QNX 6.6.0 x86 ターゲット　ランタイム環境構築ノート

QNX Software Systems Japan
ver 1.02　2014/8/14

1. はじめに

ターゲットマシンに QNX Neutrino ランタイム環境をインストールするには、ホスト環境で起動イメージを作成し、必要なファイルと共にターゲットマシンのマス・ストレージに書き込む必要があります。

ターゲットマシンの CPU が x86 の場合は BIOS を持った PC/AT アーキテクチャのハードウエアであれば、QNX SDP 6.5.0 までは QNX Neurtino ホスト環境を利用する事により、比較的容易にランタイム環境を作成する事ができました。

しかし、QNX SDP 6.6.0 からは QNX Neutrino ホスト環境が廃止されましたので、x86 ターゲットの場合でも、基本的には Windows ホスト環境もしくは Linux ホスト環境のみを使用してランタイム環境を作成する必要があります。

ただし、QNX SDP 6.6.0 の Addtional Package として VMware Player 等で動作可能な QNX Neutrino ランタイム環境が提供されていますので、これを QNX Neutrino ホスト環境の代用とすることで、カスタムのランタイム環境の作成が容易となります。

本ノートでは BIOS を持った x86 ターゲットを対象とした、簡易的に QNX Neutrino ランタイム環境を構築する事を目的とし、Windows ホストのみを利用してターゲットハードウエアで使用する HDD/SSD/CF といったマス・ストレージにランタイム環境をインストールする方法を紹介します。

2. 必要なハードウエアおよびソフトウエア

1） ターゲットハードウエア

QNX Neutrino ランタイム環境をインストールするにあたり、次のようなハードウエアを前提としています。

CPU：初代 Pentium 以降の Intel x86 CPU およびその互換 CPU
Memory：64MB 以上
HD/SSD/CF：EIDE/SATA/USB インターフェースに接続され、256MB 以上の容量
（ただし、接続された HD 等から BIOS によりダイレクトに起動が可能なこと）

2） Windows ホスト PC

QNX SDP 6.6.0 をインストールするための Windows PC を用意します。この環境を使用して起動イメージの作成やユーティリティー等のファイル類を用意します。

特に制限事項はありませんが、QNX IDE や VMware player の動作要件を満たしている事と、空いている USB ポートが必要です。

3） USB to HD/SSD/CF アダプター

USB に接続する HD/SSD/CF アダプターを利用し、ターゲットに使用するマス・ストレージに書き込みを行います。

USB ハードディスクケースや USB カードリーダー等が想定されます。

4） VMware 用ターゲットインストール CD

Windows ホストに VMware Player 等の Virtual Machine 環境を用意し、この CD を利用して QNX 6.6.0 のランタイム環境をインストールします。この CD は QNX ウエブサイトのダウンロードセンターからダウンロード可能な VM-QNXSDP660-201402230339.iso ファイルを CD-R 等に書き込んで作成します。

また、VMware Player では ISO イメージファイルから直接インストールする事も可能ですので、この機能を使用する場合は CD-R への書き込みは不要となります。

Virtual Machine 上に作成された QNX 6.6 のランタイム環境を利用して Windows PC に接続された HD/SSD/CF 等を QNX4/QNX6 フォーマットに初期化したり、ファイルを書き込む事が目的です。

5） ランタイム環境 PC（オプション）

上記の VMware 用ターゲットインストール CD は Virtual Machine だけでなく、普通の PC にもインストール可能で、Virtual Machine を使用せずに実 PC を使用してランタイム環境の作成を行う場合に用意します。

使われなくなった古い PC や x86 ボードに HD と CD を接続したハードウエアで十分です。

３. QNX Neutrino ランタイム環境の作成

QNX SDP 6.6.0 を Windows PC へインストールしたホスト環境で QNX Neutrino ランタイム環境を作成します。

また、Virtual Machine で VMware 用ランタイム環境を利用する場合は、これもインストールします。

ここで作成した QNX Neutrino ランタイム環境を HD/SSD/CF 等へ書き込みます。

３．１　ホスト環境のインストール

QNX SDP 6.6.0 - Full Installation DVD もしくは QNX SDP 6.6.0 - Windows Host ファイルを用意し、Installation Note に従ってインストールを行います。

VMware 用ランタイム環境を利用する場合は、まずは Virtual Machine の環境をインストールします。
VMware 以外の Virtual Machine でも利用可能ですが、ここでは VMware Player を例に説明します。

VMware 社のホームページからフリーの VMware Player のインストールファイルをダウンロードしてインストールします。インストールが完了したら、ゲスト OS から DVD/CD ドライブと USB ポートが利用可能なようにセットアップします。

VMware Player を起動し、新規仮想マシンの作成を選択し、DVD/CD ドライブに入れた VMware 用ターゲットインストール CD もしくは ISO イメージファイルを指定して、VMware 用ランタイム環境をインストールします。

ホスト環境およびランタイム環境の準備ができたら、本ノートに付属するいくつかのファイルを任意のフォルダにコピーしておきます。ここでは Windows 7 の環境で、ユーザーのドキュメントフォルダの下に qnx_runtime という名称でフォルダを作成し、そこに qnxbase660.build, runtime660.list, set_perm_link.sh の 3 つのファイルをコピーしたと仮定して説明します。

３．２　ビルドファイルの作成

インストールされた QNX SDP ホスト環境からターゲット用のランタイム環境を作成します。ランタイム環境は起動イメージと動作に必要なファイル群から構成されますが、ここではまず起動イメージの生成に必要なビルドファイルを作成します。

通常、カスタムの起動イメージを生成するためのビルドファイルは、オンラインヘルプの Building Embedded Systems -> Sample Buildfiles や User's Guide -> Example -> Sample buildfile を参考にして作成します。

本ノートでは一例として qnxbase660.build というビルドファイルを用意しました。このビルドファイルを機能や動作環境に合わせて、次の項目に従って変更して下さい。

１）procnto

カーネルである procnto にはいくつか種類があり、x86 用では以下の用に選択します。

	ー インスツルメンツ機能 ー	
	なし	あり
シングルコアカーネル	procnto	procnto-instr
マルチコアカーネル	procnto-smp	procnto-smp-instr

qnxbase660.build では procnto-smp-instr が指定されています。
必要に応じて上記の procnto-* に変更して下さい。

2） startup

x86 BIOS 環境で使用される startup には以下の種類があります。

	ーー 物理アドレス ーー	
	32	64
none-APIC 用	startup-bios-32	startup-bios
APIC 用	startup-apic-32	startup-apic

qnxbase660.build では startup-bios が指定されています。必要に応じて上記の startup-* に変更して下さい。

なお、startup-apic-32, startup-apic を使用する場合は pci-bios に pci-bios-v2 を指定する必要がありますので、ビルドファイルの [data=copy] セクションの下にある pci-bios 行を以下のように変更して下さい。

```
pci-bios=pci-bios-v2
```

3） Block Driver

Block Driver（マスストレージドライバ）を選択します。
qnxbase660.build では devb-eide を使用していますので、必要に応じて下記のドライバを指定して下さい。

devb-eide : IDE 接続や IDE 互換モードでの SATA 接続の HD/SSD/CF 等のドライバ。
devb-ahci : AHCI モードでの SATA 接続の HD/SSD/CFast 等のドライバ
devb-umass : USB 接続の HD/SSD/CF や USB メモリのドライバ。

また、DMA を使用しない場合は、オプションに “eide nobmstr” を追加して下さい。

4） Partition Type

qnxbase660.build では QNX6 フォーマットの Type 179 を使用しています。その他の QNX6 フォーマットである 177, 178 を使用する場合や QNX4 フォーマットの Type 77, 78, 79 を使用する場合は、下記のように変更して下さい。

例えば Type 79 を使用する場合、下記のように hd0t179 を hd0t79 に変更し、mount コマンドの -t オプションの引数も qnx6 から qnx4 に変更します。

```
waitfor /dev/hd0t79 60
mount -t qnx4 /dev/hd0t79 /
```

３．３　起動イメージの作成

3.2 で作成したビルドファイルから起動イメージを作成します。作成方法にはコマンドプロンプトからの mkifs コマンドによる方法と IDE の System Builder による方法がありますが、ここでは mkifs コマンドを使用する方法を説明します。

コマンドプロンプトを開いて以下のコマンドを実行して下さい。

```
> cd Documents¥qnx_runtime
> C:¥qnx660¥qnx660-env.bat
> sh
$ mkifs qnxbase660.build qnxbase660.ifs
```

この qnxbase660.ifs が作成された起動イメージになります。ファイル名は任意に指定できますので、複数の起動イメージを作成する場合は、ファイル名から内容が判別できる名称にすることをお奨めします。

なお、qnx660-env.bat はコマンドプロンプトで QNX SDP の機能を利用する際に必要となる環境変数を設定するバッチファイルになります。QNX SDP をデフォルトの設定でインストールした場合には C:¥qnx660 にありますが、デフォルト以外のディレクトリにインストールした場合は、そのディレクトリを指定して下さい。

３．４　ランタイム環境に必要なファイルのアーカイブ

ここではランタイム環境に必要なファイル群をひとまとめにする作業を行います。

動作に必要なファイル群はターゲットをどのように構成するかにより異なりますが、本ノートでは参考として下記のファイルリストを用意し、このリストに記載されたファイルをターゲットにコピーするようにしました。

runtime660.list :　¥qnx660¥target¥qnx6 ディレクトリの下にある etc, usr, x86 ディレクトリにある
すべてのファイル から .a (Static Library) ファイルを除いたもの

このリストに記載されているファイルを下記の手順によりアーカイブして、ターゲット環境として使用します。
なお、3.3 から引き続き実行している場合は、最初の 3 行は不要です。

```
> cd Documents¥qnx_runtime
> C:¥qnx660¥qnx660-env.bat
> sh
$ export RUNTIME=$PWD
$ cd $QNX_TARGET
$ $QNX_HOST/usr/bin/tar -T $RUNTIME/runtime660.list -cf $RUNTIME/runtime660.tar
```

これで qnx_runtime フォルダの runtime660.tar に必要なファイルをアーカイブしたファイルが作成されます。
なお、作業ディレクトリやディレクトリ名およびファイル名は任意ですので必要に応じて変更可能です。

３．５ ランタイム環境の作成

次に、3.4 で作成したアーカイブをそのままターゲットにインストールできるように、VMware 上のランタイム環境や実 PC にインストールした VMware 用ランタイムの環境にて、このアーカイブをいったん展開してディレクトリ構成やファイルパーミッションの変更、シンボリックリンクの作成を行い、再びアーカイブします。

VMware ランタイム環境に FTP や USB メモリ等を利用して qnx_runtime フォルダにあるすべてのファイルを任意のディレクトリにコピーして下さい。

まずは下記の手順でディレクトリ構成を変更します。

```
# mkdir target
# cd target
# tar -xf ../runtime660.tar
# mv etc x86
# cp -cRp usr x86
# rm -rf usr
```

次にビルドファイルと起動イメージファイルを boot ディレクトリにコピーします。

```
# cd x86
# cp -c ../../*.build boot/build/
# cp -c ../../*.ifs boot/fs/
```

最後に set_perm_link.sh を実行してファイルパーミッションとリンクの設定を行い、再びアーカイブします。

```
# sh ../../set_perm_link.sh
# tar -czf ../../archive.tgz .
# cd ../..
```

４. HD/SSD/CF 等へのランタイム環境の書き込み

「3. QNX Neutrino ランタイム環境の作成」で作成したアーカイブを必要なファイルと共に HD/SSD/CF 等に書き込みます。

４．１　HD/SSD/CF 等の初期化

VMware ランタイム環境がインストールされた Windows ホスト、もしくは VMware 用ランタイム環境を直接インストールした PC にターゲット環境で使用する HD/SSD/CF 等を取り付けます。このでは USB アダプターを利用して接続した場合で説明します。

VMware ランタイム環境はデフォルトの状態では devb-umass ドライバが起動されておらず、そのままでは USB マスストレージデバイスは認識されませんので、以下のように devb-umass ドライバを起動します。

```
# devb-umass cam pnp qnx6 sync=optional &
```

コマンドオプションは任意ですが、”qnx6 sync=optional” オプションは、QNX6 ファイルシステムをマウントする際に Read-only file system エラーとなる場合に必須となります。

HD/SSD/CF 等を接続すると、/dev/hd1 のように認識されます。

```
# ls /dev/hd*
/dev/hd0    /dev/hd0t179    /dev/hd1
```

このデバイス名は接続されてている HD や USB マスストレージ、またはパーティションの数によって異なりますので、対象となる HD/SSD/CF 等のデバイス名を間違えないように注意して下さい。

１）パーティションの作成

HD/SSD/CF 等に QNX パーティションを作成するため、以下のコマンドを実行します。
ここでは対象となるデバイスは /dev/hd1 と仮定しています。

```
# fdisk /dev/hd1 delete -a
# fdisk /dev/hd1 add -t179
# fdisk /dev/hd1 boot -t179
# fdisk /dev/hd1 loader
```

これらのコマンドでは、/dev/hd1 に存在するすべてのパーティションを消去し、最大サイズで Type 179 (QNX6 ファイルシステム) のパーティションを作成、ブートパーティションに設定して QNX ローダーを書き込んでいます。Type 79 (QNX4 ファイルシステム) でパーティションを作成する場合は、上記の t179 を t79 に変更します。

もう少し細かく設定したい場合は fdisk /dev/hd1 と実行することによりインタラクティブに操作可能ですので、他のパーティションとの共存やパーティションサイズを確認しながら設定することができます。

2） パーティションの再マウント

すでに存在していたパーティションをマウントしている可能性がありますので、デバイスを再マウントします。

```
# umount /dev/hd1
# mount -e /dev/hd1
```

この時点で /dev/hd1t179 (または /dev/hd1t79)が現れているはずですので、ls コマンドで確認します。

```
# ls /dev/hd1*
/dev/hd1    /dev/hd1t179
```

3） パーティションのフォーマット

パーティションを QNX6 ファイルシステムでフォーマットするには mkqnx6fs コマンドを使用します。実行すると本当にフォーマットするか訪ねられますので、'y' を入力します。

```
# mkqnx6fs -Truntime /dev/hd1t179
All files on /dev/hd1t179 will be lost!
Confirm filesystem re-format (y) or (n): y
Format fs-qnx6: xxxxxx blocks, xxxxx inodes, xx groups
```

mkqnx6fs が終了すると、フォーマットされたデバイスには空の .boot ディレクトリだけが作成されています。

QNX4 ファイルシステムでフォーマットする場合は、dinit コマンドでパーティションを使用します。
本当に初期化するか訪ねられますので、'y' を入力します。
また、初期化する際にブートレコードも書き換えられます。

```
# dinit -h /dev/hd1t79
All existing files on /dev/hd1t79 will be lost!  Are you sure (y or n) ? y
Using loader //x86/boot/sys/ipl-diskpc2-flop
Disk '/dev/hd1t79' contains 499905 blocks (255951K).
```

dinit が終了すると、ディスクには .altboot, .bitmap, .boot, .inode, .longfilenames ファイルが書き込まれています。この時点の起動イメージ .boot と .altboot のサイズは 0 になっています。

デフォルトの dinit の実行では、IPL のコードとして ipl-diskpc2-flop が使用されます。この IPL はフロッピーディスクや 8GB 以下の HD/SSD/CF 用の IPL です。もし、8GB 以上の HD/SSD/CF 等を使用する場合は、次のコマンドで IPL をこれに対応した ipl-diskpc2 に入れ替えて下さい。
また、ハードウエアによっては ipl-diskpc2-flop では起動できない場合もありますので、その場合も ipl-diskpc2-flop を ipl-diskpc2 に入れ替えてみてください。

```
# dinit -h -b -B ipl-diskpc2 /dev/hd1t79
```

このコマンドは以下のような dloader コマンドで実行することもできます。

```
# dloader /dev/hd1t79 pc2
```

４．２ HD/SSD/CF 等への書き込み

１） 起動イメージおよびターゲット環境の書き込み

QNX6 ファイルシステムの場合、この起動イメージをターゲットのランタイム環境で /.boot ディレクトリに置いておくことにより起動することが可能です。また /.boot ディレクトリに複数の起動イメージを置くことができ、起動時にどのイメージから起動するかを選択することも可能です。

QNX4 ファイルシステムの場合は、この起動イメージをターゲットのランタイム環境で /.boot ファイルにコピーすることにより起動することが可能です。また /.altboot ファイルにコピーした起動イメージからは、起動時に ESC キーを押すことにより起動することが可能です。

HD/SSD/CF 等に作成したパーティションを /fs の下にマウントし、作成しておいた起動イメージを HD/SSD/CF 等にコピーします。QNX6 ファイルシステムの場合は以下の通りです。

```
# mount -t qnx6 /dev/hd1t179 /fs/hd
# cp *.ifs /fs/hd/.boot
```

また、QNX4 ファイルシステムの場合は以下の通りです。もし複数の起動イメージを作成した場合は、.altboot にもコピーして下さい。

```
# mount -t qnx4 /dev/hd1t79 /fs/hd
# cp qnxbase660.ifs /fs/hd/.boot
```

次にターゲット環境をコピーします。

```
# cd target/x86
# cp -pRv . /fs/hd
```

２） 起動確認

USB で接続していた HD/SSD/CF 等を取り外し、ターゲットマシンにセットしてターゲットを起動してみます

QNX6 ファイルシステムの場合は以下のように表示されます。

```
Boot Partition 1 ? 1
QNX v1.2a Boot Loader: qnxbase660.ifs ......
Starting extra run commands
login:
```

QNX4 ファイルシステムの場合は以下のように表示されます。

```
Boot Partition 1 ? 1
Hit Esc for .altboot.....................................
Starting extra run commands
login:
```

上記のようなメッセージが表示され、パスワードなしの root アカウントでログインできれば完成です。

５. ランタイム環境インストール用 USB メモリの作成

「4. HD/SD/CF 等へのランタイム環境の書き込み」の要領で USB メモリから起動する QNX Neutrino ランタイム環境を作成し、「3. QNX Neutrino ランタイム環境の作成」で作成したアーカイブをターゲットハードウエアにインストールする手順について説明します。

５．１　QNX Neutrino 起動 USB メモリの作成

3章および4章に記載した手順で起動イメージを作成し、USB メモリにランタイム環境を書き込みます。

サンプルとして qnxbase660.usb.build ファイルを用意しました。このビルドファイルは QNX4 ファイルシステム (type 79) の USB メモリで起動するように記述されていますので、事前に Windows ホスト環境にて起動イメージ qnxbase660.usb.ifs を作成しておいて下さい。

このファイルを利用する場合、書き込みの手順は以下のようになります。

まず VMware ランタイム環境にて USB メモリを装着して devb-umass を起動します。

```
# devb-umass cam pnp &
```

USB メモリが /dev/hd1 のように認識されたら fdisk を実行し、

```
# fdisk /dev/hd1 delete -a
# fdisk /dev/hd1 add -t79
# fdisk /dev/hd1 boot -t79
# fdisk /dev/hd1 loader
```

再マウントして dinit, dloader を実行します。

```
# umount /dev/hd1
# mount -e /dev/hd1

# dinit -h /dev/hd1t79
# dloader /dev/hd1t79 target/x86/boot/sys/ipl-diskpc2
```

そしてファイルシステムをマウントして起動イメージとランタイム環境を書き込みます。

```
# mount -t qnx4 /dev/hd1t79 /fs/usbm
# cp qnxbase660.usb.ifs /fs/usbm/.boot

# cd target/x86
# cp -pRv . /fs/usbm

# mkdir /fs/usbm/runtime
# cp ../../archive.tgz /fs/usbm/runtime
```

５．２　USB メモリからの起動とランタイム環境のインストール

QNX Neutrino ランタイム環境が書き込まれた USB メモリをターゲットハードウエアに装着し、電源を入れて BIOS 設定画面にて USB メモリの起動優先順位を上げるか、BIOS Boot メニューから USB メモリを選択して起動します。

無事に起動すれば login: プロンプトが表示されますので、パスワードなしの root アカウントでログインし、下記のような手順で、ターゲットハードウエアの HD/SSD/CF 等に QNX Neutrino ランタイム環境をインストールします。

まずブロックドライバを起動します。ここでは例として HD を対象に devb-eide を起動しています。
また、ファイルシステムは QNX6 (type 179) を使用しています。

```
# devb-eide blk auto=partition dos exe=all qnx6 sync=optional cam quiet &
```

HD が /dev/hd1 のように認識されたら fdisk を実行し、

```
# fdisk /dev/hd1 delete -a
# fdisk /dev/hd1 add -t179
# fdisk /dev/hd1 boot -t179
# fdisk /dev/hd1 loader
```

再マウントして mkqnx6fs を実行します。

```
# umount /dev/hd1
# mount -e /dev/hd1

# mkqnx6fs -Truntime /dev/hd1t179
```

そしてファイルシステムをマウントして起動イメージとランタイム環境を書き込みます。

```
# mount -t qnx6 /dev/hd1t179 /fs/hd
# cp /boot/fs/qnxbase660.ifs /fs/hd/.boot/
# cd /fs/hd
# tar -xzvf /runtime/archive.tgz
```

ターゲットの HD への書き込みが終了したら shutdown を実行し、USB メモリを取り外して再起動して、ターゲットハードウエアの HD から QNX Neutrino ランタイム環境が起動することを確認して下さい。

６. ランタイム環境インストール用 CD の作成

「3. QNX Neutrino ランタイム環境の作成」で作成したアーカイブを CD から起動してターゲットハードウエアにインストールする手順について説明します。

注： QNX SDP 6.6.0 では QNX SDP 6.5.0 まで提供されていた Runtime Kit が含まれておらず、また、ISO イメージを作成するユーティリティーである mkisofs が QNX 6.6.0 の環境では動作しません。このため、ランタイム環境インストール CD を作成するためには QNX SDP 6.5.0 Neurtino Installation and Boot CD と、この CD から起動して QNX Neutrino 6.5.0 が動作する環境が必要となります。

６．１ 必要な機材の用意

ランタイム環境インストール CD を作成するためには以下の機材やファイル等が必要となります。

1) QNX SDP 6.5.0 － QNX Neutrino RTOS Installation and Boot CD [X86-only]

ダウンロードセンターからこの ISO イメージをダウンロードして CD-R に書き込んだものを使用します。（ダウンロードには myQNX アカウントでログインしておく必要があります。）
また、この CD から QNX Neutrino 6.5.0 を起動して使用するだけなら、QNX SDP 6.5.0 の開発ライセンスは必要ありません。

2) PC

上記の CD から起動して QNX Neutrino 6.5.0 が動作する PC を用意します。
Windows ホストとして使用している PC でも、上記の CD から QNX 6.5.0 が起動できれば使用可能です。

3) mkisofs ユーティリティ

ダウンロードセンターから cdrecord for QNX Neutrino 6.3.0 - binaries をダウンロードすると、cdrtools-201beta-bin-qnx.tgz というファイルが保存されます。mkisofs はこのファイルに含まれています。

4) インストレーション・スクリプトファイル

HD 等をフォーマットしてファイルを書き込むスクリプトが必要になります。これには本ノートに添付してある rtinstall ファイルを使用します。このスクリプトは QNX SDP 6.5.0 オンラインヘルプの OS Technotes -> How to create a Runtime Kit from the QNX Software Development Platform -> Sample installation script にあるスクリプトに多少変更を加えたものです。

ただし、このスクリプトはインストール先が /dev/hd0 固定で、このデバイスにあるパーティションを強制的に消去しますので、不都合がある場合には適宜編集して下さい。もちろん上記のオンラインヘルプにあるサンプルスクリプトを参考に作成されても構いません。

5) USB メモリ

Windows ホスト環境で作成した起動イメージ（qnxbase660.ifs 等）、ファイルアーカイブ（archive.tgz）、上記の cdrtools-201beta-bin-qnx.tgz および rtinstall ファイルを書き込んでおきます。

６．２　QNX Neutrino 6.5.0 の起動と CD の作成

QNX Neutrino RTOS Installation and Boot CD を PC の CD ドライブに入れて電源を入れ、CD から起動するように BIOS または BIOS Boot Menu を設定します。

起動途中で CD から QNX Neutrino を実行するか、HD に QNX をインストールするか尋ねられますので、

```
F2 - Run from CD (Hard Disk filesystems mounted under /fs)
F3 - Install QNX to a new disk partition

Select?
```

F2 を選択して実行を継続して下さい。

Photon が起動し、画面の解像度等を選択すると、ログイン画面が現れますのでパスワードなしの root でログインして下さい。

Photon のデスクトップが標示されたら、Launch → Utilities → Terminal でターミナルを起動して、以下のコマンドを実行して下さい。

```
# devb-ram ram capacity=1048576            (RAM ディスクを 512MB 確保)
# mount -t qnx4 /dev/hd1t77 /fs/ramdisk    (RAM ディスクは /dev/hd1t77 と仮定)
# cd /fs/ramdisk
# mkdir runtime
# cd runtime
```

CD 化するファイルが書き込まれた USB メモリを取り付けます。通常、USB メモリは自動的に認識され、/fs/usb0 にマウントされます。

```
# cp /fs/usb0/files/* .            (ファイル類は files ディレクトリにあると仮定)
# chmod 755 rtinstall              (rtinstall に実行パーミッションを付加)
# textto -l rtinstall              (改行コードを LF に書き直し、誤動作を防止)

# tar -xzf cdrtools-201beta-qnx-bin.tgz opt/bin/mkisofs       (mkisofs の取り出し)
```

CD イメージのディレクトリを作成し、必要なファイルをコピーします。

```
# mkdir cdimage
# cd cdimage
# mkdir -p repository
# mkdir -p boot/fs
# cp ../qnxbase660.ifs ./boot/fs
# cp /fs/cd0/boot/fs/qnxbase.qfs ./boot/fs
# cp /fs/cd0/rtkit/instflop-rtkit.dat ./instflop.dat
# mv ../archive.tgz ./repository
# cp ../rtinstall .

# ../opt/bin/mkisofs -r -b instflop.dat -c boot/isocatalog -J -o ../rtkit.iso .
```

rtkit.iso ファイルがランタイム環境インストール CD の ISO イメージファイルになります。
この ISO ファイルを Windows ホスト等の PC にコピーし、CD-R 等に書き込んで下さい。

7. ドライバの起動と Screen Graphics の準備

7. 1　各種ドライバの起動について

QNX 6.6.0 では enum-devices (Device-enumerator) ユーティリティーが廃止されたために、自動的にデバイスをサーチしてそのドライバを起動する機能が提供されていません。

このため、ハードウエアに搭載されている各デバイスのドライバはビルドファイルに記述して起動イメージの中で起動するか、起動イメージから実行されている sysinit スクリプトや rc.local 等のスクリプトの中でドライバの起動コマンドを記述する必要があります。

あるいは VMware ターゲット環境の startup_aps.sh や startup.sh のように独自の起動スクリプトを作成しても構いません。

本ノートでは rc.drivers という名称で特定のドライバを起動するスクリプトを作成し、sysinit の中から呼び出すようにしてみました。

まず、/etc/system/sysinit ファイルを次の要領で変更して下さい。

下記のように、QNET を起動する際に io-pkt が既に動作している必要がある部分がありますので、

```
# Enable qnet if user has enabled it.
if test -r /etc/system/config/useqnet -a -d /dev/socket; then
              mount -Tio-pkt lsm-qnet.so
fi
```

この記述の前あたりに、以下ように rc.drivers を呼び出す部分を追加して下さい。

```
# Run the driver startup script
if test -x /etc/rc.d/rc.drivers; then
       . /etc/rc.d/rc.drivers
fi
```

次に以下のような rc.drivers ファイルを作成して実行パーミッションを付加し、/etc/rc.d/ ディレクトリにコピーして下さい。

```
# Audio driver
io-audio -d intel_hda &

# Network driver
io-pkt-v4-hc -d e1000 -p tcpip &
if_up -p wm0
dhcp.client &

# Parallel driver
devc-par -p 0x378 &

# Serial driver
devc-ser8250 -u1 3f8,4 -u2 2f8,3 &
```

この rc.drivers ファイルでは Intel HD Audio デバイス、Intel GbE デバイス、標準パラレルポートおよびシリアルポートを持ったハードウエアを対象に、それぞれのドライバの起動やネットワークの設定を行っています。この例を参考に使用するハードウエアやネットワーク環境に合わせてドライバやネットワークの設定を行ってみて下さい。

７．２　Screen Graphics の準備について

QNX 6.6.0 では Photon が廃止され、Screen Graphics が導入されました。
しかし、Screen Graphics では Photon のように完成された GUI 環境は提供されておらず、コンソールが別に必要になるなど、少し準備がありますので、本ノートでは screen を起動してサンプルプログラムが動作するまでの手順を紹介します。

１）コンソールの用意

前述の通り、screen を起動すると VGA 等に出力されているコンソール表示は見えなくなり、グラフィックスのみの表示となりますので、別にコンソールを用意すると便利です。ここではシリアルポートに接続した PC をコンソールにします。

まず、シリアルポートから login できるように、デフォルトでは以下のようになっている /etc/config/ttys ファイルで、

```
con1 “/bin/login” qansi-m on
con2 “/bin/login” qansi-m on
con3 “/bin/login” qansi-m on
con4 “/bin/login” qansi-m on
```

下記のように “ser1 ...” を追加します。

```
con1 “/bin/login” qansi-m on
con2 “/bin/login” qansi-m on
con3 “/bin/login” qansi-m on
con4 “/bin/login” qansi-m on
ser1 “/bin/login” qansi-m on
```

これで /dev/ser1 ポートで login が動作するようになり、コンソールとして使用可能となります。

デフォルトでは 57600 baud, 8 bit, none parity, 1 stop bit に設定されますので、PC 側のターミナルソフトを同じ設定にして下さい。

２）スクリーンの起動

screen を起動するには 2 つの環境変数の設定が必要ですので、次のようなシェルスクリプトファイルを作成して、シリアルポート接続のコンソール等から実行します。ここでは多くのハードウエアで動作確認できるように、VESA BIOS 表示を指定しています。

```
export GRAPHICS_ROOT=/usr/lib/graphics/vesabios
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib:/lib:/lib/dll
```

例えば setscreen.sh という名前で作成した場合、実行パーミッションを付加して下記のように実行します。

```
# . ./setscreen.sh
# screen &
```

screen が実行されると VGA 等のディスプレイには、初期化されていないビデオメモリのイメージが表示されるはずです。

3） サンプルプログラムの実行

上記のように GPU を使用しない VESA BIOS 表示では display_image, gles1-gears, sw-vsync 等が実行可能ですので、試してみて下さい。

なお、display_image は画像ファイルを表示するプログラムですので、640 x 480 程度の JPEG ファイル等を用意して、下記のように実行して下さい。

```
# display_image -file=./sample.jpg
```

デフォルトでは 640 x 480 の解像度で表示されるように設定されています。
この設定は /usr/lib/graphics/vesabios/graphics.conf ファイルに記述されていますので、必要に応じて変更して下さい。

8. 補足事項

1） 注意点

本のノートでは簡易的な QNX Neutrino ランタイム環境を作成するための参考として作成したものであり、すべてのハードウエアで起動を保証するものではありません。あくまで参考情報としてご利用ください。

ユーザ領域を増やすためには不要なファイルを削除する必要があります。

HD/SSD/CF 等への書き込み時にはライトキャッシュが効いているため、書き込みが終了するまで数秒待つことが必要です。ライトキャッシュから HD/SSD/CF 等へのデータ書き込み前に電源を切ったりリセットすると、書き込みが不完全なものになり、起動に失敗する場合があります。

CF が接続される EIDE の Access Mode (BIOS で設定) によっては起動できない事がありますので、Auto か CHS (Normal) に設定しておく事をお勧めします。

CF を EIDE に接続する際は、プライマリー・マスターデバイスとして接続し、スレーブデバイスは接続しない事をおすすめします。これは CF によってはスレーブデバイスが存在すると問題を起こすケースがあるためです。

QNX6 ファイルシステムは QNX 6.4.0 から採用されたファイルシステムで、突然の電源断時におけるファイル保護が強化されておりますが、どちらかというと HDD 向けのファイルシステムであり、マスストレージへのアクセス回数も増加しますので、SSD や CF などの書き込み回数が制限されているデバイスを使用する場合は、QNX4 ファイルシステムを使用することをお薦めします。設定) によっては起動できない事がありますので、Auto か CHS (Normal) に設定しておく事をお勧めします。

VMware Player の仮想マシン設定で USB コントローラの “USB の互換性” を USB3.0 に設定すると、QNX Neutrino 6.6.0 の環境では、接続されている USB デバイスが認識されませんので、“USB の互換性” を USB2.0 に設定してご利用下さい。また、この環境で HD/CF/SSD 等を USB3.0 インターフェースに接続すると、QNX6 ファイルシステムでは、書き込み速度が遅くなる、書き込み途中で “Read-only file system” エラーが発生して書き込みできなくなる、等の問題が報告されていますので、HD/CF/SSD 等は USB2.0 インターフェースに接続してご利用下さい。

2） サンプルビルドファイルについて

本ノートに添付してあるビルドファイル qnxbase660.build ファイルは QNX 6.5.0 の qnxbasesmp.build をベースに、diskboot の代わりに IDE に接続された HD や CF から起動する最小限の記述を行ったものです。

CD-ROM ドライブをマウントする場合には、/dev/cd0 のマウント行のコメントを外して下さい。
また、他のパーティションや他のディスク等を自動的にマウントしたい場合は、devb-eide コマンドのオプションの変更や mount コマンドの追加が必要になります。
これは、起動後に手動でマウントするか、起動スクリプトでマウントしても構いません。

3） USB 接続デバイスからの起動について

USB 接続のデバイスからの起動については、Chipset および BIOS が USB からの起動をサポートしていることが前提条件あり、また、すべてのデバイスからの起動を保証するものではありません。

その他にも Chipset や BIOS, USB デバイスの種類等で動作が異なることが予想されますので、実用化の際には十分なテストが必要です。

4） CF から起動できない場合

CF に QNX 6.6.0 ランタイム環境をインストールしてターゲットハードウエアで起動した際に、”Missing Operating System” エラーが表示され、起動できない場合があります。

原因はいくつか考えられますが、ランタイム環境が正しくインストールされているにもかかわらず、このエラーが表示されるケースでは、CF のジオメトリが QNX のブートローダーと適合していない可能性があります。

このような場合、以下の方法で起動できるようになる可能性がありますので試してみて下さい。

(1) CF を完全に初期化

ARM 系の BSP で SD カードに IPL を書き込む際によく使用される方法ですが、

```
# dd if=/dev/zero of=/dev/hd1 bs=1024k count=10
```

のような dd によるゼロフィルを CF の先頭に対して行い、ジオメトリやパーティション情報を完全に消去します。この例では /dev/hd1 にマウントされている CF に対し、1MB x 10 で 10MB 分をゼロフィルしていますが、実際にはこれより少なくても良いと思われます。

(2) ターゲットハードウエアでの fdisk

ターゲットハードウエアで、可能ならば QNX SDP 6.5.0 Neutrino boot CD から起動、もしくはハードディスクや USB メモリにインストールした QNX 6.5.0 または QNX 6.6.0 を起動して、CF の fdisk を実行します。

ジオメトリやパーティション情報を持たない CF を fdisk すると、ターゲットハードウエアの環境で、QNX が認識可能なジオメトリにて初期化が行なわれます。

なお、パーティション情報を持たないデバイスをフルスクリーンのインタラクティブモードで fdisk する場合は、以下のように、-z オプションを指定する必要があります。

```
# fdisk -z /dev/hd1
```

コマンドラインモードで fdisk を実行する場合には -z オプションは不要です。

(3) mkqnx6fs およびファイルコピーの実行

mkqnx6fs はターゲットハードウエアでもホスト PC でも、どちらで行ってもかまいません。
その後、ホスト PC にてファイルコピーを実行して下さい。

９. 改訂履歴

1） 0.99　　ベータ版
2） 1.00　　「5. ランタイム環境インストール用 USB メモリの作成」を追加
　　「6. ランタイム環境インストール用 CD の作成」を追加
　　「7. ドライバの起動と Screen Graphics の準備」を追加
3） 1.01　　「8. 補足事項」の 1) 注意点　に VMware 環境での注意事項を追加
4） 1.02　　「8. 補足事項」に「4) CF から起動できない場合」を追加