ADB3 Driver 1.4.14 for Linux Release Note

Introduction

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

ftp://ftp.alpha-data.com/pub/admxrcg3/linux

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

Operating systems supported

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

- GNU/Linux distribution with 2.6.x kernel
- GNU/Linux distribution with 3.x.y kernel

Due to the ever-changing nature of GNU/Linux, Alpha Data cannot guarantee that this driver can be successfully configured, built, installed and run on all Linux distributions past, present and future. Alpha Data makes best-efforts to ensure compatibility with all Linux distributions, but should a problem be encountered, please contact support@alpha-data.com.

Hardware supported

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

- ADM-XRC-6TL
- ADM-XRC-6T1
- ADM-XRC-6T-DA1
- ADM-XRC-6TGE and ADM-XRC-6TGEL
- ADM-XRC-6T-ADV8
- ADPE-XRC-6T and ADPE-XRC-6T-L
- ADPE-XRC-6T-ADV
- ADM-XRC-7K1
- ADM-XRC-7V1
- ADM-VPX3-7V2

License Agreement

This release of software is licensed according to the terms of GNU Public License Version 2 (GPL V2). A copy of this license can be found in the file **gpl-2.0.txt** within this software package. Please contact Alpha Data if alternative licensing conditions are required.

Alpha Data reserves the right to use a different license agreement for future releases of this software.

Installation instructions

This release of the driver is distributed in source code form as a tarball (.tar.gz file extention). Please refer to the README file inside the tarball for instructions on how to configure, build and install the driver.

Completely uninstalling the driver

To uninstall the driver, first stop the driver by issuing the command **rmmod adb3**. Then delete the following files and symbolic links, if they exist:

1. /usr/lib/libadmxrc3.*, /usr/lib/libadb3.*
2. /usr/lib64/libadb3.*, /usr/lib64/libadb3.* (if they exist)
3. /lib/modules/<kernel version>/kernel/drivers/addon/adb3/adb3.ko
4. /etc/init.d/adb3
5. /etc/udev/rules.d/51-admxrc3.rules, /etc/udev/rules.d/51-adb3.rules

udev support

Beginning with release 1.2.1, the script **adb3.rc** provided for starting previous versions of the driver is not required, except on Linux systems that do not have the **udev** subsystem. As of release 1.2.1, when the kernel module **adb3.ko** is loaded, the **udev** subsystem automatically creates device nodes **/dev/admxrc3*** and **/dev/adb3***.

In order for this to work correctly, the driver must have been installed in the normal way (that is, by executing a command such as **make all install** as root). The install procedure copies the **udev** rules files **51-admxrc3.rules** and **51-adb3.rules** to **/etc/udev/rules.d/**.

After installation, and before loading **adb3.ko**, the rules file **51-admxrc3.rules** may be customized to change the permissions and/or owner and/or group of the device nodes **/dev/admxrc3***. Please see the comments in **/etc/udev/rules.d/51-admxrc3.rules** after installing the driver for further details.

NOTE: The Makefile for the driver still installs the script **adb3.rc** in **/etc/init.d/**, but it should not be used unless there is no **udev** subsystem.

VPD write-protection mechanism

The VPD write-protection mechanism described in the ADM-XRC Gen 3 SDK User Guide is implemented as of release 1.1.0. To enable write-to-VPD, the kernel module parameter **EnableVpdWrite** must be nonzero.

This value takes effect when the driver starts, so it can be changed only by unloading the driver and restarting the driver with a different value for **EnableVpdWrite**. If this parameter is not specified, the driver considers it to be zero (write-to-VPD disabled).

Security considerations

By default, the **udev** rules file **51-admxrc3.rules** creates device nodes in **/dev** as follows:

- Mode: 664 (octal) => owner read, owner write, group read, group write, other read
- UID: **root**
- GID: **root**

This means that the default permissions are that (i) only **root** and members of the group **root** can open devices, in read-only or read-write mode and (ii) users that are not members of the group **root** can open a device in read-only mode. However, after installing the driver, the file **/etc/udev/rules.d/51-admxrc3.rules** can be customized to relax permissions. See the comments in that file for details.

Common buffer support

Beginning with release 1.4.4, the driver can create one or more "common buffers" at startup. The main purpose of this feature is supporting applications that use Direct Master data transfer, such as an ethernet-style I/O interface where the sizes and arrival times of packets of data are not known in advance by software running on the host. These buffers have the following characteristics:

- Persist until the driver is stopped.
- Guaranteed aligned to a specified power-of-2 address boundary.
- Allocated from the appropriate pool of memory in order to be contiguous and visible to bus-master devices.
- Can be mapped into the virtual address space of a user-mode process; see "ADMXRC3 API Specification 1.5.0" or later for details of the new ADMXRC3 API functions relating to common buffers.

The driver parameter **PrimaryCommonBufferCount** determines the number of common buffers allocated; the default is zero, meaning that no common buffers are allocated by default. The parameter **PrimaryCommonBufferSizeLow** determines the size in bytes of each common buffer; the default is 64 kiB (0x10000). The parameter **PrimaryCommonBufferAlignment** determines the address boundary size to which each common buffer is guaranteed to be aligned; the default is 16 bytes (0x10).

Biarchitecture support and shared library installation

As of release 1.2.0, the "configure" script for ADB3 Driver for Linux selects a non-biarchitecture build by default. In other words, by default, only native binaries are built when the "make clean all" command is issued. In non-biarchitecture systems, the system library directory is typically **/usr/lib**. This presents no problem for the "make install" command, and it places the shared library "libadmxrc3.so" in **/usr/lib**.

If the target system is biarchitecture, such as x86_64 Linux, passing "-biarch yes" to the "configure" script selects a biarchitecture build where 64-bit native binaries and 32-bit compatibility binaries are built when the "make clean all" command is issued.

In biarchitecture systems, there are multiple different conventions for where 64-bit and 32-bit shared libraries are located:

- The Redhat / Fedora / CentOS Linux convention where 64-bit native libraries are in **/usr/lib64** and 32-bit compatibility libraries are in **/usr/lib**.
- The Ubuntu Linux convention where 64-bit native libraries are in **/usr/lib** and 32-bit compatibility libraries are in **/usr/lib32**.

As of release 1.4.5 of ADB3 Driver for Linux, the "configure" script can detect which of the above conventions is in use in the target system. It generates the ".build_defs" file accordingly so that "make install" will install shared libraries in the correct locations.

Release 1.4.4 or earlier of ADB3 Driver for Linux assumes that the target system uses the Redhat / Fedora / CentOS Linux convention, which is an incorrect assumption for some Linux distributions. Therefore, it is recommended that release 1.4.4 or earlier is not used on a system that does not follow the Redhat / Fedora / CentOS Linux convention, unless manually installing the shared libraries after doing "make clean all".

Known issues

Downgrading to an earlier version

When downgrading to an earlier version of the driver, remove all files named **/usr/lib/libadmxrc3.*** and **/usr/lib/libadb3.***, (and **/usr/lib64/libadmxrc3.*** and **/usr/lib64/libadb3.*** if on a 64-bit bi-architecture machine), before executing the **make install** command as root. Otherwise, the shared libraries remaining from the later version of the driver will be preferred by the system as they have a higher version number.

Additionally, delete any device nodes **/dev/admxrc3*** before installing the earlier driver version, as a safety precaution. This is recommended because earlier versions of the driver have different permissions mechanisms when opening devices.

Fixed-local addressing DMA transfers

The flag ADMXRC3_DMA_FIXEDLOCAL when used with the DMA functions in the ADMXRC3 currently has no effect for Gen 3 hardware.

Release history

This release implements ADMXRC3 API Specification version 1.7.1.

Release 1.4.14

Corrections:

1. On models using the LM87 system monitor chip (ADM-XRC-6TL, ADM-XRC-6T1, ADM-XRC-6TGE, ADM-XRC-6TGEL and ADM-XRC-6TDA1), fixed temperatures for sensors 8 & 9 reading as unexpectedly large positive values when the actual temperature reading (in degrees Celsius) is negative. Values for sensors 8 & 9 now read correctly as negative values when the actual temperature is negative.
2. The description of the sensor whose description was "Time since manufacture" has been corrected, for all applicable models except for the ADM-XRC-6T-ADV8, to read as "Total powered on time". This sensor actually reports the total time spent powered on since manufacture, but the description in previous driver versions incorrectly indicated that it was the time since manufacture.
3. The description of the sensor whose description was "Time since manufacture" has been corrected, for the ADM-XRC-6T-ADV8, to read as "Total 3.3VAux on time". This sensor actually reports the total time spent with 3.3VAux powered on since manufacture, but the description in previous driver versions incorrectly indicated that it was the time since manufacture.

Release 1.4.13

Corrections:

1. Improved arithmetic accuracy of calculations that determine initial SI5338 output frequencies at driver startup, and fixed some anomalous frequencies that are generated when certain SI5338 output frequencies are requested via calls to ADMXRC3_SetClockFrequency. Affects following models: ADM-XRC-6TGE, ADM-XRC-6TGEL, ADPE-XRC-6T & ADPE-XRC-6T-L.

Release 1.4.12

This release implements ADMXRC3 API Specification version 1.7.0.

Enhancements:

1 Added support for ADM-VPX3-7V2 board.

2 Added support for ADM-XRC-6TGEL board.

Release 1.4.10

This release implements ADMXRC3 API Specification version 1.6.0.

Enhancements:

1 Added support for message-signalled interrupts via **PciUseMsi** driver parameter (default 0), where a nonzero value causes MSI to be used. To enable MSI, load the driver with e.g. **modprobe adb3 PciUseMsi=1**.

2 Added new ADMXRC3 API functions: ADMXRC3_GetDeviceStatus and ADMXRC3_ClearDeviceErrors.

3 Added support for ADM-XRC-7K1 revision 2 board.

4 Added support for ADM-XRC-7V1 revision 2 board.

5 Added support for devices implemented using generic ADB3 core in target FPGA (Vendor ID = 0x4144, Device ID = 0xADB3, Subsystem Vendor ID = 0x4144, Subsystem Device ID = 0x0000).

6 When a generic ADB3 device (such as an ADB3 core in a target FPGA) is found, now counts number of DMA engines by examining registers in BAR0, instead of assuming 0. This means that DMA engines are exposed by the ADMXRC3 API when available.

Corrections:

7 Corrected sensor names for ADM-XRC-7V1.

8 Fixed compile error with Linux 3.7 kernel or later, due to missing symbol **VM_RESERVED**. Also changed memory-mapping behavior of the driver:

- The driver now requests (to the Linux kernel) that regions mapped into a process' address space by **ADMXRC3_MapWindow** are not core-dumped.
- However, for common buffers (that is, real memory) mapped into a process' address space by **ADMXRC3_MapCommonBuffer**, the driver's behaviour is unchanged (i.e. it does not request that such mapped regions are not core-dumped).

9 Fixed a problem with x86_64 Gentoo Linux where the presence of the symlink /usr/lib -> lib64 confused the "configure" script, when the "-biarch" option was used, so that the 32-bit shared libraries were erroneously installed into the 64-bit library directory (/usr/lib64) instead of the directory /usr/lib32.

Release 1.4.9

This release implements ADMXRC3 API Specification version 1.5.1.

Enhancements:

1 Added support for the ADM-XRC-6T-DA1 and ADM-XRC-7K1.

2 Added preliminary support for the ADM-XRC-7V1.

Corrections:

3 The implementation of ADMXRC3_MapWindow and ADMXRC3_UnmapWindow has been revised in order to eliminate a kernel panic that occurs when an application maps a window using ADMXRC3_MapWindow and then invokes fork() to create a child process (the kernel panic occurs when the child process terminates). The new implementation is compatible with the manner in which fork() replicates the parent's virtual address space into the child process, so that ADMXRC3_UnmapWindow works as expected in both

the parent and child process.

4 On the ADPE-XRC-6T(-L), the debug message about an AVR timeout (visible in system log) has been eliminated. The driver now correctly gets the AVR uC firmware version from the AVR uC instead of timing out.

5 On the ADPE-XRC-6T(-L), if a board has a value of 0 (which is always invalid) in VPD for the SI5338 reference clock frequency, works around it by changing it in the in-memory copy of the data to 25000000.

6 On the ADM-XRC-6T-ADV, the second bank of Flash memory (dedicated to target FPGA 1) can now be accessed successfully.

7 Fixed an unkillable thread hang that can occur when calling ADMXRC3_ReadVPD with a VPD addresses of 0x100 or above on the ADPE-XRC-6T(-L), ADM-XRC-6T-ADV8 and ADPE-XRC-6T-ADV.

Release 1.4.6

This release implements ADMXRC3 API Specification version 1.5.0.

Enhancements:

1 Added preliminary support for ADPE-XRC-6T-ADV.

Corrections:

2 Fixed a crash resulting from a regression bug in the 1.4.5 release that results in blocking API calls sometimes failing to wait correctly.

3 Added support for FMCs fitted to ADPE-XRC-6T(-L). Now reports FMC information via ADMXRC3_GetModuleInfo for ADPE-XRC-6T(-L) as expected when an FMC is fitted.

4 ADMXRC3_ReadSensor now correctly reports the values of sensors whose values are negative (e.g. a temperature sensor whose reading is less than 0 deg. C) instead of a large positive value.

5 Corrected name of sensor 1 on ADM-XRC-6T-ADV8; was "12V supply rail", now "5V/12V XMC VPWR rail"; setting driver parameter "Admxrc6tadv8CompatSensor1Name" to 1 overrides this and causes the old name to be returned by the driver.

6 On the ADPE-XRC-6T(-L), fixed clock generator 1 being incorrectly mapped to SI5338 multisynth 0; now mapped to multisynth 1.

Release 1.4.5

This release implements ADMXRC3 API Specification version 1.5.0.

Corrections:

1 Fixed a crash that can occur when ADMXRC3_Unlock is called with an invalid value for the ADMXRC3_BUFFER_HANDLE parameter.

2 Fixed a crash when multiple queued DMA transfers are cancelled, by ADMXRC3_Cancel or by killing threads, within a small window of vulnerability around to the point at which the DMA transfer would complete normally were it not cancelled.

3 Detection in "configure" script of which system library directories are present is now aware of several different conventions. Refer to "Biarchitecture support and shared library installation" for details.

- Non-biarchitecture systems which have **/usr/lib** only. This is typically the case for x86 (32-bit) Linux and 32-bit PowerPC Linux.
- Biarchitecture systems which follow the Redhat / Fedora / CentOS style of 64-bit native libraries in **/usr/lib64** and 32-bit compatibility libraries in **/usr/lib**.

- Biarchitecture systems which follow the Ubuntu style of 64-bit native libraries in **/usr/lib** and 32-bit compatibility libraries in **/usr/lib32**.

Release 1.4.4

This release implements ADMXRC3 API Specification version 1.5.0.

Enhancements:

1. Added support for Linux 3.x.y kernels.
2. Added common buffer functionality with the following ADMXRC3 API functions:
   - ADMXRC3_GetCommonBuffer
   - ADMXRC3_GetCommonBufferCount
   - ADMXRC3_MapCommonBuffer
   - ADMXRC3_UnmapCommonBuffer

Corrections:

3. Fixed a race condition that could cause a crash every few hours of constant large DMA transfers (of size 64 physical pages or more) in a typical SMP machine.
4. Fixed a regression bug introduced in the 1.4.3 release that results in a small probability of ADMXRC3_FinishDMA failing to return, due to a race condition.
5. Corrected the scaling factors for sensors 1 to 10 for the models ADPE-XRC-6T and ADPE-XRC-6T-L.

Release 1.4.3

This release implements ADMXRC3 API Specification version 1.4.0.

Corrections:

6. Fixed a crash that can occur when attempting to do two or more DMA transfers on the same DMA channel.
7. Fixed a crash that can occur if the driver is somehow called by a libadmxrc3*.so from a different driver version, due to the handlers for ADMXRC3_GetSensorInfo and ADMXRC3_ReadSensor failing to properly validate arguments.
8. Fixed an issue specific to the ADM-XRC-6T-ADV8 where the driver emitted the debug message "*** avrInit: failed to get AVR uC firmware version".
9. Fixed a crash whose frequency depends on the rate of calling poll() or select() on an ADB3 device when doing small non-blocking DMA transfers.
10. Added preliminary support for the models ADPE-XRC-6T and ADPE-XRC-6T-L.

Release 1.4.1

This release implements ADMXRC3 API Specification version 1.4.0.

New behavior:

1. Default permissions for /dev/admxrc3* device nodes are now 664 (in previous releases they were 660). This brings the ADB3 Linux driver's behaviour into line with the ADB3 Windows driver.

Corrections:

2. Fixed a bug in the SI5338 clock synthesizer code for the ADM-XRC-6TGE that could corrupt memory

when programming clock index 4. This clock generator is only available when the Si5338ExposeAllClocks driver parameter is nonzero; by default it is not available.

3 Support for ADM-XRC-6T-ADV8 is now feature-complete; added support for programming VPD and reading system monitor sensors via ADMXRC3 API.

4 Fixed a number of PowerPC-specific build issues, so that cross-building using Embedded Linux Development Kit is considerably simpler than before.

Release 1.3.1

This release implements ADMXRC3 API Specification version 1.3.0.

New behavior:

1 Added support for the ADM-XRC-6TGE.

2 Added preliminary support for the ADM-XRC-6T-ADV8.

Enhancements:

3 Now exposes (via the ADMXRC3 API) a programmable clock generator with index 0 on the ADM-XRC-6T1 when it has firmware 1.6 (PCI revision 0x06) or later.

Corrections:

4 ADMXRC3_GetClockFrequency now correctly returns the current clock frequency for a given clock generator. The driver now interrogates the hardware at startup to determine the current frequencies generated by each clock generator, so that ADMXRC3_GetClockFrequency can return the correct frequency even before any call to ADMXRC3_SetClockFrequency.

5 ADMXRC3_GetClockFrequency now correctly validates the pointer argument (3rd argument) passed to it, and returns ADMXRC3_NULL_POINTER if it is NULL.

6 Corrected the maximum frequency allowed for the clock generator with index 0 on the ADM-XRC-6TL. Previously, the driver incorrectly permitted frequencies up to 210 MHz to be requested, whereas 140 MHz is the correct maximum frequency.

7 Fixed ADMXRC3_EraseFlash failing to correctly validate the region specified to ensure that it is wholly within the unprotected region of a Flash memory bank.

Release 1.2.1

This release implements ADMXRC3 API Specification version 1.2.0.

New behavior:

1 Added **udev** support. Uses the **udev** subsystem for creating device nodes in **/dev/**; the script **adb3.rc** is no longer required, except on Linux distributions that do not use **udev**.

2 For ADM-XRC-6TL, now recognizes "Extended" temperature range value (2) in VPD at offset 0x3E.

3 For ADM-XRC-6T1, now recognizes "Extended" temperature range value (2) in VPD at offset 0x42.

Corrections:

4 Fixed a race condition that occurs in (at least) Centos 5.6, between the default catch-all **udev** rule and the **adb3.rc** script. This race may result in unpredictable permissions for the **/dev/admxrc3*** device nodes. As of release 1.2.1, **udev** is used for creating device nodes and the **adb3.rc** script is obsoleted except for Linux distributions that do not use **udev**.

This issue affects any Linux distribution that has a catch-all **udev** rule which automatically creates device

nodes in **/dev** for devices that do not match any other rule. CentOS 5.6 is among the Linux distributions that show this behaviour.

5 Fixed a regression bug in the "configure" script of the 1.2.0 release in detecting the correct directory in which to install the **libadmxrc3.so** and **libadb3.so** shared libraries. In 1.2.0, if a non-biarchitecture build and install is performed on a bi-architecture machine, the 64-bit native libraries are (incorrectly) installed in **/usr/lib/** instead of (correctly) in **/usr/lib64/**.

Release 1.2.0

This release implements ADMXRC3 API Specification version 1.2.0.

New behavior:

1 Bi-architecture build is now not performed by default, as most 64-bit Linux distributions do not install (by default) the necessary compatibility packages for building 32-bit binaries. To build both 32-bit and 64-bit binaries for the API libraries, specify '-biarch yes' when running the 'configure' script.

Corrections:

2 Fixed some build problems relating to kernel features in RHEL / CentOS 5.5 kernels (derived from Linux 2.6.18). The 'configure' script now detects those kernel features correctly.

3 Fixed a thunking issue that may occur if the 32-bit and 64-bit C compilers on the same bi-architecture-capable system have different struct packing rules, where certain ADMXRC3 API calls made by a 32-bit executable running under a 64-bit Linux kernel fail with ADMXRC3_UNKNOWN_ERROR.

4 Fixed a problem where, when a process closes a device handle that has ongoing non-blocking operations, the process may crash due to incorrect cleanup being performed by the driver.

Enhancements:

5 Added new API functions for performing DMA transfers with 64-bit local addresses:

- ADMXRC3_ReadDMAEx
- ADMXRC3_ReadDMALockedEx
- ADMXRC3_StartReadDMAEx
- ADMXRC3_StartReadDMALockedEx
- ADMXRC3_StartWriteDMAEx
- ADMXRC3_StartWriteDMALockedEx
- ADMXRC3_WriteDMAEx
- ADMXRC3_WriteDMALockedEx

6 Added support for new sensors in ADM-XRC-6TL and ADM-XRC-6T1 with firmware 1.4 or later. This provides additional sensors that show internal temperature and voltages in the PCI Express to OCP Bridge.

7 Changed the way that permissions are checked on opening a device to allow for a more flexible system that can be customized via the **adb3.rc** script. See the section "Security considerations" and the comments in the file driver/adb3.rc for details.

8 Made the driver's build system more compatible with cross-compilation, now using the same ARCH and CROSS_COMPILE environment variables as the Linux kernel. See the README file for details.

Release 1.1.2

Corrections:

1. Fixed a user-mode memory leak that can occur when the API functions ADMXRC3_LoadBitstreamA and ADMXRC3_LoadBitstreamW return a failure status code.
2. Fixed a kernel-mode memory leak that can occur when a device handle is closed when at least one non-blocking operation has not been finished.
3. Fixed a bug in ADMXRC3_SetClockFrequency where on failure, the wrong status code was returned.

Release 1.1.0

Corrections:

1. Fixed a potential memory corruption issue on some architectures due to incorrect byte size calculation in device context structure.
2. Fixed a memory leak related to CFI Flash functionality when stopping and restarting the driver.
3. Fixed user VPD area (VPD space address range 0x100000-0x1FFFFF) not being accessible on ADM-XRC-6TL and ADM-XRC-6T1.
4. Corrected error codes returned by several ADMXRC3 API functions when invalid parameters are passed:
    1. Non-Locked DMA functions now return 'ADMXRC3_INVALID_BUFFER' if the 'pBuffer' and/or 'length' parameters are invalid.
    2. Locked DMA functions now return 'ADMXRC3_INVALID_BUFFER_HANDLE' if the 'hBuffer' parameter is invalid.
    3. ADMXRC3_Unlock now returns 'ADMXRC3_INVALID_BUFFER_HANDLE' if the 'hBuffer' parameter is invalid.
5. Fixed 'ADMXRC3_Unconfigure' failing with 'ADMXRC3_NOT_OWNER' even when the target FPGA has no owner.
6. Fixed the ADMXRC3 DMA functions not properly co-validating the 'local' and 'length' parameters.
7. Fixed the ADMXRC3 Locked DMA functions not properly co-validating the 'offset' and 'length' parameters.
8. Fixed the ADMXRC3 nonblocking DMA functions incorrectly returning 'ADMXRC3_SUCCESS' instead of 'ADMXRC3_PENDING' when 'length' is zero and all other parameters are OK.
9. Fixed the ADMXRC3 non-Locked DMA functions not properly co-validating the 'pBuffer' and 'length' parameters.
10. Fixed certain ADMXRC3 API functions not trapping NULL pointers being passed, typically resulting in an application crash.
11. Fixed a bug in driver parameter handling so that driver parameters are now correctly applied and have correct default values.
12. Implemented VPD write-protection mechanism (now protected by default).
13. Added workaround for 4k crossing issue in ADB3 Bridge rev 0x01 and earlier. Workaround is not applied for rev 0x02 or later.

Enhancements:

14. Added support for ADM-XRC-6T1.
15. Added API function ADMXRC3_OpenEx, which allows an unprivileged process to open a device in "read only" mode and use a subset of the ADMXRC3 API functions.

16 Added API functions ADMXRC3_StartNotificationWait and ADMXRC3_FinishNotificationWait, which allow Linux applications to wait for events (since there is no Linux equivalent of ADMXRC3_RegisterWin32Event).

17 Added API function ADMXRC3_GetCardInfoEx (with structure ADMXRC3_CARD_INFOEX), which returns a superset of data supplied by ADMXRC3_GetCardInfoEx.

18 Added diagnostic API functions ADMXRC3_GetSensorInfo and ADMXRC3_ReadSensor, with associated types and structures. These functions allow applications to monitor the health of a Gen 3 reconfigurable computing card.

Release 1.0.0

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

Page Intentionally left blank

Address: 4 West Silvermills Lane,
Edinburgh, EH3 5BD, UK
Telephone: +44 131 558 2600
Fax: +44 131 558 2700
email: sales@alpha-data.com
website: http://www.alpha-data.com

Address: 3507 Ringsby Court Suite 105,
Denver, CO 80216
Telephone: (303) 954 8768
Fax: (866) 820 9956 toll free
email: sales@alpha-data.com
website: http://www.alpha-data.com