-based files. The documentation contains plenty of examples of such build files. Although there's no graphical tool to accomplish this, customizing and configuring an image never proved to be a major problem during this evaluation.

2.2 Windows CE 3.0

The first step to using Windows CE 3.0 is to install the platform builder software. Platform builder is the set of tools that is used to create a custom Windows CE platform. The platform builder comes on 14 CDs and is capable of creating ARM, MIPS, PowerPC, SH or Intel x86 based platforms. For this evaluation, only the Intel x86 component was installed. Installing platform builder is similar to installing any other Microsoft software application, and is pretty straightforward and user-friendly.

The next step is to use the platform builder to create, customize and configure a platform. Configuring the platform to your requirements is a complicated and intricate process. Although the platform builder integrated development environment (IDE) includes wizards for creating platforms and components, most of the configuration work will happen through manually editing registry files, manipulating a set of environment variables and modifying various other configuration scripts. This makes the configuration process a difficult task to newcomers.

To simplify the process of building a custom platform somewhat, platform builder includes eight Windows CE operating system configurations. An OEM can use one of these configurations as a starting point and build further on that.

3 RTOS Architecture

For a description of the ratings, the reader is referred to appendix D in the document “report definition and test plan”, which can be downloaded from our website (<http://www.dedicated-systems.com/encyc>)

QNX RTOS v6.1	0	<div style="display: flex; width: 100px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div>	9	10
Windows CE 3.0	0	<div style="display: flex; width: 100px; height: 15px; background-color: #ccc; border: 1px solid #000;"></div>	7	10

3.1 System Architecture

3.1.1 QNX RTOS v6.1

The QNX RTOS v6.1 has a true client-server architecture, consisting of a microkernel and optional cooperating processes. The microkernel implements only the core services, like threads, signals, message passing, synchronization, scheduling and timer services. Additional functionality is implemented in cooperative processes, which act as server processes and respond to the request of client processes (e.g. an application process). Examples of such server processes are the file system manager, process manager, device manager, network manager, etc. While the kernel runs at privilege level 0 of the Intel processor, the managers and device drivers run at levels 1 and 2 (to perform IO operations). Application processes on the other hand run at privilege level 3, and can therefore only execute general instructions of the processor.

Every process runs in its own virtual memory space. The QNX RTOS is a message based OS, and can seamlessly be distributed over multiple nodes. The RTOS supports SMP, and implements several HA (High Availability) features.

3.1.2 Windows CE 3.0

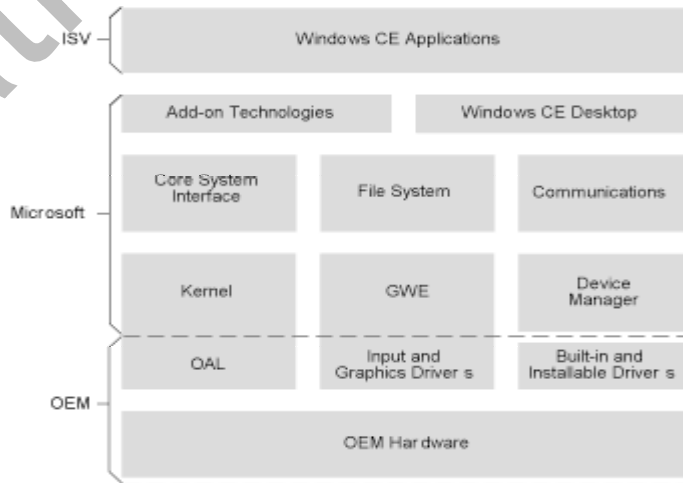


Figure 3-1: Windows CE modular architecture