

QNX 6.5.0 SP1 ランタイム環境構築方法

QNX ソフトウェアシステムズ

エンジニアリングサービス

2019 年 3 月 19 日

1. はじめに

本文書では Windows ホスト環境のみを用いて x86 CPU ターゲットハードウェアに QNX 6.5.0 SP1 のランタイム環境をインストールする方法を説明します。

概要としては、Windows ホスト環境の作成、ターゲット用のファイルのアーカイブ、インストール用 USB メモリの作成となります。

2. 必要なハードウェアおよびソフトウェア

1) Windows PC

QNX SDP (Software Development Platform) 6.5.0 SP1 で正式にサポートされているホスト OS は以下の通りですが、最新の Windows 10 でも基本的には動作します。

- Windows 7 Professional 32- and 64-bit
- Windows Vista Business 32- and 64-bit
- Windows XP Professional SP3
- Windows 2000 SP4

2) USB メモリ

ランタイム環境をターゲットハードウェアにインストールするためのインストール USB メモリを作成します。容量は 1GB 以上であれば問題ありません。

3) ディスクイメージ書き込みソフトウェア

インストールディスクイメージを USB メモリに書き込むユーティリティソフトウェアです。

一般的には下記のソフトウェアが知られていますが、同等の機能を持ったソフトウェアであればそれでも構いません。

- Win32 Disk Imager
- DD for Windows

4) サンプルファイル

この構築方法では本書と共にアーカイブされている以下のサンプルファイルを使用します。

- disk.cfg
- install-usb.build
- mkusbimage.sh
- root.build
- set-perm-link.sh
- targetfile.list

3. Windows ホスト環境の作成

Windows ホスト環境の作成には下記の QNX Download Center サイトからダウンロードできるファイルを使用します。

<http://www.qnx.com/download/group.html?programid=20905>

すでに Windows PC に QNX SDP 6.5.0 SP1 のホスト環境をインストール済みの場合は、「3.2 パッチのインストール」のみ確認して下さい。

3.1 QNX SDP 6.5.0 SP1 のインストール

下記の QNX SDP 6.5.0 インストーラーファイルをダウンロードし、Installation Note に従ってインストールします。 Full Installation DVD からインストールしても構いません。

- QNX Software Development Platform 6.5.0 [Build 201007091524] - Windows Hosts

次に下記の QNX SDP 6.5.0 SP1 アップデートファイルをダウンロードし、Installation Note に従ってインストールします。 こちらも Full Installation DVD からインストールしても構いません。

- QNX Software Development Platform 6.5.0 Service Pack 1 - Windows Hosts

3.2 パッチのインストール

Windows ホストのみでインストールディスクイメージを作成するために必要な以下のパッチをインストールしてください。

- QNX Software Development Platform 6.5.0 SP1 applypatch Patch [Patch ID 3792]
- QNX Software Development Platform 6.5.0 SP1: diskimage Patch [patch ID 4245]
- QNX Software Development Platform 6.5.0 SP1 mkxfs Patch [Patch ID 4392]

その他にも QNX SDP 6.5.0 SP1 用の各種パッチが同じページで公開されていますので、必要に応じてインストールしてください。

4. x86 BIOS/APIC BSP のインストール

4.1 x86 BIOS/APIC BSP のダウンロード

下記の Foundry27 サイトから x86 BIOS/APIC BSP (Board Support Package) をダウンロードしてください。

<https://community.qnx.com/sf/wiki/do/viewPage/projects.bsp/wiki/X86Bios>

このページには QNX 6.6.0 用と QNX 6.5.0 SP1 用がありますので、間違えないように注意してください。x86 BIOS/APIC BSP には QNX SDP 6.5.0 SP1 リリース以後に発売された CPU チップセットや SoC のサポート、USB3.0 コントローラのサポート等が含まれています。

2018 年 7 月現在では下記の BSP が最新となっていますが、今後も更新される可能性がありますので、できるだけ最新の BSP をインストールする事をお勧めします。

- BSP_x86-generic_br-650_be-650sp1_SVN821080_JBN30.zip (BSP 本体)
- BSP_x86-generic_br-650_be-650sp1_PATCHES_JBN30.zip (BSP のパッチ)

4.2 x86 BIOS/APIC BSP のビルド

x86 BIOS/APIC BSP のビルド方法には QNX Momentics IDE を使用した方法とコマンドプロンプトでコマンドラインによる方法があり、ここではコマンドプロンプトを使用した方法を説明します。

BSP 本体と BSP のパッチを適当なフォルダ(ここでは C:\Users\qnxusr\x86bsp とします)にコピーします。コマンドプロンプトを開いて、次のコマンドを実行します。

```
c:\Users\qnxusr> cd x86bsp  
c:\Users\qnxusr\x86bsp> sh  
sh-3.1$ unzip BSP_x86-generic_br-650_be-650sp1_SVN821080_JBN30.zip  
sh-3.1$ unzip BSP_x86-generic_br-650_be-650sp1_PATCHES_JBN30.zip  
sh-3.1$ cp -R patches/* ./prebuilt  
sh-3.1$ make
```

これで c:\Users\qnxusr\x86bsp\install フォルダに x86 BIOS/APIC BSP にあらかじめ含まれていたバイナリーやソースコードからビルドされたバイナリーが集約されます。

4.3 x86 BIOS/APIC BSP の適用

x86 BIOS/APIC BSP をビルドしただけでは、生成されたバイナリーは開発環境のターゲット用のファイルにはコピーされておらず、これらのバイナリーをホスト環境やターゲット環境に適用するには目的によっていくつか方法があります。

1) 開発環境のターゲット用のファイルを変更したくない場合

従来の開発環境を変更したくない場合や x86 BIOS/APIC BSP を一時的に試したい場合には、以下のよう
に x86 BIOS/APIC BSP を利用します。

- 起動イメージのビルドには c:\Users\qnxusr\x86bsp\images\x86-generic.build ファイルを変更して make するか、カスタムのビルドファイルを c:\Users\qnxusr\x86bsp\images フォルダにコピーして下記のように mkifs を実行します。この例では qnxbasesmp.build ファイルから qnxbasesmp.ifs 起動イメージを作成しています。

```
mkifs -r../install qnxbasesmp.build qnxbasesmp.ifs
```

- ターゲットハードウェアにはビルドした起動イメージと c:\Users\qnxusr\x86bsp\install フォルダにあるバイナリーを手動でコピーします。

2) 開発環境のターゲット用のファイルを変更する場合

c:\¥Users¥qnxusr¥x86bsp¥install フォルダに集約されているバイナリーを次のコマンドで開発環境のターゲット用のファイルに上書きコピーします。

```
c:\¥Users¥qnxusr> cd x86bsp
```

```
c:\¥Users¥qnxusr¥x86bsp> sh
```

```
sh-3.1$ cp -pR install/usr $QNX_TARGET
```

```
sh-3.1$ cp -pR install/x86 $QNX_TARGET
```

これにより開発環境は x86 BIOS/APIC BSP が完全に適用された状態になります。

5. 起動イメージの作成

x86 BIOS/APIC BSP が適用された環境で起動イメージを作成します。

ここではビルドファイルの例として QNX SDP セルフホスト用として用意されている下記のビルドファイルを使用します。

- qnxbasesmp.build

- qnxbasesmp-apic.build

これらのビルドファイルを適当なフォルダ(ここでは C:\¥Users¥qnxusr¥runtime とします)にコピーして起動イメージを作成し、ターゲットハードウェア用の起動イメージとします。

```
c:\¥Users¥qnxusr> cd runtime
```

```
c:\¥Users¥qnxusr¥runtime> sh
```

```
sh-3.1$ cp $QNX_TARGET/x86/boot/build/qnxbasesmp*.build .
```

```
sh-3.1$ mkifs qnxbasesmp.build qnxbasesmp.ifs
```

```
sh-3.1$ mkifs qnxbasesmp-apic.build qnxbasesmp-apic.ifs
```

注:もし x86 BIOS/APIC BSP を開発環境のターゲット用のファイルにコピーしていない場合は、「4.3 x86 BIOS/APIC BSP の適用」の「1) 開発環境のターゲット用のファイルを変更したくない場合」に従って起動イメージを作成してください。

なお、QNX SDP セルフホスト用として用意されている上記のビルドファイルは xHCI (USB3.0) をサポートしていませんので、もしターゲットハードウェアに xHCI のコントローラしか搭載されていない場合は USB 機器が使用できません。

このようなケースでは、本書とともにアーカイブされている下記のビルドファイルを使用してください。

- qnxbasesmp.xhci.build
- qnxbasesmp.xhci-apic.build

これらのビルドファイルを適当なフォルダ(ここでは C:¥Users¥qnxusr¥runtime とします)にコピーして、上記と同様な手順で起動イメージを作成します。

```
c:¥Users¥qnxusr> cd runtime
c:¥Users¥qnxusr¥runtime> sh
sh-3.1$ mkifs qnxbasesmp.xhci.build qnxbasesmp.ifs
sh-3.1$ mkifs qnxbasesmp.xhci-apic.build qnxbasesmp-apic.ifs
```

6. ランタイム環境に必要なファイルのアーカイブ

開発環境に含まれる、ターゲット用のファイルからターゲットハードウェアにインストールするファイルを収集してアーカイブします。

アーカイブするファイルは目的によって異なりますが、ここではサンプルとしてほぼすべてのファイルをアーカイブするための targetfile.list を用意しましたので、このファイルを C:¥Users¥qnxusr¥runtime にコピーして次の手順でアーカイブを作成します。

```
c:¥Users¥qnxusr> cd runtime
c:¥Users¥qnxusr¥runtime> sh
sh-3.1$ export RUNTIME=$PWD
sh-3.1$ cd $QNX_TARGET
sh-3.1$ tar -T $RUNTIME/targetfile.list -czf $RUNTIME/targetfile.tgz
sh-3.1$ cd $RUNTIME
```

7. インストール用 USB メモリの作成

7.1 起動イメージの作成

インストール用 USB メモリ用の起動イメージを作成するためのビルドファイル `install-usb.build` をサンプルとして用意しましたので、このファイルと `set-perm-link.sh` ファイルを `C:\Users\qnxusr\runtime` にコピーして次の手順で起動イメージを作成します。

```
c:\Users\qnxusr> cd runtime  
c:\Users\qnxusr\runtime> sh  
sh-3.1$ mkifs install-usb.build install-usb.ifs
```

7.2 ディスクイメージの作成

`mkxfs` および `diskimage` ユーティリティを使用して、これまでに作成した起動イメージやアーカイブをディスクイメージに書き込みます。

サンプルとして用意した `disk.cfg`, `root.build`, `mkusbimage.sh` ファイルを `C:\Users\qnxusr\runtime` にコピーして次の手順でディスクイメージを作成します。

```
c:\Users\qnxusr> cd runtime  
c:\Users\qnxusr\runtime> sh  
sh-3.1$ sh mkusbimage.sh
```

これによりディスクイメージファイル `usb.img` が作成されます。

7.3 USB メモリへの書き込み

Windows PC に USB メモリを装着し、Win32 Disk Imager 等のディスクイメージ書き込みソフトウェアを起動します。

イメージファイルに `usb.img` を指定し、書き込み先のディスクに装着した USB メモリを指定し、書き込みを実行します。

8. ターゲットハードウェアの準備

8.1 ターゲットハードウェアの BIOS 設定

1) ブートデバイスの選択

BIOS のブートデバイスの優先順位の設定で、USB デバイスを HDD よりも高く設定しておきます。

起動時に USB デバイスからブートするように選択できれば、それでも構いません。

2) SATA モードの選択

デフォルトでは AHCI になっている事が多く、そのまま構いませんが、AHCI で何か問題があれば、IDE 互換モードに設定してください。

3) USB Legacy Support 設定

BIOS によって名称が異なったり、そのメニューがない場合もありますが、この設定は USB Keyboard や Mouse を PS/2 マウスとしてエミュレートするかどうかを Enable / Disable します。

この機能を Enable にしていると起動出来ない、特定のドライバーが動作しない、等の問題が発生する事があるため、それが疑われる場合はこの機能を Disable にしてみてください。

注:この機能を Disable にすると、QNX 6.5.0 の USB ドライバーが動作するまで、USB Keyboard は使用できません。

8.2 ターゲットハードウェアの起動

ターゲットハードウェアにインストール用 USB メモリを装着して電源を入れ、USB メモリから QNX 6.5.0 がブートしてシェルが起動する事を確認します。

次にランタイム環境の入った USB メモリを以下のコマンドで /fs/usb0 にマウントして USB メモリの内容が表示される事を確認します。

```
# sh /scripts/mount-usb.sh
```

```
# ls -lR /fs/usb0
```

8.3 ブロックドライバーの起動

SATA モードを AHCI に設定している場合は、次のコマンドを実行します。

```
# sh /scripts/start-ahci.sh
```

SATA モードを IDE 互換モードに設定している場合は、次のコマンドを実行します。

```
# sh /scripts/start-eide.sh
```

上記のスクリプトは接続されている HD が1台の場合を想定しており、複数の HD が接続されている場合は、スクリプトを変更するか手動でマウントして下さい。

ブロッコドライバー起動後に /dev の下に hd0 デバイスが作成されている事を確認します。

```
# ls /dev/hd0
```

8.4 パーティションの作成

QNX 6.5.0 をインストールするためのパーティションを作成します。

以下のコマンドで /dev/hd0 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 filesystem) のパーティションを作成、ブータブルに設定して QNX ロードーを書き込んでいます。

```
# sh /scripts/format-hd0.sh
```

次に QNX6 ファイルシステムを /fs/hd0 にマウントし、/fs/hd0 に .boot ディレクトリが作成されている事を確認します。

```
# sh /scripts/mount-hd0.sh
```

```
# ls -al /fs/hd0
```

これでターゲットハードウェアに QNX 6.5.0 SP1 をインストールするための準備ができました。

なお、これらのスクリプトでは QNX6 ファイルシステムを作成と /fs/hd0 へのマウントを行っていますが、パーティションの作成方法やフォーマットを変更したい場合はこのスクリプトを変更するか手動で行ってください。

9. ランタイム環境の作成

ターゲットハードウェアの準備が完了したら、次のスクリプトで USB メモリに格納されているランタイム環境に必要なファイルのアーカイブを一旦ターゲットハードウェアのハードディスクに展開し、ディレクトリ構成を調整してファイルパーミッションの変更やシンボリックリンクの設定を行います。

```
# sh /scripts/mk-runtime-img.sh
```

これにより直接ターゲットハードウェアのハードディスクにインストール可能なランタイム環境 runtime.tgz が作成され USB メモリにアーカイブされます。

このスクリプトは毎回実行する必要はなく、インストールするファイルが変更された場合等に行ってください。

また、引き続き「10. ランタイム環境のインストール」を行う場合は、次のスクリプトでハードディスクを再フォーマットしておきます。

```
# sh /scripts/reformat-hd0.sh
```

10. ランタイム環境のインストールと動作確認

ランタイム環境のインストールを実行するスクリプトを実行します。

```
# sh /scripts/inst-runtime.sh
```

これによりターゲットハードウェアのハードディスクに起動可能な QNX 6.5.0 SP1 のランタイム環境のインストールされましたので、シャットダウンして電源を切り、USB メモリを取り外します。

```
# shutdown -b
```

ターゲットハードウェアの電源を入れ、QNX 6.5.0 SP1 が起動し、login: プロンプトが出る事を確認します。

このランタイム環境では Photon を直接起動しないように /etc/system/config/nophoton ファイルを作成していますので、このファイルを削除すると直接 Photon が起動するようになります。

Photon を最初に起動した時のディスプレイの解像度は SVGA モードで 640x480 に設定されています。

これは Photon 起動後に Configure -> Display メニューから変更する事ができますが、ターゲットハードウェアの Video BIOS によってはディスプレイモードや解像度の切り替えがうまく働かない場合があります。

この場合は、/etc/system/config/display.conf ファイルを直接編集して、ディスプレイモードや解像度を設定して下さい。

11. 変更履歴

2018 年 7 月 4 日 初版

2019 年 3 月 19 日 「5. 起動イメージの作成」に xHCI 対応の起動イメージ作成方法を追記