



ADB3 Driver 1.1.1 for Wind River VxWorks Release Note

Introduction

This release note accompanies the ADB3 Driver for Wind River VxWorks. The latest version of this driver can be found at:

<ftp://ftp.alpha-data.com/pub/admxrcg3/vxworks>

For support, send e-mail to support@alpha-data.com

Operating systems supported

This release of the ADB3 Driver for VxWorks supports the following operating systems:

- Wind River VxWorks 5.5 and 6.x

Hardware supported

This release of the ADB3 Driver for VxWorks supports the following Alpha Data hardware:

- ADM-XRC-6TL
- ADM-XRC-6T1
- ADM-XRC-II (legacy hardware)

License agreement

Please refer to the files **license.rtf** or **license.txt** within this software package for the licensing terms that apply to this software. Please contact Alpha Data if alternative licensing terms are required.

Alpha Data reserves the right to use different licensing terms for future releases of this software.

Building the driver

Prerequisites for building the driver are a Linux or Windows host machine with either Tornado 2.2 & Vxworks 5.5 or Workbench & VxWorks 6.x installed on it.

The driver is supplied in source code form so that it can be cross-built for a variety of CPU architectures and hardware platforms. To build the driver, follow the instructions in the appropriate subsection.

Cross-building the driver on a Windows host

To build the driver on a Windows host, follow these steps:

1. Unpack this package somewhere, for example

```
C:\MyTesting\adb3_drv-1.1.1
```

For convenience, the remainder of this document refers to this directory as %ROOT% (although it should be noted that no such environment variable is created nor referenced by the driver's build system).

2. Start a command prompt that is capable of performing command-line VxWorks builds. For VxWorks 6.x, use the "VxWorks Development Shell" shortcut. For VxWorks 5.5, it is necessary to start a normal command prompt, and then execute the **torVars.bat** batch file that can normally be found in

```
C:\Tornado2.2\host\x86-win32\bin
```

3. In the command prompt, change directory to %ROOT% from step 1.

4. Execute MAKE with the appropriate options, as described in "MAKE options". For example, to build a debug VxBus driver for a SMP Pentium 4 system, use

```
make CONFIG=p4-vxbus-6.7 VSB=smp clean all
```

This uses the predefined configuration **p4-vxbus-6.7**. Assuming the build is successful, the binaries are:

```
%ROOT%\driver\monolithic\vxworks\bin\p4-vxbus-6.7\debug_smp\adb3Driver.out
%ROOT%\api\modules\admxrc3\vxworks\bin\p4-vxbus-6.7\debug_smp\admxrc3Api.out
```

At this point, you are ready to proceed to starting the driver as described in "Starting the driver".

Cross-building the driver on a Linux or UNIX host

To build the driver on a Linux or UNIX host, follow these steps:

TBA

MAKE options

The top-level Makefile for the ADB3 VxWorks driver accepts a number of options which are passed on the MAKE command line. These are:

- **CONFIG=<configuration>**
Specifies a predefined configuration defined by the file

```
%ROOT%\driver\monolithic\vxworks\rules.<configuration>
```

The rules file may contain any of the following options; for an example, see

```
%ROOT%\driver\monolithic\vxworks\rules.p4-vxbus-6.7
```
- **CPU=<cpu>**
Specifies the CPU being targetted; for example PPC604 or PENTIUM4 (default). Must be appropriate for the TARGET option.
- **DEBUG=<false|true>**
Specifies a release or debug (default) build.
- **EXTRA_CCOPTS=<extra compiler options>**
Specifies extra C compiler options.
- **EXTRA_LDOPTS=<extra linker options>**
Specifies extra linker options.
- **TARGET=<target spec>**
Defines the target specification, which must be appropriate for the CPU option. Examples of valid target specifications for the DIAB toolchain are **-tPPC604FH:vxworks55** (PowerPC) and **-tPENTIUM4LH:vxworks67** (default, Pentium 4). Examples of valid target specifications for the GNU toolchain are **-mcpu=604** (PowerPC 604) and **-mtune=pentium4 -march=pentium4** (Pentium 4).
- **TOOLCHAIN=<diab|gnu>**
Specifies the toolchain to be used to build the driver; legal values are **diab** (default) or **gnu**. If the **gnu** toolchain is selected, the following additional options must be specified (which can be in the rules file specified by the CONFIG option, for convenience):
 - **CC=<compiler>**
Specifies the C compiler; must be appropriate for the CPU and TARGET options. For example, **ccppc** selects the PowerPC GNU compiler.
 - **LD=<linker>**
Specifies the linker; must be appropriate for the CPU and TARGET options. For example, **ldppc** selects the PowerPC GNU linker.
 - **NM=<object dumper>**
Specifies object dumper; must be appropriate for the CPU and TARGET options. For example, **nmpcc** selects the PowerPC GNU object dump utility.

- **TYPE=<legacy/vxbus>**
Specifies whether the driver should be built as a legacy driver or a VxBus driver (default).
- **VSB=<variant>**
Specifies variant libraries, if required. If omitted, the normal libraries are used. The most common value for this option is **smp**.

When the **CONFIG** option is specified, the driver's build system reads a rules file that contains values for the other options. For example, the configuration **hcd5220-6.7** has a rules file

```
%ROOT%\driver\monolithic\vxworks\rules.hcd5220-6.7
```

This results in a VxBus driver targeting the Mercury HCD5220 single-board computer with SMP libraries. By way of illustration, this rules file contains:

```
CPU=PPC604
VSB=smp
TYPE=vxbus
ifeq ($(TOOLCHAIN),diab)
EXTRA_CCOPTS=-Xcode-absolute-far -Xdata-absolute-far -DHCD5220
TARGET=-tPPC604FH:vxworks67
else
ifeq ($(TOOLCHAIN),gnu)
EXTRA_CCOPTS=-mlongcall -DHCD5220
CC=ccppc
LD=ldppc
NM=nmpc
TARGET=-mcpu=604
else
$(error *TOOLCHAIN $(TOOLCHAIN) not recognized.*)
endif
endif
```

If no **CONFIG** option is specified, the default configuration is **default**. The **rules.default** file contains:

```
CPU=PENTIUM4
TYPE=vxbus
ifeq ($(TOOLCHAIN),diab)
TARGET=-tPENTIUM4LH:vxworks67
else
ifeq ($(TOOLCHAIN),gnu)
CC=ccpentium
LD=ldpentium
NM=nmpentium
TARGET=-mtune=pentium4 -march=pentium4
else
$(error *TOOLCHAIN $(TOOLCHAIN) not recognized.*)
endif
endif
```

It is possible that none of the predefined configurations supplied by Alpha Data is appropriate for your hardware platform. If this is the case, a new configuration can be created by using one of the existing rules files in

```
%ROOT%\driver\monolithic\vxworks
```

as a template, and modifying it appropriately.

Starting the driver

To start the driver in the target system, follow these steps:

1. Download the modules **adb3Driver.out** and **admxcrc3Api.out** to the target system. This can be done using the **ld** command in the VxWorks shell or the target system's console. For example:

```
-> ld <hostname:C:/MyTesting/adb3_drv-1.1.1/driver/monolithic/vxworks/bin/
ppc604-5.5/debug/adb3Driver.out
-> ld <hostname:C:/MyTesting/adb3_drv-1.1.1/api/modules/admxcrc3/vxworks/bin/
```

```
ppc604-5.5/debug/admxrc3Api.out
```

- To start the driver, use the entry point **adb3DrvStart**:

```
-> adb3DrvStart
```

This entry point accepts two parameters:

- debugLevel** (int), default 0
Verboseness of debug output sent to console using **logMsg**. The release version of a driver produces no output. In the debug version of the driver, a value of 0 results in minimal output and increasing values (up to 10) result in more output.
- bLegacyHardware** (int), default 0
Nonzero to enable support for legacy hardware such as the ADM-XRC-II.

For example, to start the driver with some extra debug output and support for legacy hardware, use:

```
-> adb3DrvStart(2,1)
```

A **debugLevel** value greater than zero in the debug version of the driver may greatly slow down execution of the driver, so 0 is recommended during normal usage.

Starting the driver with a **debugLevel** of 0 should result in output of the following form on the console:

```
-> adb3DrvStart(0,1)
0xf68b790 (tShell): adb3: dfDriverEntry: ADB3 Monolithic Driver, version=1.1.1.4
0xf68b790 (tShell): adb3: identifyPci9656: identified ADM-XRC-II
```

Known issues

Fixed-local addressing DMA transfers

The flag **ADMXRC3_DMA_FIXEDLOCAL** currently has no effect for the ADM-XRC-6TL and ADM-XRC-6T1 when used with the DMA functions in the ADMXRC3 API.

Compiler warnings during builds

The following compiler warnings may be generated when building the driver for certain configurations:

- "../core/admxrc6tx_common.c", line 314: warning (dcc:1546): dangerous to take address of member of packed or swapped structure
This warning may occur more than once in multiple source files, but can be ignored.
- "../framework/vxworks/legacy_methods.c", line 154: warning (dcc:1500): function **pciIntDisconnect2** has no prototype
This warning can be ignored, and occurs because the **pciIntDisconnect2** function appears to be missing from the VxWorks 5.5 header files, although it exists and can be used in some VxWorks kernel images.

Release history

Release 1.1.1

This is the first release of the ADB3 Driver for VxWorks.