WNDRVR

VxWorks 7 SDK for Raspberry Pi 4

Ver.1.2

2021 年 5 月

目次

1．はじめに

VxWorks 7 SDK は、VxWorks アプリケーション開発者向けの開発環境であり、次の機能が含まれています:

- ダウンロード可能なカーネルモジュール（DKM）と RTP（リアルタイムプロセス）アプリケーションの両方をビルドできる clang / LLVM を使った標準のクロスコンパイルツール
- 簡易ビルドマネージャ：makefile、cmake、
- ターゲット/アーキテクチャ固有：ターゲットプラットフォームで起動可能な汎用 VxWorks カーネルが含まれています
- アプリケーション開発用のヘッダーファイルとライブラリ
- ウインドリバーデバッガ（wrdbg）
- ドキュメンテーション（英語版）

このガイドは、VxWorks プラットフォーム用のアプリケーションの開発を開始して実行するのに役立ちます。 これを使用して、新しいアプリケーションを作成したり、VxWorks の機能を体験することができます。

2．開発環境のセットアップ

https://labs.windriver.com から RaspberryPi4B プラットフォーム用の VxWorks SDK をダウンロードし、解凍してください。 アプリケーションの作成とデバッグの詳細については、SDK のドキュメントを参照してください。

ホスト動作環境
この SDK は、Linux ホストで実行することを目的としています。 このドキュメントの例は、Debian 系に固有のものです。（ubuntu18.04 で説明します）

ホスト依存環境の設定
Debian 系では、次のパッケージを事前にインストールする必要があります:

```
$ sudo apt install build-essential libc6:i386
```

FTP サーバを開発ホストにインストールすると、アプリケーションを容易に展開でき、VxWorks ターゲットからホストファイルシステムにアクセスできるようになります。

SDK に含まれている VxWorks カーネルイメージのランタイム構成に対応するために、pyftpdlib の FTP サーバオプションを使用しています。

では、下記の手順で pyftpdlib をインストールしてください。

```
$ sudo apt install python-pip
$ sudo pip install pyftpdlib
```

3．Raspberry Pi 4 で VxWorks をブート

1) SD カードを作成するには、下記のファームウェアをダウンロードします:

   https://github.com/raspberrypi/firmware/archive/1.20200212.tar.gz

   ```
   $ cd ~
   $ mkdir WindRiver
   $ wget https://github.com/raspberrypi/firmware/archive/1.20200212.tar.gz
   $ gunzip 1.20200212.tar.gz
   $ tar xvf 1.20200212.tar
   ```

   SD カードを FAT32 ファイルシステムとしてフォーマットし、「boot」ディレクトリの内容をダウンロードしたファームウェアから SD カードにコピーします。

   ```
   $ cd firmware-1.20200212/boot
   $ cp *.*  /media/xxxxx/xxxx-xxxx/
   ```

2) Raspberry Pi 4 の u-boot をコンパイルし、バイナリーを SD カードにコピーします

   Ubuntu ホストでの例

   u-boot のビルド

   ```
   $ sudo apt install gcc-aarch64-linux-gnu
   $ cd ~/WindRiver
   $ git clone https://gitlab.denx.de/u-boot/u-boot.git
   $ cd u-boot
   $ CROSS_COMPILE=aarch64-linux-gnu- make rpi_4_defconfig
   $ CROSS_COMPILE=aarch64-linux-gnu- make
   ```

   u-boot.bin を u-boot-64.bin にコピーして SD カードにコピー

   ```
   $ cp u-boot.bin u-boot-64.bin
   $ cp u-boot-64.bin /media/xxxx/xxxx-xxxx/
   ```

3) SDK のインストール先/vxsdk/sdcard/内のファイルを SD カードにコピーします。

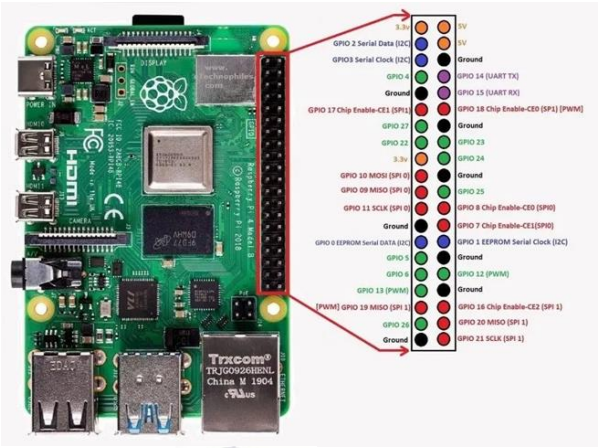
```
$ cd ~/WindRiver/wrsdk-vxworks7-raspberrypi4b/vxsdk/sdcard
$ cp  –rL *  /media/xxxx/xxxx-xxxx/
```

VxWorks カーネルイメージを SD にコピーします

```
$ cd ~/WindRiver/wrsdk-vxworks7-raspberrypi4b/bsps/rpi_4_0_1_1_0
$ cp   uVxWorks  /media/xxxx/xxxx-xxxx/
```

4) Raspberry Pi 用 TTL シリアルケーブルで USB 接続します。

Raspberry Pi 4 モデルでは、GPIO 14 および 15（UART1 、ピン 8 / GPIO14 で送信、ピン 10 / GPIO15 で受信）を使って UART 送信および受信（UART1）します。



白	GPIO14
緑	GPIO15
黒	GND

RaspberryPi と PC 間は USB-シリアルケーブルアダプターで接続します。 次に、シリアル通信プログラム（minicom や gtkterm など）を起動し、シリアル接続パラメーターを次のように構成します。

Bps/Par/Bits　　: 115200 8N1
Hardware Flow Control : No
Software Flow Control : No

Raspberry Pi に SD カードを接続して電源を入れると、SD カードにコピーされた VxWorks カーネルが自動的に起動します。

```
U-Boot 2020.07-rc2-00133-ged9a3aa645 (May 19 2020 - 12:49:43 +0300)

DRAM:  3.9 GiB
RPI 4 Model B (0xc03112)
MMC:   emmc2@7e340000: 0, mmcnr@7e300000: 1
Loading Environment from FAT... OK
In:    serial
Out:   serial
Err:   serial
Net:
Warning: genet@7d580000 MAC addresses don't match:
Address in DT is             dc:a6:32:86:d8:ef
Address in environment is    dc:a6:32:07:b3:a4
eth0: genet@7d580000
Hit any key to stop autoboot:  0
8944948 bytes read in 773 ms (11 MiB/s)
## Booting kernel from Legacy Image at 00100000 ...
   Image Name:   vxworks
   Image Type:   AArch64 VxWorks Kernel Image (uncompressed)
   Data Size:    8944884 Bytes = 8.5 MiB
   Load Address: 00100000
   Entry Point:  00100000
   Verifying Checksum ... OK
   Loading Kernel Image
   !!! WARNING !!! Using legacy DTB
## Starting vxWorks at 0x00100000, device tree at 0x00000000 ...
Target Name: vxTarget
Instantiating /tmp as rawFs,  device = 0x1


 _________      __________
 \77777777\       /77777777/
  \77777777\     /77777777/
   \77777777\   /77777777/
    \77777777\ /77777777/
     \77777777\ \7777777/
      \77777777\ \77777/            VxWorks 7 SMP 64-bit
       \77777777\ \777/
        \77777777\ \7/     Release version: SR0660
         \77777777\ -      Build date: Feb  3 2021 14:26:49
          \77777777\
           \7777777/       Copyright Wind River Systems, Inc.
            \77777/ -              1984-2021
             \777/ /7\
              \7/ /777\
               - -------

                 Board: Raspberry Pi 4 Model B - ARMv8
               CPU Count: 4
            OS Memory Size: ~3891MB
          ED&R Policy Mode: Deployed
       Debug Agent: Started (always)
          Stop Mode Agent: Not started
        BSP Status: *** UNSUPPORTED ***

 Adding 13658 symbols for standalone.
->
```

4．SDK でのサポート機能

- SD カード
- USB ストレージ
- DHCP を使ったネットワーク

5．アプリケーションの開発方法

Linux ターミナルウィンドウを開き、VxWorksSDK のインストール場所に移動しビルド用の環境変数を設定します。

```
$ source sdkenv.sh
```

アプリケーションのビルド方法

SDK 付属のサンプルを使ってビルドしましょう。

VxWorks には、カーネルモードで実行できるアプリケーションとユーザモードで実行できるアプリケーションの 2 種類があります。

SDK のインストール先にある examples を下記の場所にコピーしてください。

```
$ cd ~/WindRiver
$ mkdir opt
$ cd opt
$ cp -r ~/WindRiver/wrsdk-xworks7-raspberrypi4b/examples .
```

ユーザモードでのコンパイルは

```
$ cd examples/rtp/xxxx
$ make
```

カーネルモードでのコンパイルは

```
$ cd examples/dkm/xxxx
$ make
```

CMake を使ったアプリケーションの開発

デフォルトで、VxWorks SDK には CMake が含まれています。 アプリケーション開発者は SDK 環境を使うと、CMake に自動的にアクセスできるようになります。 CMake を個別にダウンロード、インストール、構成する必要はありません。 VxWorks CMake を使用すると、SDK に含まれている既存の CMake サンプルをビルドしたり、独自のプロジェクトを作成することもできます。

CMake の例は、/examples/rtp/hello_cmake_rtp と /examples/dkm/ hello_cmake_dkm のディレクトリにあります。

手順：1 my_project というプロジェクトディレクトリを作成します

```
$ mkdir my_project
```

1. 作成したプロジェクトフォルダーに CMakeLists.txt ファイルを作成します。
2. プロジェクトをビルドするには、次のコマンドを実行します。

```
$ cmake -D CMAKE_TOOLCHAIN_FILE=${WIND_SDK_HOME}/vxsdk/sysroot/mk/rtp.toolchain.cmake .
```

注：この例では RTP を使用しています。 DKM プロジェクトをビルドする場合は、ツールチェーン名を dkm.toolchain.cmake に変更してください。

6．アプリーケーションの実行方法

Linux ホストのユーザホームにて、ユーザ「target」でパスワード「vxtarget」にて FTP ルートを使用してポート 21 で FTP サーバーを起動します。

```
$  sudo python -m pyftpdlib -p 21 -u target -P vxTarget -d $HOME/WindRiver
```

開発ホストの IP アドレスを確認してください。

Raspberry Pi が DHCP ネゴシエーションを正常に完了すると、VxWorks リモートファイルデバイスを介してホストファイルシステムにアクセスできるようになります。

VxWorks シェルで次のコマンドを実行すると、「wrs」という名前のデバイスが作成されます。

```
-> netDevCreate ("/wrs", "192.168.10.191", 1)
```

注：192.168.10.191 を開発ホストの IP アドレス（FTP サーバーを実行しているホスト）に置き換えてください。

cmd モードシェルに切り替えて、作成した RTP の場所に移動します。これで、以前に作成したリモートファイルデバイスを介してターゲットで使用できるようになります。

実行例

```
-> cmd
[vxWorks *]# cd wrs
[vxWorks *]# pwd
wrs/
[vxWorks *]# cd opt
[vxWorks *]# ls
hello
hello.c
[vxWorks *]# more hello.c
#include <stdio.h>

int main(int argc, char **argv) {
    printf("Hello World\n");
    return 0;
}
[vxWorks *]#
[vxWorks *]# hello
Launching process 'hello' ...
Process 'hello' (process Id = 0xffff8000005e85d0) launched.
Hello world
[vxWorks *]#
```

アプリケーションの作成とデバッグの詳細については、docs ディレクトリーにあるドキュメントを参照してください。

```
$ source <SDK_DIR>/sdkenv.sh
```

7． Python アプリケーションの作成と実行

前提条件

以前に作成した SD カードには、VxWorks で Python アプリケーションを実行するためのすべての前提条件が含まれています。

cmd シェルから Python を起動して、Hello World！が表示するのを確認してください ：

```
[vxWorks *] cd /usr

[vxWorks *]# python3

Launching process 'python3' ...

Process 'python3' (process Id = 0xffff800000641380) launched.
```

```
Python 3.8.0 (default, Feb  2 2021, 19:28:52)
[Clang 10.0.1.1 (http://gitlab.devstar.cloud/compilers/llvm/clang.git 5449acbae on vxworks
Type "help", "copyright", "credits" or "license" for more information.

>>> import sys
>>> print("Hello World!")
Hello World!
>>> print (sys.platform)
vxworks
```

Python スクリプトの場合は次のようにして実行できます。

```
[vxWorks *] python3 helloworld.py


Hello World!
```

注：シェルから Python を起動する場合は、Python を起動する前に rtpSpStackSize の値を設定する必要がある場合があります（たとえば、rtpSpStackSize を 0x1000000 に設定します）。 詳細については、API リファレンスの rtpSp（）を参照してください。

```
rtpSp( )
NAME
    rtpSp( ) - launch a RTP with default options.

SYNOPSIS
long rtpSp
   (
   char * execAndArgs,       /* path to the executable file + arguments */
   int    initTaskPrio,      /* priority of RTP's initial task */
   long   userStackSize,     /* size of the initial task's user stack */
   int    launchOptions,     /* options to apply to the RTP at launch */
   int    launchTaskOptions  /* task option for the RTP's initial task */
   )
```

商用版の VxWorks に関する詳細は、https://www.windriver.com/japan/products/vxworks/をご覧ください。