



Schnellstart

**QNX Momentics Toolsuite
QNX Neutrino RTOS**

10 Schritte zu Ihrem ersten QNX-Programm

In 10 einfachen Schritten
QNX-Tools und Betriebssystem
installieren, konfigurieren und losprogrammieren!



1

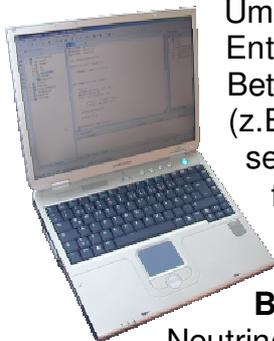
Die Voraussetzungen

Um QNX-Programme zu schreiben, brauchen Sie zum einen die QNX-Entwicklungsumgebung, **QNX Momentics**. Diese können Sie unter Windows 2000 oder Windows XP installieren. Alternativ läuft diese auch unter Linux, Solaris oder QNX Neutrino.



Falls Sie keine QNX Momentics CD haben, laden Sie sich diese als ISO-Image herunter:
<http://www.qnx.com/products/eval/index.html>

Zum anderen brauchen Sie natürlich eine Plattform, auf der Sie das **QNX Neutrino** Echtzeitbetriebssystem selbst und die von Ihnen entwickelten Programme laufen lassen können.



Um sich mit QNX Neutrino vertraut zu machen, gibt es zwei Wege: Entweder Sie installieren Neutrino auf einem normalen PC als Betriebssystem, oder Sie lassen Neutrino auf einer Referenzplattform (z.B. mit PPC-, XScale- oder SH4-CPU) laufen. Da das System selbst aber auf allen Plattformen gleich aufgebaut ist und gleich funktioniert, wählen wir hier die Installation auf einem PC. Möchten Sie alternativ eine Referenzplattform (z.B. Referenzdesign eines Prozessor-Herstellers) verwenden, benötigen Sie dafür ein **QNX Board Support Package**. Die Dokumentation dazu erläutert, wie Neutrino auf dem entsprechenden System installiert wird.

Für den schnellen Start verwenden Sie am besten einen zweiten PC - es kann auch ein etwas älteres Modell sein. Auf der Festplatte sollte eine Partition von ca. 300 MB (nur das OS mit allen Komponenten) bis ca. 2 GB (zusätzlich komplette Entwicklungsumgebung für alle unterstützten CPU's) frei sein. Ein auf einer anderen Partition vorhandenes Windows wird nicht berührt. Sie können Neutrino auch direkt von CD starten, falls Sie beispielsweise keinen Platz mehr auf Ihrer Festplatte haben.



Falls Sie keine QNX Neutrino CD haben, laden Sie diese als ISO-Image herunter:
<http://www.qnx.com/products/eval/index.html>

Free 30-day QNX Momentics Development Suite v6.3 commercial product evaluation.

Download today!

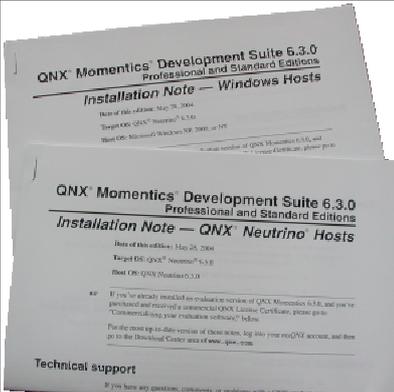
Step 1: Select the desired host

To proceed with the download, please select your preferred development host:

<input type="checkbox"/> Windows Host	<input type="checkbox"/> Linux x86 Host
<input type="checkbox"/> Solaris SPARC Host	<input type="checkbox"/> QNX Neutrino RTOS x86 Host



Die QNX-Download-Webseite.



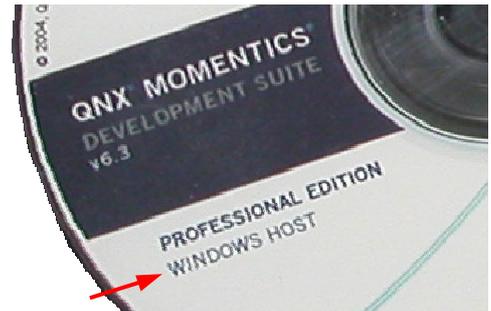
Install- und Release-Notes beachten.

2

Installation von QNX Momentics

QNX Momentics beinhaltet alles, was Sie zur Entwicklung von Programmen für das Betriebssystem QNX Neutrino benötigen : Umfangreiche Integrierte Entwicklungsumgebung (IDE), Compiler, Linker, Libraries und diverse Neutrino-Komponenten, vorkompiliert für alle von QNX Neutrino unterstützten CPU-Architekturen.

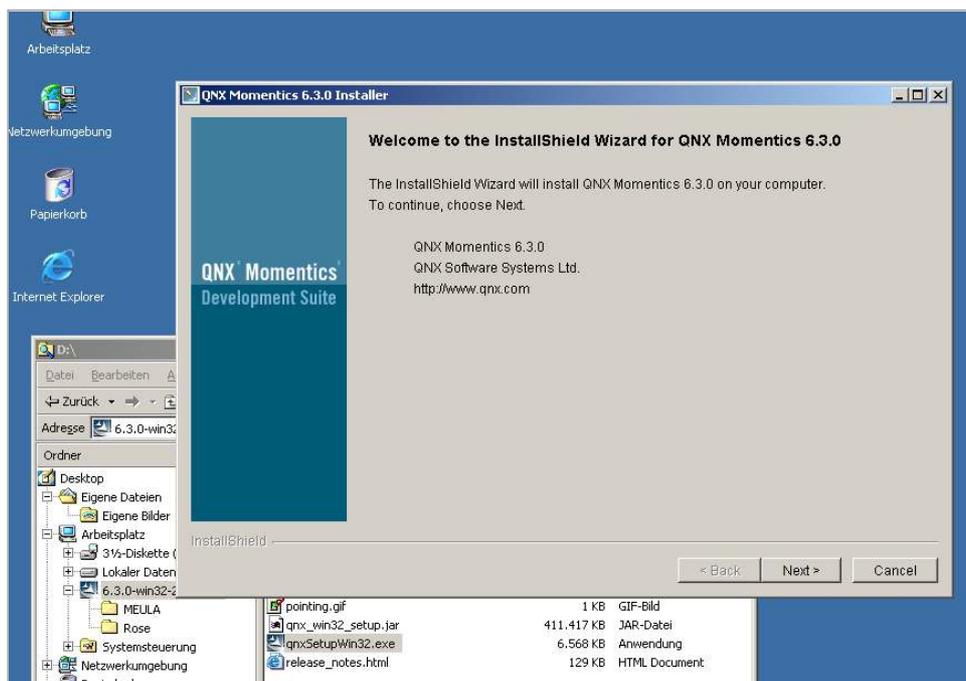
Starten Sie Ihr Windows 2000 oder XP-System und legen Sie die QNX Momentics **Windows** Host CD ein. Sollte der Installationsdialog nicht von selbst starten, führen Sie von der CD das Programm „QNXsetupWin32.exe“ (o.ä.) aus. Sie werden durch den Installationsvorgang geführt.



Das Installierungsprogramm fragt Sie nach einem License Key. Haben Sie sich Momentics als ISO-Image heruntergeladen, ist Ihnen der Key per E-Mail zugegangen. Andernfalls haben Sie ihn beim Kauf Ihres QNX Momentics Development Seat erhalten.

Bei der Frage, welche Komponenten installiert werden sollen, sind defaultmäßig alle angehakt - dies sollten Sie so lassen, denn wer weiß, ob Sie diese nicht noch brauchen?

Nach der Installation finden Sie auf Ihrem Desktop ein Icon für die QNX Momentics IDE sowie ein HTML-File mit weiteren interessanten Informationen zu QNX Momentics und QNX Neutrino.

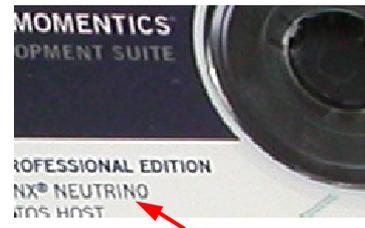


Der Installationsvorgang der QNX-Entwicklungsumgebung unter Windows.

3

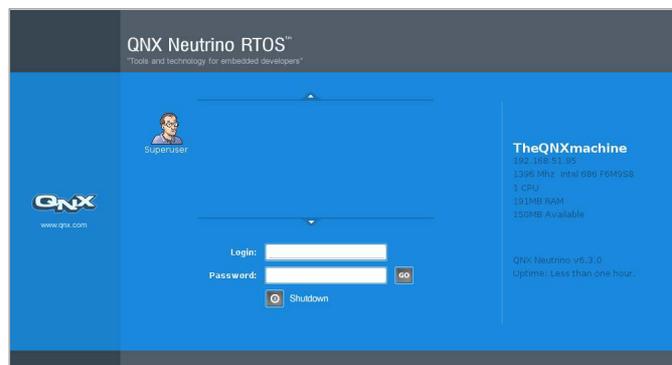
Das Neutrino RTOS installieren

Legen Sie die QNX Momentics **Neutrino** CD in das Laufwerk Ihres zweiten Rechners ein und booten Sie von ihr. Sie können dann wählen, ob Sie (z.B. für erste Tests) das System von CD starten wollen, oder ob Sie auf Festplatte installieren möchten. Wählen Sie die Installation und folgen Sie den Anweisungen auf dem Bildschirm.



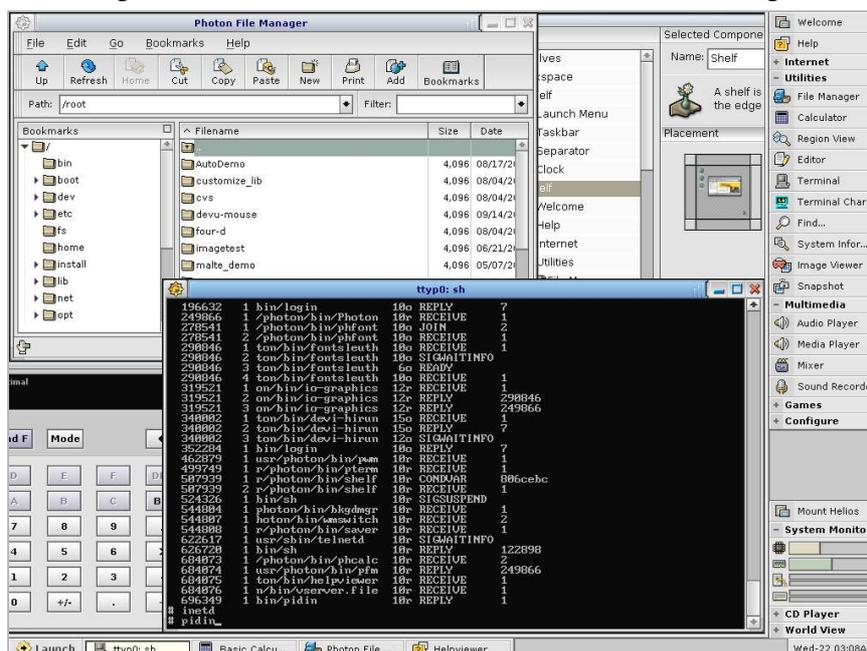
Sie werden während der Installation gefragt, ob Sie auch die Entwicklungsumgebung unter Neutrino installieren wollen. Falls Sie dies bejahen, wird die komplette QNX Momentics Entwicklungsumgebung (inkl. Eclipse-basierter IDE) auch auf Ihrem Neutrino-Rechner installiert. Sie können dann auch dort entwickeln, kompilieren und direkt auf dem selben System testen.

Beim Neustart wird Ihre Hardware automatisch erkannt. Nach Wahl des Grafikmodus können Sie sich als „root“ ohne Passwort anmelden. Rechts auf dem Bildschirm finden Sie eine Icon-Leiste. Von da aus können Sie z.B. mittels „Terminal“ eine Shell öffnen.



Sie haben nun ein vorkonfiguriertes, full-featured QNX Neutrino Echtzeitsystem inkl. Photon GUI vor sich. (Selbstverständlich können Sie Neutrino auch ohne Grafik betreiben.)

Mit dem Befehl `ps` können Sie sich anschauen, welche Prozesse vorhanden sind. Jeder dieser Prozesse ist optional, sprich Sie können ihn später (wenn Sie Ihr Projekt angehen) weglassen, um Ressourcen zu sparen – oder andere Prozesse hinzufügen, um mehr Funktionalität zu erhalten. Das gleiche gilt für Grafik, Netzwerk



oder Audio – bei QNX sind alle Teile einzelne Prozesse, die dynamisch geladen werden können. Mit der QNX Neutrino-Installation, die Sie soeben erstellt haben, können Sie jedoch sehr gut vieles von dem, was QNX bietet, kennenlernen, ohne dass Sie selbst erst ein komplettes Boot-Image konfigurieren und erzeugen müssen.

Großer Funktionsumfang und doch voll echtzeitfähig: Die Neutrino-CD-Installation.

4

Netzwerk unter QNX

Schließen Sie Ihr Zielsystem an das Netzwerk an, mit dem auch Ihr Entwicklungssystem verbunden ist. Haben Sie einen DHCP-Server im Netz, wird Ihr Neutrino-Rechner automatisch eine IP-Adresse bekommen. Diese können Sie mit `ifconfig en0` sehen.



Im Terminalfenster können Sie den Befehl `ifconfig en0 192.168.51.40` (bzw. Ihre gewünschte IP-Adresse)

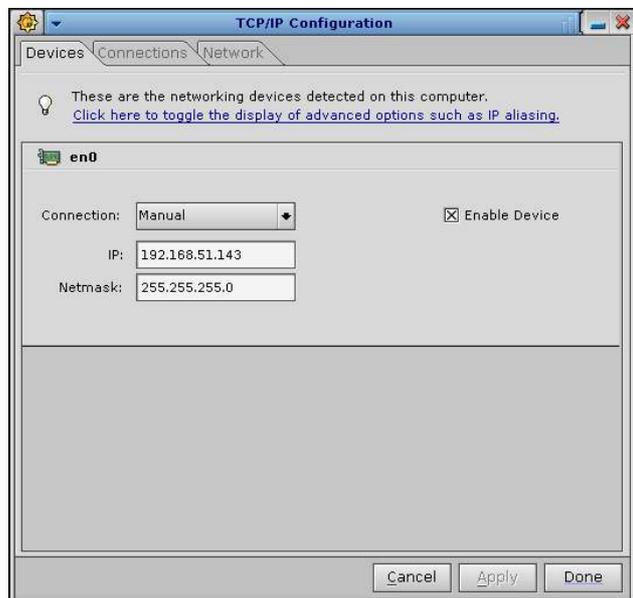
benutzen, um manuell eine IP-Adresse zuzuweisen. Alternativ

können Sie auch das Netzwerk-Konfigurations-

```
ttyp2: sh
# ifconfig en0
en0: flags=8c43<UP, BROADCAST, RUNNING, OACTIVE, SIMPLEX, MULTICAST> mtu 1500
address: 00:0c:29:1d:84:df
inet 192.168.51.143 netmask 0xfffff00 broadcast 192.168.51.255
# -
```

Tool nutzen, welches bei der Neutrino-Installation mit aufgespielt wird. Sie finden es unter dem Punkt „Configure“ auf der Icon-Leiste rechts am Bildschirm oder im „Launch“-Menu.

Hier kann unter **Devices** eingestellt werden, ob DHCP oder manuelle Zuweisung gewünscht ist. Unter **Network** tragen Sie Gateway und DNS ein.



Mittels `ping <IP-Adresse>` können Sie testen, ob sich Ihr Entwicklungsrechner (Host) und Ihr Zielsystem (Target) im Netz gegenseitig erreichen können.

```
# ping 192.168.51.140
PING mmundt.han.qnx.de (192.168.51.140): 56 data bytes
64 bytes from 192.168.51.140: icmp_seq=0 ttl=128 time=1 ms
```

```
C:\>ping 192.168.51.143
Ping wird ausgeführt für 192.168.51.143 mit 32 Bytes Daten:
Antwort von 192.168.51.143: Bytes=32 Zeit<10ms TTL=255
```

Falls das Netzwerk nicht funktioniert, beachten Sie bitte, dass QNX Neutrino nicht jede beliebige Netzwerk-, Grafik- oder Audiokarte unterstützt, die sich auf dem PC-Markt befindet.

Eine Liste der unterstützten Hardware finden Sie unter

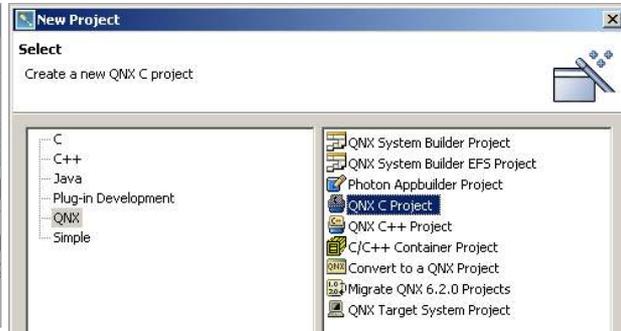
http://www.qnx.com/developers/hardware_support.

Sollten Sie weitere Fragen zu Hardware-Support haben, rufen Sie uns einfach an.

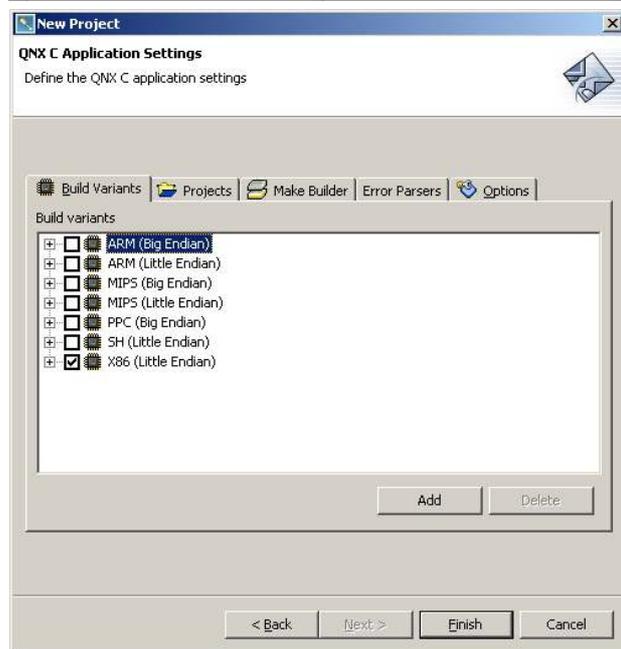
Tip: Die Neutrino-Vollinstallation benutzt automatische Hardware-Erkennung, um die entsprechenden Device-Treiber zu starten. Mit dem Befehl `enum-devices -n` können Sie sehen, welche Hardware die Enumeratoren erkannt haben und welche Treiber daraufhin beim Bootvorgang gestartet wurden.

5

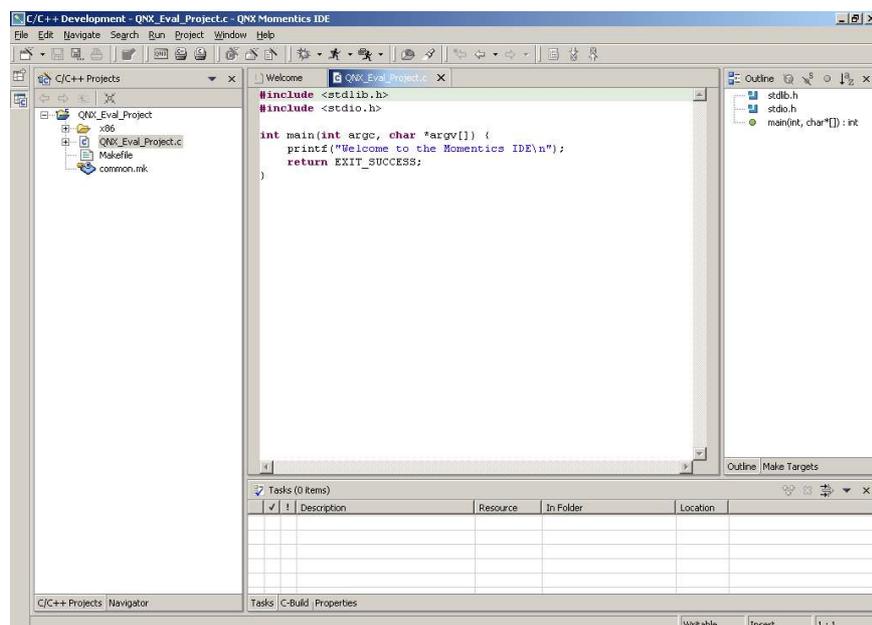
Programm-Projekt anlegen



Starten Sie die QNX Momentics IDE auf Ihrem Entwicklungsrechner. Legen Sie nun ein „QNX C Project“ an: Im Menü **File** wählen Sie **New, Project** aus und im folgenden Dialogfenster selektieren Sie **QNX** und rechts dann **QNX C Project**. Vergeben Sie einen Namen für Ihr Projekt.



Wählen Sie nun im nächsten Schritt aus, für welche CPU-Architektur das Kompilat erzeugt werden soll – für einen PC als Zielsystem ist dies x86. (Für Projekte auf anderen Prozessoren wählen Sie den entsprechenden Prozessortyp - PPC, SH4, ARM oder MIPS). Eine sofort benutzbare Projektstruktur mit Makefile wird aufgebaut, inkl. eines kleinen Programms („Hello World“), das in der automatisch angelegten Source-Code-Datei steht.



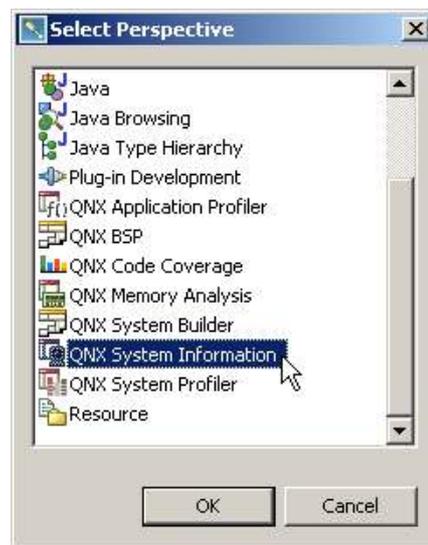
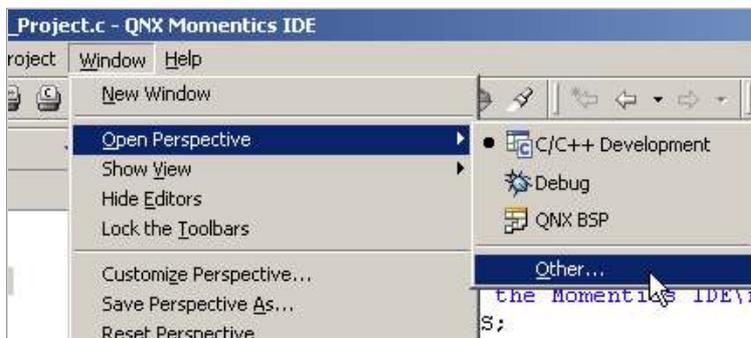
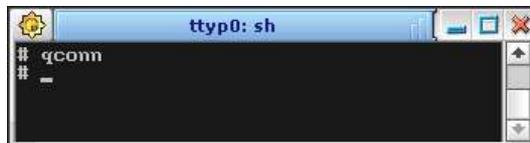
Sie landen nun in der so genannten „C/C++ Perspective“ der IDE mit dem Navigator, dem Editor und einigen anderen nützlichen „Views“ - das sind Bereiche, in denen Informationen angezeigt werden.

Ein QNX C Projekt zeichnet sich durch eine vordefinierte Makefile-Struktur aus.

6

Kommunikation mit Neutrino

Damit das Zielsystem unter QNX Neutrino Anfragen der Entwicklungsumgebung beantworten kann, muss das kleine Programm `qconn` gestartet werden. Auf einem Neutrino-PC können Sie dies einfach von einem Terminalfenster aus tun.



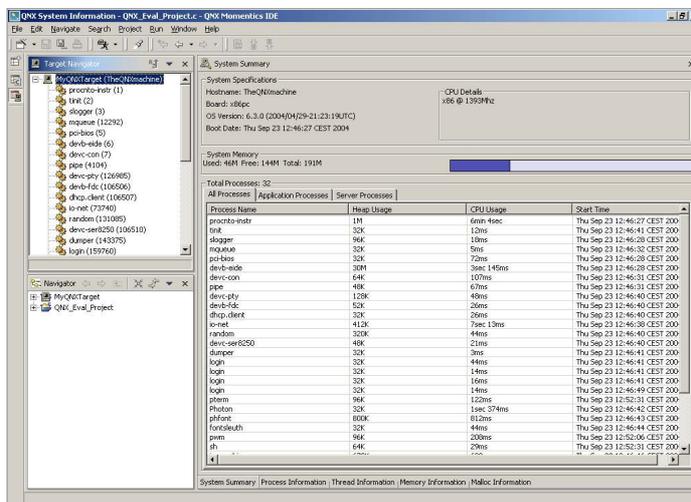
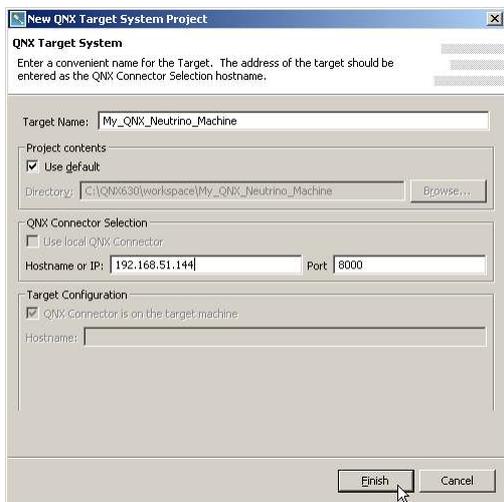
Um von der IDE aus auf Ihr Zielsystem zugreifen zu können, muss ein „Target Project“ angelegt werden.

Wechseln Sie dazu in die „System Information Perspective“: Im Menü **Window** wählen Sie **Open Perspective**, und dort **Other**. Nun selektieren Sie **QNX System Information**.

Im leeren Bereich „Target Navigator“ drücken Sie bitte die rechte Maustaste und wählen aus dem Kontext-Menü **Add New Target**. Vergeben Sie nun einen Namen für Ihr Zielsystem und geben Sie dessen IP-Adresse an.



Sie sehen nun eine Liste aller Prozesse in Ihrem QNX-System. Die sogenannten „Views“ (Tabs unten) geben vielfältige Informationen. Weitere nützliche Views finden Sie im Menü **Window** unter **Show View**.



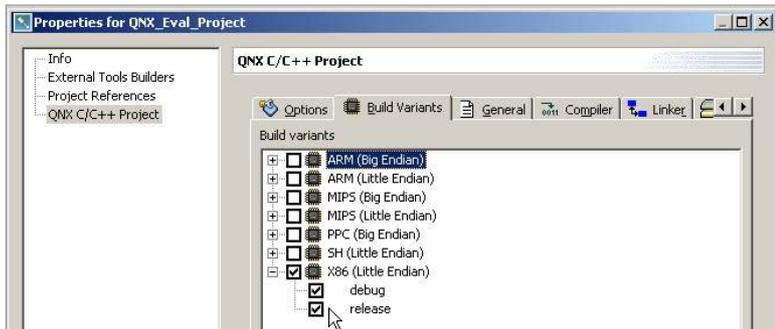
Der Target-Konfigurations-Dialog.

Hier sehen Sie, was sich auf Ihrem Neutrino-System tut.

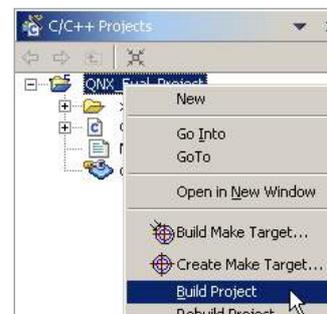
7

Kompilieren und Linken

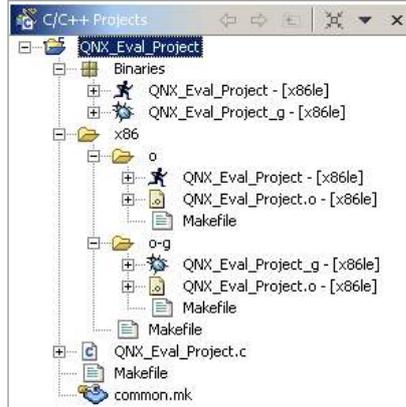
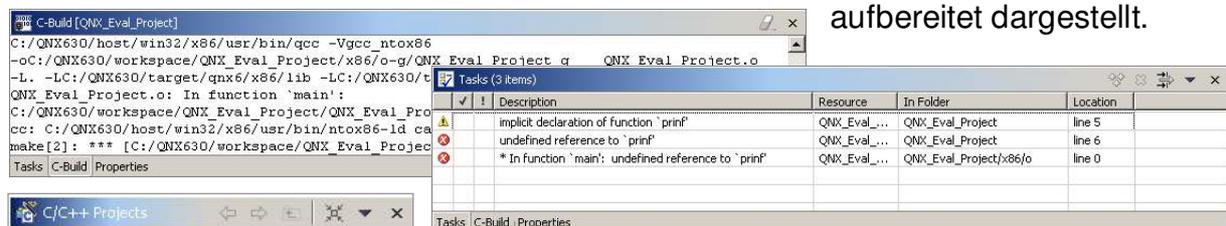
Schalten Sie wieder zurück zur C/C++-Perspektive, in dem Sie das kleine "C"-Icon links am Bildschirm anklicken. Vor dem Kompilieren werden Sie evtl. einstellen wollen, ob mit oder ohne Debug-Informationen kompiliert werden soll. Dazu klicken Sie mit der rechten Maustaste auf den **Projektnamen** und wählen **Properties**. Unter **QNX C/C++ Projects** klicken Sie auf **Build Variants** und dort neben **x86** auf das kleine „+“-Zeichen. Defaultmäßig ist das Kompilieren mit Debug-Information eingestellt. Statt dessen – oder zusätzlich – können Sie hier nun auch ohne Debug-Informationen („**Release**“ Variante) kompilieren.



Beim Anlegen des QNX C Projektes wurde bereits eine von QNX erdachte Directory-Struktur mit Makefiles für Sie erstellt. Um nun ein Kompilat zu erzeugen, klicken Sie mit der rechten Maustaste auf den **Projektnamen** und wählen **Build Project**. Compiler und Linker werden nun ihre Arbeit verrichten.

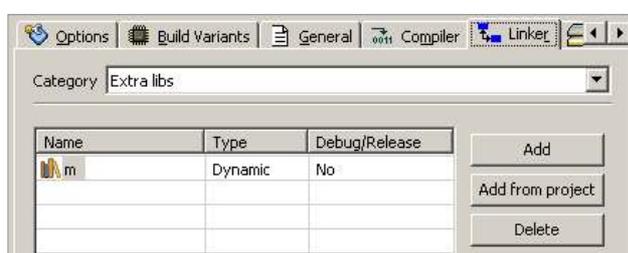


Die Ausgaben des Compilers finden Sie in der sog. „C-Build“ View. Bei Fehlermeldungen werden Sie die „Tasks“ View jedoch als nützlicher empfinden, denn dort wird der Output des Compilers für Sie interpretiert und aufbereitet dargestellt.



Nach dem Build-Vorgang werden Ihre Binaries im „Binaries“-Folder angezeigt. Physisch sind sie im CPU-Directory unter „o“ (für Object) und „o-g“ (-g für Debug-Compileroption) zu finden. Die entsprechenden Makefiles wurden von der IDE automatisch erzeugt.

Die QNX-Library „libc.so“ mit vielen Funktionen wird defaultmäßig dynamisch hinzugelinkt. Wollen Sie später zusätzliche Libraries hinzufügen, so können Sie dies ebenfalls unter den **Project-Properties** tun.



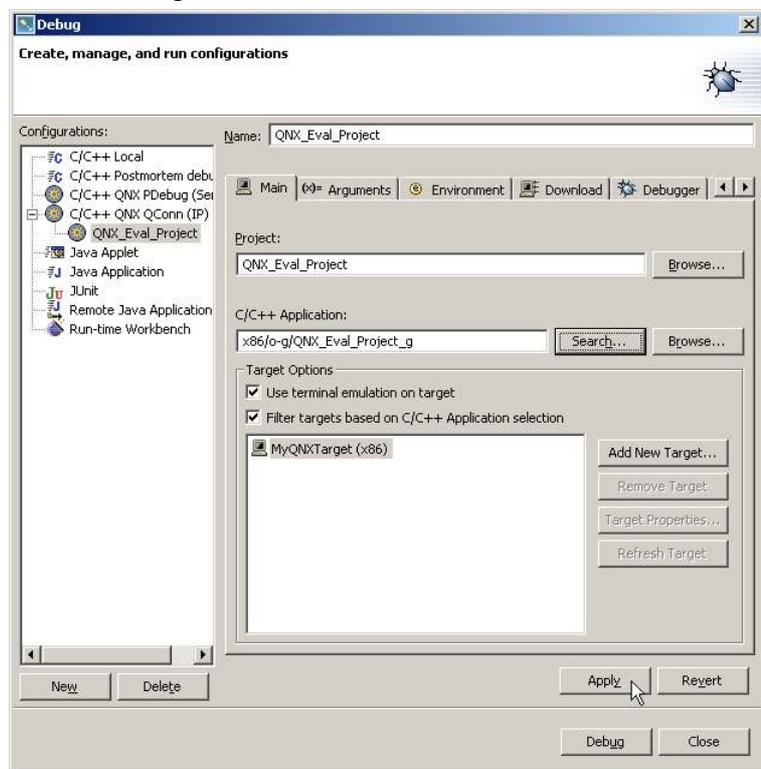
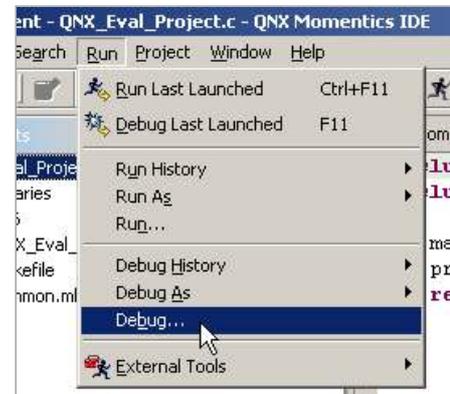
Klicken Sie dort dann auf **Linker** und wechseln Sie unter **Category** zu **Extra Libs**. Die Library, z.B. „libm.so“, geben Sie einfach als „m“ an, also ohne „lib“-Präfix und ohne Endung. Der Linker wird diese dann beim Build-Vorgang hinzulinken.

8

Startvorbereitungen

Um das gerade erzeugte Programm auf Ihrem Zielsystem laufen zu lassen und zu debuggen, benötigen Sie eine sog. „Launch Configuration“. Hier können Sie umfangreiche Einstellungen zum Programmstart (Kommandozeilen-Parameter, Env-Variablen usw.) einmal einstellen und diesen Satz Einstellungen dann immer wieder verwenden.

Legen Sie jetzt eine Launch Configuration an: Im Menü **Run** wählen Sie **Debug...** - es öffnet sich ein Dialogfenster, in dem Sie die Launch Configurations starten, verwalten und verändern können. Wählen Sie nun links den Typ der Launch Configuration: **C/C++ QNX QConn**. Dieser Typ ist für die (Cross-)Entwicklung gedacht, bei der auf der Zielsystemseite QNX mit dem Programm **qconn** läuft. Nun erstellen Sie per Klick auf **New** eine neue Konfiguration. Sie sehen jetzt eine ganze Reihe von Einstellungsmöglichkeiten, die alle mit dem Start Ihres Executables zu tun haben. Defaultmäßig müssen Sie außer unter **Main** nichts konfigurieren, Sie sollten sich jedoch später die angebotenen Möglichkeiten ansehen.



Klicken Sie nun neben dem Feld „C/C++ Application“ den **Search** Button an und wählen Sie Ihr Binary aus. Es trägt als Symbol einen „Käfer“ (=Bug), wenn es mit Debug-Informationen kompiliert wurde, oder einen „Running Man“ für ein normal kompiliertes Binary. Wir wollen im nächsten Schritt den Debugger starten, deshalb wählen Sie bitte jetzt das Binary mit den einkompilierten Debug-Informationen.

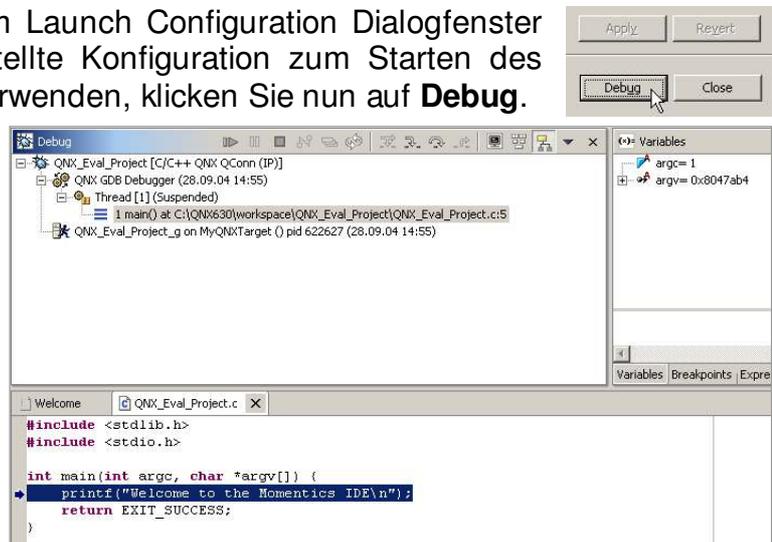
Unter „Target Options“ bitte nun Ihr **Zielsystem** anklicken und auf **Apply** klicken – die Launch Configuration ist bereit.

9

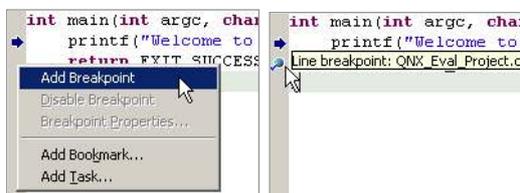
Starten und Debuggen

Sie sollten sich noch immer im Launch Configuration Dialogfenster befinden. Um die gerade erstellte Konfiguration zum Starten des Programms im Debugger zu verwenden, klicken Sie nun auf **Debug**.

Die IDE wechselt in die „Debug Perspective“, transferiert Ihr Programm über das Netzwerk von Ihrem Entwicklungs-PC auf Ihr Neutrino-System und startet es im Debugger. Sie sehen, wie der Debugger in der ersten Zeile Ihres Programms anhält. Im „Debug“-Bereich erhalten Sie einen Überblick über Ihren Prozess und den Call-Stack. Mit den Buttons in der Leiste der Debug-View steuern Sie den Debugger.



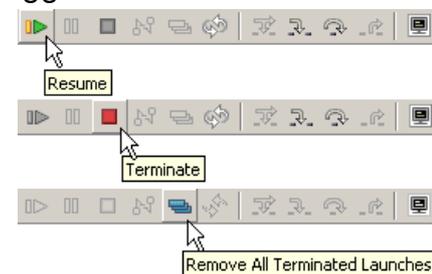
Mit dem **Step Over**-Button können Sie zur nächsten Zeile springen. Beim Debuggen Ihrer Programme werden Sie dann rechts in der „Variables“-View beobachten, wie sich Ihre Variablen verändern. Der **Step Into** Button wird benutzt, um mit dem Debugger in Funktionen hineinzugehen (wenn Sie den Source zu der Funktion haben).



Wollen Sie einen Breakpoint setzen, rufen Sie am linken Rand der Source-Anzeige das Kontextmenü auf und wählen **Add Breakpoint**. Er erscheint als kleine „Stecknadel“ und kann auch beim Programmieren gesetzt/entfernt werden.

Trifft das Programm auf den Breakpoint, hält der Debugger es an und Sie können z.B. Variablen in Augenschein nehmen. Bei Klick auf den **Resume**-Button läuft Ihr Programm weiter bis zum nächsten Breakpoint.

Um das Programm (vorzeitig) zu beenden, klicken Sie den **Terminate**-Button. Nach Programmende können Sie mittels **Remove All Terminated Launches** den Debugger beenden und das vorher übertragene Binary auf dem Zielsystem löschen lassen.



Um Ihr Programm einfach so (ohne Debugger) laufen zu lassen, klicken Sie im Menü **Run** statt auf **Debug...** bitte auf **Run...** – dann können Sie die im vorigen Schritt beschriebene Launch Configuration einfach für das Starten verwenden oder eine neue erstellen, bei der Sie das Binary ohne einkompilierte Debug-Informationen auswählen. Natürlich können Sie Ihr Programm auch in der System Information Perspective mit dem File System Navigator (Menü **Window, Show View**) übertragen und direkt vom Terminal-Fenster aus starten. Netzwerkfreigaben mounten ist ebenfalls möglich (mehr dazu in der Dokumentation unter **fs-cifs**).

10

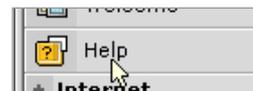
Ihr Programm unter QNX Neutrino

Damit das ganze nun Ihr persönliches erstes QNX-Programm wird, werden Sie Modifikationen und Erweiterungen am gerade erstellten Source-Code vornehmen wollen. Am besten, Sie probieren ein paar unserer Beispiel-Programme aus und übernehmen Code daraus in Ihr Projekt. Außerdem wollen Sie nun sicher eine ganze Menge wissen, z.B. wie Sie eigene Threads erzeugen, wie das QNX Neutrino Message Passing funktioniert, welche Prozess-Synchronisations-Methoden es gibt, wie Sie Zugriff auf I/O-Bereiche bekommen oder wie ein QNX Resource Manager aufgebaut ist. Doch keine Sorge: Es geht alles (fast) so einfach wie Ihr gerade vollzogener Start!



Die Einführung **Welcome to QNX Momentics** gibt Ihnen einen guten Überblick, was es wo gibt. Wir empfehlen Ihnen: Schauen Sie sich unbedingt den **Neutrino System Architecture Guide**, den **IDE User's Guide** und den **Neutrino Programmer's Guide** an. Sämtliche Dokumentation finden Sie im Menü **Help** unter **Help Contents**. Wählen Sie im Helpviewer der IDE **QNX**

Momentics Professional Edition aus. Die Hilfe ist auch unter Neutrino selbst verfügbar: Auf der Iconleiste rechts finden Sie oben den Button **Help**. Nutzen Sie auch die Suchfunktion!



Source-Code Beispiele für Thread-Erzeugung, Mutex-Benutzung, Message Passing und andere Inter-Prozess-Kommunikation sowie QNX Resource Manager finden Sie auf unserem FTP-Server *ftp.qnx.de* – einfach als User *qnx* mit Kennwort *qnx!neutrino* verbinden und das Source-Paket herunterladen. Die dort enthaltene ReadMe-Datei erklärt, wie sie diese Source-Beispiele als Projekte in die QNX Momentics IDE importieren. Der Source ist ausführlich kommentiert. Zusätzlich empfiehlt sich, zu jeder Funktion, die Sie interessiert, die Doku im Helpviewer anzuschauen.

Sicher werden Sie bei der Entdeckung der QNX Momentics Tools und dem QNX Neutrino RTOS noch weitere Fragen haben. Wenden Sie sich bitte an Ihren QNX Account Manager, Field Application Engineer oder unsere Support-Abteilung. Wir möchten Sie von Anfang an begleiten, denn Ihr Erfolg ist auch unser Erfolg!

QNX Software Systems GmbH
Am Listholze 76
30177 Hannover
Tel. 0511 / 94091-200
E-Mail: info@qnx.de

