Intel® Firmware Engine

Intel® Firmware Engine Application Release 4.0.0
Dec 18, 2017

DISCLAIMER

This release note as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document. Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

INDEX

FILE LIST

“Release_Note_R4.0.0.pdf”

This File. Release note for the Intel® Firmware Engine 4.0.0 application install package.

"IntelFirmwareEngineSetup_4_0.exe"

User Interface Installer Package including support for the MinnowBoard Turbot B21 Platform (Windows*).

"IntelFirmwareEngineSetup_4_0.bin"

User Interface Installer Package including support for the MinnowBoard Turbot B21 Platform (Linux*).

Quick Start Guides

-- Quick start guide for the User Interface Application

`"Intel Firmware Engine 4.0 Quick Start (EN).pdf”`	– English
`"Intel Firmware Engine 4.0 Quick Start (FR).pdf”`	– French
`"Intel Firmware Engine 4.0 Quick Start (DE).pdf”`	– German
`"Intel Firmware Engine 4.0 Quick Start (ZH_CN).pdf”`	– Simplified Chinese
`"Intel Firmware Engine 4.0 Quick Start (ZH_TW).pdf”`	– Traditional Chinese

INSTALL INSTRUCTIONS

Previous versions of Intel® Firmware Engine (prior to 4.0.0) must be uninstalled prior to installing this release.

Windows*

1. Under ‘Control Panel’, use the ‘Uninstall a program’ menu to uninstall any previous versions of Intel® Firmware Engine. Please backup any prior work prior to performing the uninstall operation.

2. Follow the Quick Start Guide for system requirements prior to installing the latest version. Execute the "IntelFirmwareEngineSetup_4_0.exe" file and follow the click through menus.

Linux*

1. To uninstall:
   a. Open a command Terminal in the directory to the directory where the Intel® Firmware Engine was installed (typically `/opt/Intel/Intel(R) Firmware Engine/4.0.0.xxxxxx`)
   b. Execute the command "Change Intel(R) Firmware Engine Installation" from a command Terminal, type:
      ```
      'sudo ./Change\ Intel\(R\)\ Firmware\ Engine\ Installation'
      ```
   c. Follow the uninstall instructions and keep any prior work desired.
2. To install:
   a. Follow the Quick start guide for system requirements.
   b. Open a command Terminal in the directory where the downloaded program file (`.bin`) is located.
   c. Change the .bin file properties to include 'execute' and grant required permissions using the following command: `'sudo chmod 777 ./IntelFirmwareEngineSetup_4_0.bin'`
   d. To Install, type '`sudo ./IntelFirmwareEngineSetup_4_0.bin`'
   e. Follow the installer click-through menus to complete installation.

NEW FEATURES AND CHANGES

1. Intel Firmware Engine application now supported in Linux (Ubuntu Linux 16.04.2 LTS X64).
2. Firmware core updated, based on the TianoCore UDK2017 release.
3. Microcode patch update for newer platforms and support for .mcb files.
4. More PCDs can be configured through the properties of Hardware modules.
5. GUI performance improvements.
6. Repository Maintenance Tool (RMT) improvements for Multiple Package support.
7. Secure Capsule/recovery solution alignment.
8. Improvements to GPIO table configuration:
   a. Transfer by Connector component.
   b. Connect one SoC pin to Multiple GPIO devices. (Multiple devices with same attribute can be connected to the same GPIO PIN)
   c. Display help information.
9. Provide the ability to turn off help on PCDs and Questions.
10. Trigger 'save project' prompt if changes were made.

PLATFORM DEPENDENT CONFIGURATION

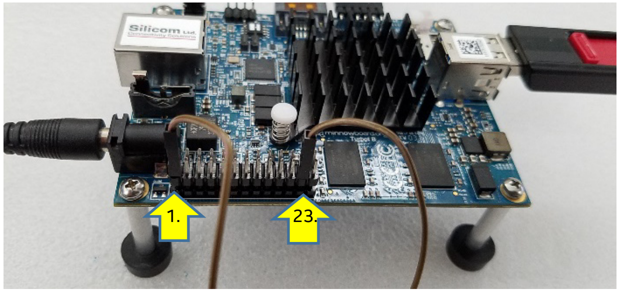
How to enable a "User Setup" ("BIOS Setup") menu (example using MinnowBoard Turbot B21):

1. Open a new project based on the "MinnowBoardTurbotB21" Source project.
2. Open 'Components->Firmware applications'.
3. *Drag and Drop* the "UIApp module" Into the box labeled "Added applications list".
4. Click the "Build" button to generate a new firmware binary image .FD file. (The `.FD` file will be located in the directory path as shown in the "Log Panel". Typically the "`../`*`ProjectName`*`/OUTPUT`" directory)
5. Flash the new image on the MinnowBoard Turbot B21 (See Help "Flashing a Firmware Device" for how to flash the image to your system under test device)
6. To go to "Setup" menu on MinnowBoard Turbot B21:
   a. Boot up to the UiApp.
   b. Select "Device Manager".

How to enable UEFI Secure Boot (example using MinnowBoard Turbot B21):

1. Under Project -> Properties -> Settings, enable the "Enable Variable Authentication Support" and "Enable Image Authentication Support" options.
2. Open 'Components->Firmware applications'.

3. *Drag and Drop* the "UIApp Module" Into the box for "Added applications list".
4. Click the Build button to generate the .FD file.
5. Flash FD image to the MinnowBoard Turbot B21.
6. To Enable Secure Boot on MinnowBoard Turbot B21 using the UiApp:
   a. In order to enable the "Custom" Secure Boot menu, the low speed expansion connector Pins 1 and 23 need to be shorted. Use a jumper wire to connect pins 1-23 together as shown in the figure below

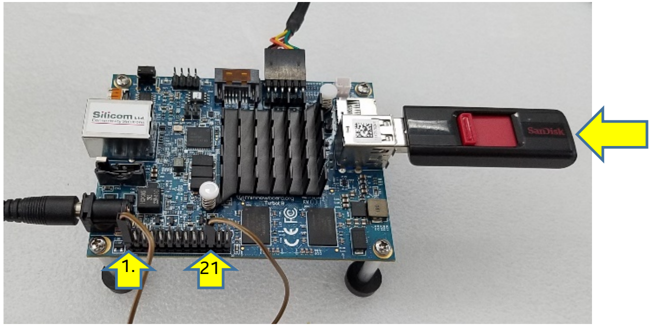


See Pin layout: https://minnowboard.org/tutorials/connecting-device-sensor-lse

   b. Boot up to the UiApp.
   c. Enter the UiApp -> Device Manager -> Secure Boot configuration
   d. Change "Secure Boot Mode " to "Custom Mode"
   e. Enter "Custom Secure Boot Options" menu item and there will be options to modify PK, KEK, DB, DBX and DBT.
   f. To enable the UEFI secure boot certificates for PK, KEK, DB, DBX and DBT, download the following document : https://github.com/tianocore-docs/Docs/raw/master/User_Docs/SigningUefiImages%20-v1dot31.pdf. Follow the instructions in section 1.9.4 of the downloaded user document for how to enable UEFI Secure Boot.

How to enable Secure Recovery (example using MinnowBoard Turbot B21):

1. Under Project -> Properties -> Settings, enable the "Enable Recovery File Generation". Choose setting "RSA 2048 SHA 256 based authentication using a test signing key" in "Firmware Recovery and Firmware Update Authentication Support" option.
2. Connect 'USB Flash Drive' component to 'Intel Atom(R) processor E3800 series', right click 'USB Flash Drive' properties, under 'Enable OS Boot and/or Recovery Support', choose 'OS Boot and Firmware Recovery' item. Apply the configuration and exit. (Note this must already be selected on current flash image of the MinnowBoard Turbot B21)
3. Click the Build button to generate the FD file and RECOVERY.Cap File. (The RECOVERY.Cap will be located in the same directory path as shown in the "Log Panel" of the .FD file created Typically the "../*ProjectName*/OUTPUT" directory)
4. In order to enable the 'Recovery Mode', the low speed expansion connector Pins 1 and 21 need to be shorted. Use a jumper wire to connect pins 1-21 together as shown in the figure below



5. Copy RECOVERY.Cap file to the root directory of a USB flash disk for Recovery
6. Insert the USB flash disk to the MinnowBoard Turbot B21.
7. Apply power (Note, Do not remove power until the process has finished)

KNOWN LIMITATIONS

Installation:

Virus scanner issues

1. MacAfee 8.008* has a recent security update that requires the addition of the following directory and sub-directories to the Exception list: C: Users\Public\Repository. This will reduce the install length.
2. Avast Virus* needs to have the following directory added to the exceptions list will not open a project unless the repository directory is added to the exception list: C: Users\Public\Repository
3. Kaspersky* virus scanner needs the following file added to the exceptions: C: Users\Public\Repository.

Linux installation issues.

1. After installation is complete you must reboot in order to see the icon under Ubuntu*.
2. Ubuntu Software* "Remove" option from the Intel Firmware Engine icon menu will not uninstall the application. Uninstall with the command in Terminal "Change Intel(R) Firmware Engine Installation" using 'sudo'.

Application:

Issue opening the application

1. If you get the message "Can't connect to server" uninstall and reinstall the application. This can be caused by a corrupt repository.

Project naming

There are restriction on project names based on reserved characters in Linux and Windows operating systems.

1. The names must be less than 100 characters.
2. The following characters cannot be used in project names (Windows and Linux) and repository name (Linux):
    a. Windows: back-slash, forward-slash, colon, asterisk, question mark, double quotes, left angle bracket and right angle bracket
    b. Linux: forward splash
    c. Linux has an additional restriction as non-ascii characters are not allowed in repository names

Issues with the project file (.bimx)

1. The read-only file MinnowBoardTurbotB21.bimx will not build if opened by doubling clicking on the file icon or by opening using the browse. The project (MinnowBoardTurbotB21) will build if opened through the "Recent projects" menu.
2. Setting a restore point will fail if the following:
    a. Windows* if the .bimx file resides in C:\Program Files\(x86)\Intel\Intel(R) Firmware Engine or C:\Users\Public\Repository\Intel_Firmware_Engine_4_0.
    b. Linux* if the .bimx file resides in opt/Intel/Intel(R) Firmware Engine/etc.

Search

1. The search function is limited to the scope of the current open window.

GPIO issues

1. When dragging one of the GPIO "Device Components" from one pin to another pin in the GPIO table with the purpose of connecting multiple device components, it is necessary to check whether the dragged device attributes are the same with GPIO pin attributes which are being connected.
2. Long delays may occur if the table has a majority of it connections occupied.

Adding your own source drivers

1. When adding a driver from “UEFI driver from source build” option, first make sure the UEFI driver builds properly under the EDI II build environment.
2. If adding a driver using “UEFI driver with metadata” fails, try adding a driver from source build.
3. When adding a driver from “UEFI driver from source build” option the driver will not appear in the Firmware Inventory report.
4. When adding a driver from “UEFI driver from source build’ option the build will only build once from the source files. Subsequence changes to the source will not be recognized. Removing the build directory will also cause the build to fail. The work around is to remove the source file from the “UEFI driver from source build” option before making a modification to the source code driver and/or removing the build directory. Then re-add using "UEFI driver from source build” option.
5. On Linux* when adding “UEFI driver from source build” upon clicking the 'Browse' button to select a file or folder, the application may exit abnormally.

Linux*

1. There is no Graphical user application for Repository Maintenance tool under Linux. Use the command line tool in Terminal at /opt/Intel/Intel (R) Firmware Engine /4.0.xxxx/bin/RMT

Platform specific:

1. When there is USB Flash drives with a UEFI Boot path (EFI/BOOT/…) connected to the target platform (example: MinnowBoard Turbot B21), there may be a long delay using the “UiApp” Firmware applications module and/or pressing "R" to update boot targets in the first page boot menu..
2. On the target MinnowBoard Turbot B21 platform, after a change of the Serial port Line Control Settings to (Stick or Mark) and then Build and flashing the FD image into board the image will not boot to the UEFI Shell.
3. If the serial port isn't initialized and not included as drivers for the platform, booting to Yocto 1.8/2.2* on the target system may be slow.

Interactions with Third Party applications

1. The project file (.bimx) cannot be opened via Windows Remote Desktop*.
2. For third party tools such as the DediProg* flash programmer, there is no support for file name paths with non-ASCII characters. Recommend directory path names must use ASCII formatting.
3. The build process is extremely slow when the Avast Virus* scanner is installed and enabled.

Application Help:

1. Opening “Help” in Microsoft Internet Explorer* is not recommended.
2. Microsoft Edge* and Google Chrome* do not support the Search function in in the Chinese Windows* operating system.
3. Microsoft Edge* may not play some tutorials on Windows 10*.
4. In Microsoft Edge* and Google Chrome* the tutorials forward and previous buttons do not work.
5. On Ubuntu*, the platform project help menu for “Project” will not open when the project name is in Chinese.

Recommendations:

1. It is recommended that an imported BIM file (.bmx) be modified only by the Intel Firmware Engine application tool. Modifications from other applications and/or tools could cause the .bimx file to become corrupt or invalid.
2. It is NOT recommended to flash a built .FD image into a system under test if there is a warning from the build: “WARNING: The selected components exceed the available flash space.” Doing so could result in a non-bootable system under test. This symptom occurs by Enabling the Symbolic Debug Support and Enabling Recovery File Generation in the “Project properties” menu. The work around is to remove components until the build does **not** have the exceeded flash space warning message.

3. It is recommended to save your project before making several successive changes in a row (i.e. adding, removing devices and making configuration changes in succession.

. * Other names and brands may be claimed as the property of others.

[END OF RELEASE NOTES]