

10ステップで初めての QNX プログラム作成

クイックスタート ガイド

第3版

QNX クイックスタート ガイド

本書では、QNX Momentics ツールと QNX Neutrino オペレーティング システムのインストールと設定方法を解説します。簡単ガイドで、すぐに開発をスタートできます。

- 1 システム要件
- 2 QNX Momentics 開発スイートのインストール
- 3 QNX Neutrino RTOS のインストール
- 4 QNX Neutrino でのネットワーキング
- 5 プログラム プロジェクトの作成
- 6 QNX Neutrino との通信
- 7 コンパイルとリンク
- 8 プログラム起動の準備
- 9 起動とデバッグ
- 10 ユーザー プログラムの作成

1

システム要件



QNX® Neutrino® リアルタイム オペレーティング システム (RTOS) で実行するプログラムを作成するには、まず、QNX Momentics® 開発キットが必要です。この開発キットは、Windows Vista、Windows 2000、Windows XP、Linux にインストールすることができます。または、セルフホストの QNX Neutrino システムとして使用することができます。

QNX Momentics CD をお持ちでない場合は、評価版を <http://www.qnx.co.jp/products/eval> からダウンロードすることができます。x86 ターゲットでのみ QNX Neutrino の評価を希望する場合は、x86 ターゲット用 QNX Momentics 開発キット (より小さなサイズ) のダウンロードが可能です。ホストごとに別個の CD がありますのでご注意ください。

開発システムのほかに、QNX Neutrino RTOS と、これから開発するプログラムを実行するターゲットプラットフォームが必要となります。

QNX Neutrino に慣れるには、いくつかのやり方があります。

- ▶ およそ 2.5 GB の空きパーティションがある普通の PC に QNX Neutrino ホストバージョンの開発キットをインストールします。(OS そのものは、およそ 300 MB しかなく、QNX Neutrino x86 ランタイム キットをインストールすると取得できます。) QNX Neutrino をインストールしても、既存のパーティションがダメージを受けることはありません。ハードディスクに十分なスペースが無い場合は、CD から直接 QNX Neutrino を起動することもできます。
- ▶ CPU ベンダが提供するリファレンス設計用のリファレンス プラットフォーム (例: PPC、XScale、SH4 CPU など) で QNX Neutrino を実行します。この場合、該当プラットフォームに対応する QNX ボード サポート パッケージ (BSP) が必要になります。BSP に添付の資料に、ターゲットシステムへ QNX Neutrino をインストールする方法が記載してあります。
- ▶ QNX Neutrino を VMware セッションでバーチャル マシンとして実行します。ただし、VMware は QNX Neutrino 評価の手段としては便利ですが、バーチャル マシンではハード リアルタイムに対応できるとは限らないのでご注意ください。



インストール ガイドをよくお読みください。

QNX Neutrino は、すべてのプラットフォームで同様に設計されています。このガイドでは、PC へのインストールを例に解説します。

QNX Neutrino ホスト CD をお持ちでない場合は、<http://www.qnx.co.jp/products/eval> から評価版をダウンロードすることができます。

2

QNX Momentics 開発スイートのインストール

QNX Momentics には、QNX Neutrino RTOS プログラムを記述するために必要なすべての要素が揃っています。総合的な統合開発環境 (IDE)、コンパイラ、リンカ、ライブラリ、およびその他の QNX Neutrino コンポーネントが、すべての対応 CPU アーキテクチャ用にコンパイルされた形で含まれています。



ご使用の開発ホストで Windows ホスト CD をご使用ください。

Windows Vista、2000 または XP システムを立ち上げ、**QNX Momentics Windows ホスト CD** を挿入します。自動的にインストールが開始されない場合は、CD から "**qnxSetupWin32.exe**" を実行します。画面の指示に従ってインストールを進めてください。詳しくはインストール ガイドをお読みください。



Windows での QNX 開発環境インストール

インストール中に、ライセンス キーの入力を求められます。QNX Momentics の評価版を QNX ウェブサイトからダウンロードした場合、電子メールにてお知らせしたライセンス キー、あるいは、QNX Momentics 開発ライセンスを購入した際に取得されたライセンス キーを入力してください。

インストールが完了すると、Windows デスクトップに QNX Momentics IDE のアイコンが表示されます。



3

QNX Neutrino RTOS のインストール



ターゲット上の QNX Neutrino ホスト CD を使用します。

次に、評価用マシンのドライブに **QNX Momentics Neutrino** ホスト CD を挿入し、CD から起動します。この時点で、OS を CD から直接スタートするか、ハードディスクに QNX Neutrino をインストールするか選択することができます（初回のテストでは、CD からの直接起動が良いでしょう）。ハード ディスクへのインストールを選択し、画面の指示に従ってください。詳細については、インストール ガイドをお読みください。

再起動を行うと、自動的にハードウェアが検出されます。グラフィックス モードを選択すると、パスワードを必要としない **root** としてログインすることができます。



これで、事前定義が済んだ状態のフル機能 QNX Neutrino リアルタイム オペレーティング システムを、QNX Photon® microGUI® も含めて立ち上げた状態になります（もちろん、グラフィックス抜きで QNX Neutrino を実行することも可能です）。

4

QNX Neutrino でのネットワークング

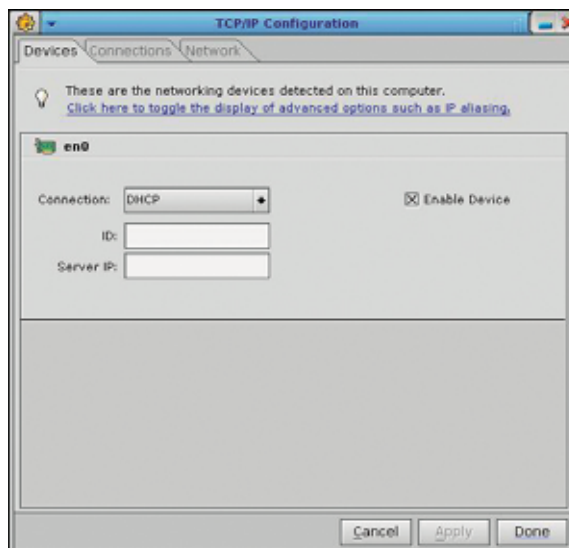


次に、QNX Neutrino マシン（ターゲット マシン）をネットワークにつなげてみましょう。このとき、開発マシンも同じネットワークに接続します。DHCP サーバにより、QNX Neutrino マシンは、自動的に IP アドレスを受け取ります。IP アドレスは、`ifconfig en0` コマンドで確認することができます。

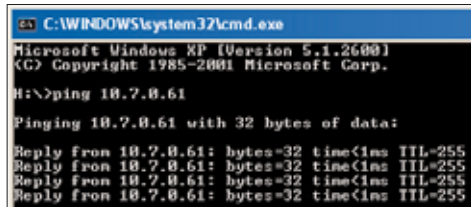
```
ttyn0: sh
# ifconfig en0
en0: flags=0043<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    address: 00:e0:29:82:13:c6
    inet 10.7.0.61 netmask 0xffff0000 broadcast 10.255.255.255
# _
```

手動で IP アドレスを割り当てるには、端末ウィンドウでコマンド `ifconfig en0 IP_address` を使用します（つまり、自分のシステムで使用したい IP アドレスを `IP_address` に指定）。

または、QNX Neutrino についてくるネットワーク設定ツールを使用することもできます。このツールは、画面右側のアイコンバーの **[Configure]**、または、**[Launch]** メニューから見つけることができます。**[Devices]** で、DHCP を使用するか、手動で IP アドレスを割り当てるか選択することができます。**[Network]** タブでは、ゲートウェイの IP アドレスと DNS を入力します。



`ping IP_address` コマンドを使用すると、開発マシン（ホスト）がネットワーク上の QNX Neutrino システム（ターゲット）と接続されているか、またターゲットがホストに

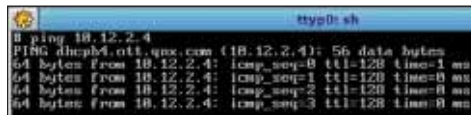


```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
H:\>ping 10.7.0.61

Pinging 10.7.0.61 with 32 bytes of data:

Reply from 10.7.0.61: bytes=32 time<1ms TTL=255
Reply from 10.7.0.61: bytes=32 time<1ms TTL=255
Reply from 10.7.0.61: bytes=32 time<1ms TTL=255
Reply from 10.7.0.61: bytes=32 time<1ms TTL=255
```

接続されているかをテストできます。



```
ttty0: sh
# ping 10.12.2.4
PING dhrphd4.cdt.qnx.com (10.12.2.4): 56 data bytes
64 bytes from 10.12.2.4: icmp_seq=0 ttl=128 time=1 ms
64 bytes from 10.12.2.4: icmp_seq=1 ttl=128 time=0 ms
64 bytes from 10.12.2.4: icmp_seq=2 ttl=128 time=0 ms
64 bytes from 10.12.2.4: icmp_seq=3 ttl=128 time=0 ms
```

注意：もしホスト マシンがファイアウォールを使用している場合には、ターゲットから `ping` コマンドを使用できないことがあります。Windows XP SP2 上では、ICMP 設定中の [エコー要求の着信を許可する] を有効にしなければならない場合があります。スタート メニューからコントロール パネルを選択、さらにセキュリティ センターを選択し、詳細設定を行います。

ネットワークが正常に作動しない場合、サポートされていないネットワーク カードを使用していることが考えられます。対応ハードウェアの完全リストについては、http://www.qnx.com/developers/hardware_support をご覧ください。ハードウェア サポートについて、その他ご質問がある場合は、弊社営業担当者までご連絡ください。

ヒント：Neutrino の完全インストール版では、対応するデバイス ドライバをスタートするために自動ハードウェア検出を行います。`enum-devices -n` コマンドを使用すると、どのハードウェアが検出され、ブート時にどのドライバが起動されたか見ることができます。

5

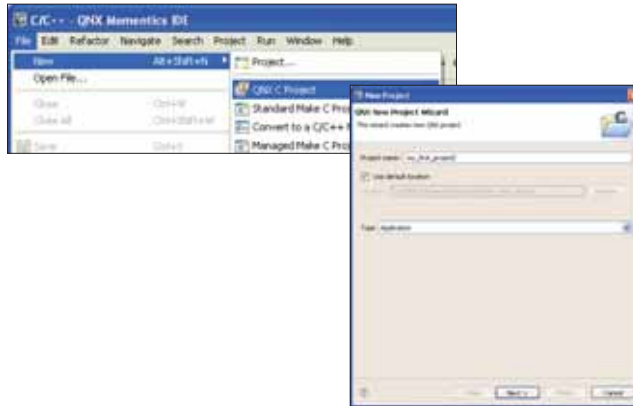
プログラム プロジェクトの作成

開発ホストで QNX Momentics IDE をスタートしてみましょう。IDE の初回起動時に、プロジェクトその他のファイルを格納するフォルダ、*workspace* を選択するよう促されます。その後、IDE のウェルカム ページが表示されます。準備ができれば、Workbench アイコンをクリックしてください。



次に、QNX C プロジェクトを作成します。

[File]メニューから[New] > [QNX C Project] を選択してください。



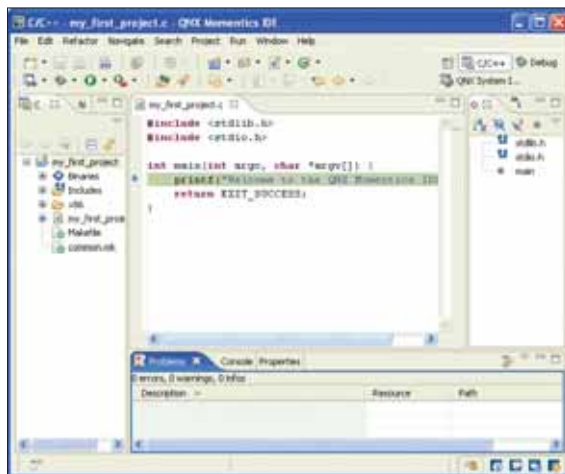
次に表示されるダイアログでプロジェクトに名前をつけ、[Next] をクリックしてください。次に、[Build Variants] タブから、作成するバイナリを実行する CPU アーキテクチャを選択します。PC がターゲットの場合は、x86 を選択します。その他のプロセッサを対象としたプロジェクトでは、それぞれ対応する CPU タイプ：PPC、SH、ARM、MIPS を選択してください。また、コンパイル時のデバッグ情報の有無を選択することもできます。後で両方の場合を見ていきますので、debug および release の両方をチェックしてください。



ビルドの種類を選択

[Finish]をクリックします。Makefileを含む、すぐに使用できるプロジェクト構造が作成されます。これには、小さなプログラム ("Welcome to the QNX Momentics IDE") が含まれており、自動的に生成されるソースコード ファイルの中にあります。

この時点で、IDEの「C/C++ Perspective」が表示されているはずですが、ここでは、ナビゲータ、エディタ、および作業に関連するさまざまな情報を表示できる複数の「ビュー」があります。



定義済み Makefile 構造を備えた QNX C プロジェクト

6

QNX Neutrino との通信

ターゲットシステムは、開発環境からのリクエストに回答しなければなりません。これを可能にするため、**qconn** というプログラムを開始します。これは、QNX Neutrino を実行している PC の端末ウィンドウで行います。



IDE からターゲットシステムにアクセスするには、ターゲット プロジェクトを作成します。これを行うには、**[System Information]** パースペクティブを使います。まず、**[Window]** メニューから **[Open Perspective]** を選択し、**[Other]** をクリックします。ここで、**[QNX System Information]** を選択し、**[OK]** を選択します。

[Target Navigator] の空白エリアで、マウスを右クリックし、コンテキストメニューから **[Add New Target]** を選択します。

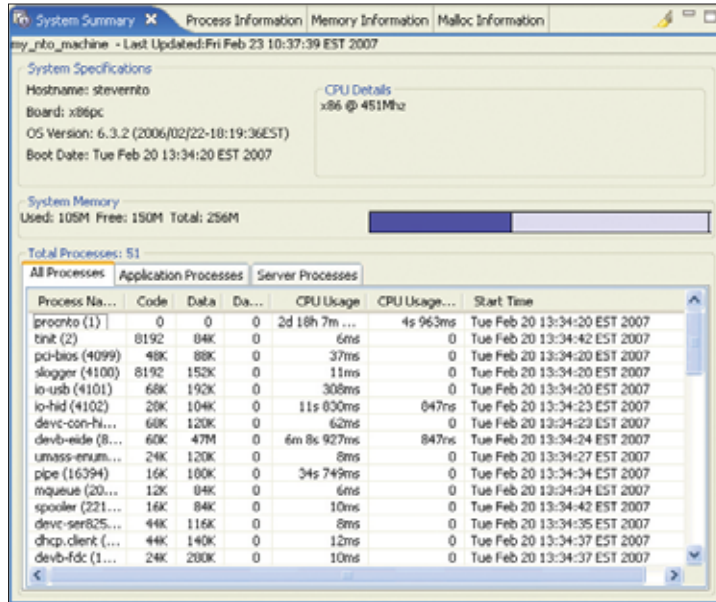


ここで、ターゲットシステムの名前と IP アドレスを対応するフィールドに入力します。



ターゲット設定ダイアログ

[Finish] をクリックし、[Target Navigator] で新しいターゲットを選択してください。これで、QNX Neutrino システムにあるすべてのプロセスを見ることができるようになります。ビュー（画面上部にある複数のタブ）でさまざまな情報を表示することができます。[Window] メニューから [Show View] を選択すると、さらに多くの便利なビューを見ることが出来ます。

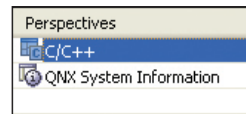


QNX Neutrino システム内で何が起きているかを表示

7

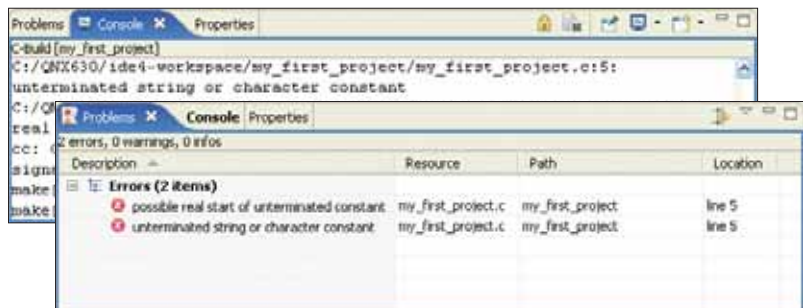
コンパイルとリンク

次に、以下のいずれかの手順で、C/C++ パースペクティブに戻ります。まずは、画面右上のドロップダウン メニューから「C/C++ パースペクティブ」を選択するという方法。あるいは、コントロール キーを押したまま、ポップアップ メニューに[C/C++] が強調表示されるまで [F8] キーを繰り返し押し、それから両方のキーを放すという方法。あるいは、[Window] > [Navigation] > [Next Perspective] を選択し、ポップアップ メニューから [C/C++] を選択するという方法もあります。



コンパイルを開始する前に、デバッグ情報を含めてコンパイルするかどうかを選択すると良いでしょう。これを行うには、プロジェクト名を右クリックし、[Properties] を選択します。[QNX C/C++ Projects] をクリックし、[Build Variants] をクリック、そして [x86] の横にある小さな [+] 記号をクリックします。デバッグとリリースの両方の変種をチェックしてください。[OK] をクリックします。プロジェクトを再コンパイルするが尋ねられます。QNX C プロジェクトの作成をおこなった際に、**Makefile** を含む QNX 製ディレクトリ構造が生成されています。バイナリを作成するには、プロジェクト名を右クリックし、[Build Project] を選択します。これで、コンパイラとリンクが作業を開始します。

エラーも含め、コンパイラの出力が [Console] ビューの中の [C-Build] 出力に表示されます（エラーは発生しないはずですが、下の例ではエラーを追加しています。）しかし、コンパイルの途中でエラーが発生する場合、[Problems] ビューの方が便利です。このビューは、[Console] ビューに比べ、コンパイラの出力を解釈された読みやすい形で表示するからです。[Editor] ビューも、ポインタをその上に置いたままにすると、エラーについての情報を示します。



ビルド操作の完了後、生成されたバイナリが **[Binaries]** フォルダに表示されます。物理的には、これらのファイルは CPU ディレクトリの **o** (object の o) および **o-g** (**-g** はコンパイラに渡されたデバッグ オプション) 以下にあります。対応する **Makefile** 群は、IDE によって自動的に生成されています。



基本的な関数を多く含んでいる QNX ライブラリの **libc.so** は、デフォルトでバイナリに動的にリンクされます。後で別のライブラリを追加したい場合は、**[Project]** の **[Properties]** セクションで行うことができます。ここで、**[Linker]** をクリックし、**[Category]** フィールドから **[Extra Libs]** を選択します。



[Add] をクリックし、**Lib** プレフィックスや拡張子なしに、ライブラリの名前を入力します。例えば、算術演算ライブラリの **libm.so** を追加するためには、**[Name]** のフィールドに **"m"** とタイプします。



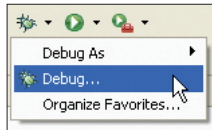
[OK] をクリックします。これで、プロジェクトのビルド時に、リンカがライブラリにリンクします。

8

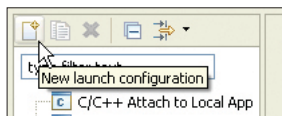
プログラム起動の準備

新しくビルドしたプログラムをターゲットシステムで実行しデバッグするには、「起動設定 (Launch Configuration)」を行う必要があります。起動設定は、プログラムの起動に影響する様々な設定を構成するものです (コマンドラインのパラメータ、環境変数など)。これらの設定事項を一度入力すると、後から何度でも使用することができます。

それでは、自分の起動設定を用意してみましょう。まず、ツールバーの「虫 (バグ)」アイコン横のドロップダウンメニューから、**[Debug...]** を選択します。



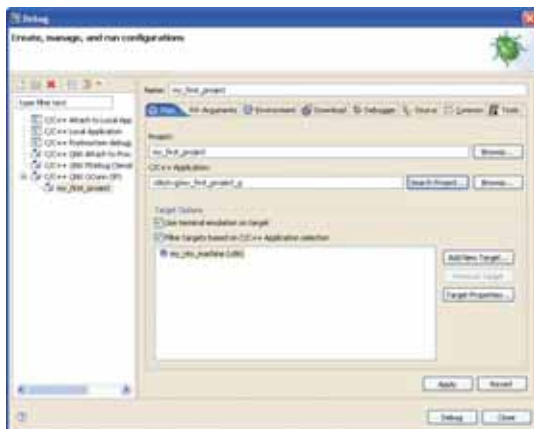
ウィンドウが開き、ここで、既存の起動設定のスタート、変更、あるいは新規作成ができます。ダイアログウィンドウの左側で、起動設定のタイプ **[C/C++ QNX QConn (IP)]** を選択します。このタイプは、**qconn** プログラムを使用してターゲットシステムで QNX Neutrino を実行するネットワークベースの (クロス) 開発用です。次に、**[New Launch configuration]** をクリックして、新しい起動設定を作成します。



実行プログラムの起動を制御するさまざまな設定オプションが表示されます。デフォルトでは、**[Main]** タブのみ入力が必要となりますが、後から他のタブの設定も試してみてください。

[C/C++ Application] フィールドの横にある **[Search]** ボタンを押し、バイナリを選択します。デバッグ情報をつけてコンパイルしたバイナリの場合は、名前に **_g** が含まれるようになります。デバッグ情報をつけずにコンパイルされたバイナリは、**_g** が含まれない名前になります。次のステップではデバッガ起動の練習を行うので、ここではデバッグ情報がコンパイル時に含まれたバイナリを選択し、**[OK]** をクリックします。

[Target Options] でターゲットシステムが表示されていることを確認し、**[Apply]** をクリックします。これで起動設定の準備が完了しました。



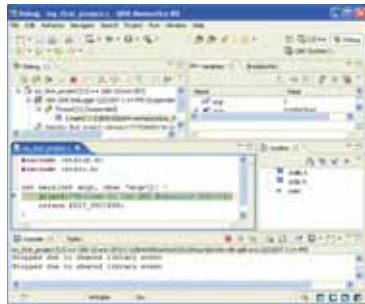
9

起動とデバッグ



引き続き、このステップは起動設定ダイアログから始めます。プログラムの起動設定を行ったので、これをデバッガでスタートします。**[Debug]** をクリックしてください。

これで、IDE がデバッグ パースペクティブに切り替わります。ユーザーのプログラムは、ネットワークを介して開発マシンからターゲットとする QNX Neutrino システムに転送され、デバッガで起動されます。デバッガは、プログラムの最初の行でストップします。**[Debug]** ビューで、コール スタックを含むプロセスの概要を見ることができます。**[Debug]** ビューのメイン バーにあるボタンを使用して、デバッガを制御することができます。



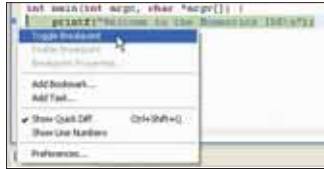
IDE からアプリケーションを実行またはデバッグすると、IDE のコンソールにすべての入力を読み込まれ、また出力はここに表示されます。実行の流れが `printf()` コールのラインを通過すると、コンソール ウィンドウに "Welcome to QNX Momentics" というメッセージが表示されます。

[Step Over] ボタンを使用すると、コードの次の行にジャンプします。



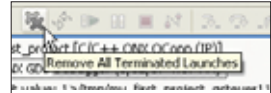
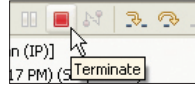
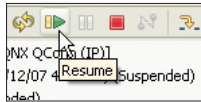
デバッグ中は、変数の変更を表示する右側の **[Variables]** ビューに注意します。**[Step Into]** ボタンは、デバッガをコードの関数部分に進めます（当然ながら、この関数のソース コードがある場合のみ役に立つ機能です）。

ブレークポイントを設定するには、ソース表示枠の左側にマウスのポインタを合わせ、右クリックしてコンテキストメニューから **[Toggle Breakpoint]** を選択します。ブレークポイントはチェックマークがついた丸で表され、コードの記述をしながらの設定と削除が可能です。



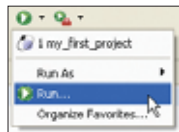
ブレークポイントの設定

実行中のプログラムがブレークポイントに達すると、デバッガで停止するので、変数を調べるなどの作業を行うことができます。**[Resume]** ボタンをクリックすると、プログラムは次のブレークポイントまで実行を再開します。プログラムの実行を中止するには、**[Terminate]** ボタンを使用します。プログラムの実行が完了したら、**[Remove All Terminated Launches]** ボタンを押してデバッガを終了し、バイナリ（以前に転送されたもの）をターゲットシステムから削除します。



注意：デバッガは、プログラムの実行中、プロジェクトのファイルを開いたままにします。プロジェクトを再ビルドする前に、デバッガのセッションを終了してください。これを怠ると、ビルドが失敗します。

プログラムをスタンドアロンのバイナリとして実行するには（つまり、デバッガを使用しない）、[Run] メニューから、[Debug...] の代わりに **[Run...]** をクリックします。



これで、起動設定を使用して、プログラムを起動することができます（前のステップで説明）。あるいは、新規の起動設定を作成し、デバッグ情報なしのバイナリを選択することもできます。[System Information Perspective] の [File System Navigator] ([Window] メニューの [Show] ビュー) を使用して、バイナリを手動で転送し、端末から起動することもできます。共有ネットワーク ドライブのマウントも可能です (QNX Neutrino ユーティリティ リファレンス マニュアルの **fs-cifs** の項を参照)。

10

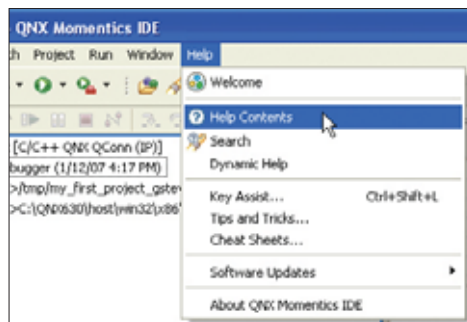
ユーザープログラムの作成

ここまでの作業をまとめ、はじめての QNX プログラムを完成させるには、作成したソースコードに独自の変更と拡張を加えます。練習として、提供されているサンプルプログラムから、自分のプロジェクトにコードを取り入れてみてください。この時点で、スレッド作成や、QNX メッセージパッシングのメカニズム、プロセス同期の方法、I/O エリアへのアクセス方法、QNX Neutrino リソースマネージャの記述方法など、さまざまな情報についてもっと知りたいと思うでしょう。これらはすべて、このクイックガイドで説明した内容と（ほぼ）同じくらいシンプルです！

IDE には、QNX に慣れるためのチュートリアルが含まれています。IDE のツールバーから **[Help] > [Welcome]** を選択し、チュートリアルアイコンをクリックします。



IDE の Help システムには、QNX マニュアルと Eclipse プラットフォームに関する情報が含まれています。**[Help]** メニューで **[Help Contents]** をクリックします。



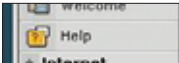
必要な情報を探すには、*Welcome to QNX Momentics* を参照してください。QNX Neutrino System Architecture Guide、IDE User's Guide、QNX Neutrino Programmer's Guide の各マニュアルも一読をお勧めします。

[Help] ウィンドウから以下を選択します。

- ▶ QNX Momentics、および QNX についての情報は、[A Roadmap to the QNX Momentics Development Suite] を選択します。
- ▶ IDE の使用方法については、[IDE User's Guide] を選択します。



セルフホスト Neutrino システムでマニュアルを参照するには、画面右側アイコンバーにある [Help] ボタンをクリックします。



スレッド生成、ミューテックスの使用法、メッセージ パッシングその他プロセス間通信のメソッドをカバーするサンプル ソースコード、および QNX リソース マネージャのテンプレートを、弊社 FTP サーバー ftp.qnx.de から入手することができます。FTP クライアントを使用し、ユーザー **qnx** としてログインの上、ソース パッケージをダウンロードしてください。パスワードは **qnx!neutrino** です。このパッケージの中には、**readme.txt** が含まれており、ソース サンプルを QNX Momentics IDE にインポートする方法を解説しています。ソースには、詳しいコメントが含まれています。関心のある関数については、QNX Neutrino Library Reference も参照してください。

お問い合わせ先

QNX では、導入初期段階からお客様とのコミュニケーションを大切にしています。
QNX Momentics ツールと QNX Neutrino RTOS についてのご質問は、QNX 営業担当者、フィールドアプリケーションエンジニア、またはサポート担当までお寄せください。

QNX ソフトウェア システムズ株式会社

ウェブサイト：www.qnx.co.jp

メール：japan_info@qnx.com

QNX ソフトウェア システムズ株式会社

〒102-0075 東京都千代田区三番町六番地
三番町 KB-6 ビル 四階

Tel: 03-3511-6450

Fax: 03-3511-6451

Email: japan_info@qnx.com

Web: www.qnx.co.jp

QNX Software Systems (本社)

175 Terence Matthews Crescent
Ottawa, Ontario

Canada, K2M 1W8

Tel: 1-613-591-0931

Fax: 1-613-591-3579

Email: info@qnx.com

Web: www.qnx.com



QNX SOFTWARE SYSTEMS