FW Update SDK
User Manual

Version 0.3

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/#/en_US_01

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

This document contains information on products in the design phase of development.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
*Other names and brands may be claimed as the property of others.

Copyright © 2013, Intel Corporation. All rights reserved.

Manual Contents

1. Introduction 5
2. FW_update SDK structure and installation 6
3. FW Update SDK – usage model 7
   3.1. Usage model 7
   3.2. FW update – sample high level flow 7
   3.3. FW Update source code sample 7
4. FW_update SDK API 8
   4.1. Data types and values 8
   4.2. Functions 8
      4.2.1. fw_update_check_driver_connection (static library) 8
      4.2.2. fw_update_perform_update (static library) 8
      4.2.3. FwUpdateCheckDriverConnection (dynamic library) 9
      4.2.4. FwUpdatePerformUpdate (dynamic library) 9
      4.2.5. FwUpdateGetCurrentNvmVersion (dynamic library) 9

Glossary

Acronyms	Expression
TBT	Thunderbolt
NVM	Non-volatile memory
FR	Falcon Ridge
NHI	Native Host Interface

1. Introduction

Scope

This document describes the usage instructions for "FW_update SDK".

Summary

"FW_update SDK" is a SW kit for developing TBT NVM update applications of FR based host system. Purpose of this document is to introduce package structure, usage model, describe SW API set and present coding samples.

Target Audience

- Thunderbolt FR based host systems providers.

OSs supported

- Windows 7 32 bit, Windows 7 64 bit, Windows 8 32 bit, Window 8 64 bit.

2. FW_update SDK structure and installation

Four main SDK components are kernel mode driver, static library, dynamic library and header file.

- Thunderbolt ™ Controller driver is needed to be installed to communicate with Thunderbolt ™ Host Interface. Driver installation must be done using Thunderbolt ™ Software Installer.

- SDK SW static library is FwUpdateLibWin32.lib (for Windows 32bit) or FwUpdateLibWinx64.lib (for Windows 64 bit). SDK users should link their applications with the library.

- SDK SW dynamic library is FwUpdateSdkWin32.dll (for Windows 32bit) or FwUpdateSdkWinx64.dll (for Windows 64bit). SDK users should load this library in their application to call FW Update related functions.

- SDK users must include tbt_fw_update_lib.h file.

3. FW Update SDK – usage model

3.1. Usage model

1.) FW Update library needs Thunderbolt Software to be installed, because it uses Thunderbolt Driver API to communicate with HW.
2.) FW Update library needs at least one device connected to Thunderbolt port to perform FW Update process (plugging in a device will power up the Thunderbolt controller and load the required device driver).
3.) FW Update library is able to work with any security level set by user (there is no need to set "debug mode" via BIOS strapping).

3.2. FW update – sample high level flow

Disclaimer: This code is not intended to be production code. It's brought here to demonstrate SDK API usage.

```
status = fw_update_check_driver_connection ();

if (status != TBTDRV_STATUS_OK)
        return status;

status = fw_update_perform_update (buffer_size, buffer);

if (status != TBTDRV_STATUS_OK)
        return status;
```

3.3. FW Update source code sample

Please find working C# sample application located under "Samples" directory. The sample demonstrates proper way to perform FW Update.

4. FW_update SDK API

4.1. Data types and values

FW_UPD_LIB_STATUS	This type holds dynamic library APIs return status.
TBTDRV_STATUS	This type holds static library APIs return status.
TBTDRV_STATUS_OK	This value indicates success return status (for static and dynamic libraries).

4.2. Functions

4.2.1. fw_update_check_driver_connection (static library)

TBTDRV_STATUS fw_update_check_driver_connection();

This function checks, if Thunderbolt ™ driver exists in the operating system. This function should be called prior to any library usage.

Arguments: None

Return value: TBTDRV_STATUS_OK is indication for success; any other value is error indication.

4.2.2. fw_update_perform_update (static library)

*TBTDRV_STATUS fw_update_perform_update(unsigned int buffer_size, unsigned char *buffer);*

This function performs FW Update process for Thunderbolt ™ controller using FW image read to buffer.

Arguments: *buffer_size – size of buffer (in bytes)*
buffer –pointer to buffer containing image read from file

Return value: TBTDRV_STATUS_OK is indication for success; any other value is error indication.

4.2.3. FwUpdateCheckDriverConnection (dynamic library)

FW_UPD_LIB_STATUS FwUpdateCheckDriverConnection();

This function checks, if Thunderbolt ™ driver exists in the operating system. This function should be called prior to any library usage.

Arguments: None

Return value: TBTDRV_STATUS_OK is indication for success; any other value is error indication.

4.2.4. FwUpdatePerformUpdate (dynamic library)

*FW_UPD_LIB_STATUS FwUpdatePerformUpdate(unsigned int buffer_size, unsigned int *buffer);*

This function performs FW Update process for Thunderbolt ™ controller using FW image read to buffer.

Arguments: *buffer_size – size of buffer (in bytes)*
buffer –pointer to buffer containing image read from file

Return value: TBTDRV_STATUS_OK is indication for success; any other value is error indication.

4.2.5. FwUpdateGetCurrentNvmVersion (dynamic library)

*FW_UPD_LIB_STATUS FwUpdateGetCurrentNvmVersion (unsigned int *nvm_version);*

This function returns the NVM version on the FW currently written on the host Thunderbolt™ controller.

Arguments: *nvm_version – a pointer to the NVM version (returned by the function)*

Return value: TBTDRV_STATUS_OK is indication for success; any other value is error indication.