Movidius
an Intel company

moviUsbBoot

Manual

Version 00.88.0 / 2017-12-15

Table of Contents

Movidius
an Intel company

Copyright and Proprietary Information Notice

Movidius Ltd.
1730 South El Camino Real, Suite 200
San Mateo, CA 94402
http://www.movidius.com/

Movidius
an Intel company

1. Introduction

1.1. Tool Overview

moviUsbBoot is a command line utility for communicating with the MA2X5X ROM. Its main use is to transmit an application MVCMD file to boot the application, however it also provides functions to read data back from the ROM.

1.2. Other relevant documents

This sections describes some other documents that user may wish to refer to.

moviToolsOverview.pdf – gives overview of all tools, conventions used, directory structure etc.
MDKProgrammersGuide.pdf – details the Myriad programming environment

2. Overview

2.1. Features

Supports transfer of an `.mvcmd` file to the `ma2x5x` chip for booting.

3. Command Line Reference

3.1. Invocation

```
moviUsbBoot <options> <Command list>
```

3.2. Options & Parameters

The <options> start with a - character and may contain any of the switches described in the table below (the case of the letters is significant).

Switch	Description
-h	display help
-v <vid>	Specify VID (default: 0x03e7)
-p <pid>	Specify PID (default: 0x2150)
-g	Find device by GUID instead of VID/PID
-I <address>	Specify the address of the device to use (as listed by -S)
-S	Display a list of all the devices that match the VID/PID
-s <size>	Specify maxmium readback size in KiB (default: 1024 KiB)
-n	Always write an empty packet after writing a file
-l	Display hexdump of all data sent to device
-b	Output readback data directly, do not display as hex. Note that if enabled, messages will be output to stderr
-t <ms>	Set timeout for bulk reads (device to host) (default is 1000ms)
-T <ms>	Set timeout for bulk writes (host to device) (default is 10000ms)
-i <ms>	Set different timeout for initial bulk read (device to host)
-N <ms>	Set timeout for empty packet writes (default is 1000ms)
-B <size>	Specify stdin buffer size in KiB (default: 1024 KiB)
-x <kb>	Specify maximum transfer chunk in KiB (default: 1024 KiB)
-a	Waits for device before starting
-A	The same as -a, but when finished, waits for disconnection and reconnection, then re-executes command list
-c <val>	Sets first connect timeout in 200ms units (0 for no timeout). Default is 50 (10000 ms)
-r <val>	Sets reconnect timeout in 200ms units (0 for no timeout). Default is 50 (10000 ms)
-q	Do not output any messages (except errors)
-L	Verbose mode (loud, twice to enable debug)
-E	Ignore errors (where possible)
-V	Output version number and exit

The command list includes the following commands:

Switch	Description
`<file.mvcmd>`	Writes the specified .mvcmd file to the device
`-`	Writes contents of stdin to the device
`:null`	Write an empty packet to the device
`:rb`	Reads back data from the device and output the data to stdout in hex or binary format (see `-b`)
`:rbq`	Reads back data with no data output
`:rbc`	Reads back data and outputs a `CRC32`
`:rbr`	Reads back data and outputs a `ROM`-style checksum
`:re`	Close and re-open the device (waiting if necessary)

4. Usage guidelines

4.1. Pre-usage setup

Windows:

Windows 8 or later: No driver files are needed. Windows will automatically use the built-in WinUSB driver when the device is first attached.

Windows 7 or earlier: The driver should be installed from the driver package in the windriver directory. Drivers can be installed using the Movidius MA2X5X Install tool (choose 32-bit or 64-bit depending on your system type), or by choosing the Update driver option in Device Manager.

Linux:

libusb-1.0.20 (or later) must be installed. This can be done on Debian systems using:

```
sudo apt-get install libusb-1.0
```

For other systems, a similar command for the relevant package manager should be used.

Board:

Please see the MDK user guide for instructions on how to configure the proper boot selection settings for the board that is being used.

4.2. Usage

moviUsbBoot uses POSIX style command line arguments. A full list of arguments will be output when no arguments are specified, or the -h argument is provided. Given an application MVCMD called app.mvcmd, the general usage will be:

```
moviUsbBoot app.mvcmd
```

Under Windows, you may want to use the WinUSB GUID (Globally Unique Identifier) feature to find the device, instead of the VID/PID. This can be enabled using the -g option. This feature is unavailable under Linux. As well as files to be transmitted to the device, moviUsbBoot can be provided with other commands. These allow for reading of data from the device, reconnecting, or transmitting the contents of the standard input. If multiple MA215X devices are connected to the host computer, the -I argument can be used to specify an index. Other options allow for override of the default VID and PID, specification of buffer sizes, various timeouts, and other output and behavior options. The -a and -A options enable looping over the full command list. This can be used to automate boot operations.

5. Source code distribution

The source code for moviUsbBoot is available as a tarball within the Movidius tools package at:

common/moviUsbBoot/moviUsbBoot.tar

5.1. Directory structure

README	README for users (info available also in this document)
README.dev	README for developers (info available also in this document)
Makefile	Linux/POSIX makefile
windriver/	Windows driver directory
Includes/	Include directory
linux/	Linux specific includes
win/	Windows specific includes
Sources/	Source directory
linux/	Linux specific source code
win/	Windows specific source code
libusb/	LibUSB files (including header)
x64/	x64 LibUSB lib and DLL
Win32/	x32 LibUSB lib and DLL
Binaries/	Build output directory
Debug/	Debug build output
Release/	Release build output
moviUsbBoot.vcxproj	Visual Studio 2015 Project
moviUsbBoot.sln	Visual Studio 2015 Solution

5.2. Build versions

moviUsbBoot has been written to build with either LibUSB or WinUSB.

Under **Linux**, LibUSB is always used.

Under **Window**, moviUsbBoot can be built to use LibUSB, but this is only for testing and historical purposes. Due to a bug in older versions of Renesas host drivers, LibUSB is unable to find the device on some systems. Additionally, as recent versions of LibUSB use WinUSB, using WinUSB directly avoids an extra dependency.

More information on the Renesas bug can be found here: https://github.com/libusb/libusb/wiki/Windows

The preferred version of moviUsbBoot for Windows should be the WinUSB version.

5.3. Building

Under **Windows**, the provided Visual Studio project should be used. Targets are provided for all combinations of: WinUSB or LibUSB x86 or x64 Debug or Release

Under **Linux**, the provided Makefile should be used.

The libusb-1.0-0-dev package must be installed to provide the libusb header and libraries.

moviUsbBoot can be cross-compiled by specifying a compiler prefix with the CROSS_COMPILE variable. However, no provision is made for building both x86 and x64 variants using the -m32 and -m64 command line options.

5.4. Source details

Sources/usb_boot.c:
  Main source code. Details below.
Sources/usb_common.h:
  Shared header for USB layer.
Sources/usb_libusb.c:
  LibUSB implementation for USB layer.
Sources/usb_winusb.c:
  WinUSB implementation for USB layer.
Sources/crc32.c, Includes/crc32.h:
  CRC32 calculation
Sources/dumphex.c, Includes/dumphex.h:
  Routine to output a hex dump, in a format similar to hexdump -C
Includes/linux/highrestime.h:
  High resolution timer routines for Linux/POSIX. Used for rate calculations.
Includes/win/highrestime.h:
  High resolution timer routines for Windows. Used for rate calculations.
Sources/win/getopt.c, Includes/win/getopt.h:
  Basic implementation of getopt(), for use under Windows.

5.5. Functions

main(): + Parses command line arguments, performs various initializations, and then loops through the command list.
parse_args():
  Processes the command line arguments using getopt().
usage():
  Outputs the usage string.
find_dev():
  Finds the USB device.
open_dev():
  Opens the USB device.
wait_findopen():
  Waits for the device to be found.
wait_disconnect():
  Waits for the device to be disconnected.
readback_command():
  Reads data from the device, displaying the output in one of 4 different ways. (see enum readback_mode)
send_file():
  Loads and writes a file to the device.

5.6. Windows driver details

Under **Windows**, the WinUSB driver is used, which provides user-level access to USB devices. Additionally, the ROM provides various USB descriptors that supply information to Windows in order to ease the installation process. Under Windows 8 and later, these descriptors allow Windows to automatically install the device, without requiring any user intervention. Prior to Windows 8, the user will have to manually install the driver, either using the provided utility, or by using the 'Update Drivers' option in Device Manager.

More details:

https://msdn.microsoft.com/en-us/library/windows/hardware/ff540283(v=vs.85).aspx

5.7. Files

Movidius_MA2X5X.cat

An empty catalogue file. Ideally Movidius would have the driver package signed by Microsoft. This file would then contain binary details of the driver package and be digitally signed. More details can be found at:

http://www.davidegrayson.com/signing/
https://msdn.microsoft.com/en-us/library/windows/hardware/dn653569(v=vs.85).aspx

delete_usbflags.reg

A registry file to delete keys that are automatically created when the device is first attached. For testing purposes only and should not be part of the release.

Movidius_MA2X5X.inf

The .inf file for driver installation, using the default VID and PID.

Movidius_MA2X5X-altvidpid.inf-test

A modified .inf file for driver installation when using an alternative VID/PID pair. For testing purposes only and should not be part of the release. Rename the extension to .inf to use.

Movidius MA2X5X Install - 32 bit.exe
Movidius MA2X5X Install - 64 bit.exe

32 and 64-bit versions of DPInst.exe tools. These will automatically install the drivers for .inf files in the same directory. They have been renamed to be more user-friendly.

amd64/
x86/
ia64/

Platform specific driver installation files (CoInstallers). Note that the actual WinUSB driver is not provided as part of this package. It is bundled with Windows 7 and upwards, and prior to that, the CoInstaller should take care of downloading the required driver file.