A More Secure and Better User Experience for OS-Based Firmware Update

Spring 2017 UEFI Seminar and Plugfest
March 27 - 31, 2017
Presented by David Liu (Phoenix Technologies)
Agenda

• What is OS-based Firmware Update
  – BIOS roles
  – What to prepare from IHV?
    • Capsule format
    • Flash Update Driver
    • ESRT update
• User Experience
  – Security
  – Visual
  – Stability
• Reference
• Questions?

UEFI Plugfest – March 2017
www.uefi.org
What is OS-based Firmware Update?
OS-based Firmware Update

• OS like Windows 10 or Ubuntu provides a common, simple and automatic method for end-users to update firmware on the platform
• Firmware updates are not only limited to the system firmware (UEFI BIOS), but also other SOC device firmware
• Firmware updates also includes plug-in devices such as OPROM
What is the Role of UEFI BIOS?

• BIOS provides EFI System Resource Table (ESRT) for OS to identify device and system firmware resources
• BIOS provides capsule update mechanism to process the capsule from the OS
• BIOS provides signing and verification process for capsule update
ColorHugALS Firmware
Firmware for the ColorHug Ambient Light Sensor

Install

Updating the firmware on your ColorHugALS device improves performance and adds new features.

This stable release fixes the following bugs:
• Fix the return code from GetHardwareVersion
• Scale the output of TakeReadingRaw by the datasheet values

Website

Details

<table>
<thead>
<tr>
<th>Version</th>
<th>3.0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updated</td>
<td>Never</td>
</tr>
<tr>
<td>Category</td>
<td>None</td>
</tr>
<tr>
<td>Source</td>
<td>Hughski Limited</td>
</tr>
</tbody>
</table>

License  GPL-2.0+
Size  9.7 kB
UEFI BIOS Firmware Update

• Most OEM/ODM/IBV already support the UEFI BIOS firmware update from the OS such as Windows UEFI Firmware Update (WUFU) or Linux fwupd.

• But few of the device firmware updates exist on the market.

  – We want to encourage all IHV to work with OEM/ODM/IBV to have their own device firmware update through OS.
Independent Hardware Vendor (IHV) provides:

– New device firmware image
– UEFI Device flash update driver

• Utilize the UEFI standard protocols (Recommended)
• Work with OEM/ODM/IBV to find a solution to communicate the device on the platform unless there is no standard protocol (Optional)
Capsule Format
• Capsule header contain the EFI_GUID, which represent the unique GUID of the ESRT device
Device Flash Update Driver
What are the Problems?

• Some UEFI BIOS services, which are required by the device flash update driver, may not necessarily exist on other OEM platforms
• Device flash update driver often needs special modification for different OEM platforms
• Device flash update driver may require modification based on different OEM hardware configuration
Firmware Management Protocol

• OPRROM provide the instance of the FMP
• New firmware image could be passed through capsule update
• UEFI BIOS check the capsule GUID and consume the proper FMP to update the OPRROM
• UEFI BIOS update the ESRT to report current flash status and firmware version
Non FMP Flash Update Driver

• IHV provides the standalone self-execute UEFI driver to install flash in its own device firmware

• Pack the self-execute driver into the capsule. UEFI BIOS will execute its driver entry to start the flash update process while processing the capsule
What Needs to Improve?

UEFI Plugfest – March 2017

www.uefi.org
UEFI Standard Protocols

• Depending on the device it’s connected to on the platform, IHV should choose the suitable protocols for the UEFI device flash update driver.

• PCI Bus protocols
  – Chapter 16, Protocols — PCI Bus Support, UEFI Spec 2.6

• SMBus protocols
  – Chapter 3, SMBus Host Controller Code Definitions, UEFI VOLUME 5: Platform Initialization Specification Standards, V1.5

• USB Protocols
  – Chapter 16, Protocols — USB Support, UEFI Spec 2.6

• I2C protocols
  – Chapter 17, I2C Protocol Stack, UEFI VOLUME 5: Platform Initialization Specification Standards, V1.5
Vendor Configuration Data

IHV

Device Flash Drivers

Update Device Firmware V1.1

Vendor Configuration Data (Optional)

Modify

OEM A

Capsule Device_GUID

Vendor Configuration Data for OEM_A (Optional)

Device Flash Drivers

Update Device Firmware V1.1

OEM B

Capsule Device_GUID

Vendor Configuration Data for OEM_B (Optional)

Device Flash Drivers

Update Device Firmware V1.1
EFI System Resource Table (ESRT) Update
Update ESRT for Device Firmware

• Device firmware should have “Unique GUID” to represent its own device on ESRT
• IHV could provide a UEFI driver to bundle within the BIOS ROM to report the firmware information during each boot (Recommend)
• Device Flash Driver could use BIOS Non-Violate (NV) variable to store the device firmware information and flash status (Alternative)
ESRT Update from EDKII
ESRT_MANAGEMENT_PROTOCOL

UEFI BIOS

ESRT

Device A
Ver 1.0

Device A Update Driver

Device B
Ver 2.0

Device B Update Driver

Device C
Ver 3.0

Device C Update Driver

OS Initial Update

UEFI BIOS

UpdateCapsule

Flash Update Process

Variable

L"EsrtNonFmp"

L"EsrtFmp"

Create

UEFI BIOS

ESRT

Device A
Ver 1.1

Device B
Ver 2.1

Device C
Ver 3.1

OS Initial Update

UEFI Plugfest – March 2017

www.uefi.org
ESRT DXE Update Driver

• Device creates its own ESRT update DXE Driver to update ESRT configuration table directly (Recommended)

• Why it is recommended?
  • Avoid accidental variable deletion
  • If the variable is deleted, then it is lost forever
  • Avoid using other caller to modify the variable data during runtime
ESRT Update Example from BIOS Driver through ESRT Protocol

- **Device A**: DXE Driver
  - **Ver 1.2**

- **Device B**: DXE Driver
  - **Ver 2.5**

- **Device C**: DXE Driver
  - **Ver 3.1**

**UEFI BIOS**

- **Flash Update**
  - UEFI BIOS UpdateCapsule
    - Device A Update Driver
    - Device B Update Driver
    - Device C Update Driver

**ESRT MANAGEMENT PROTOCOL**

- **BIOS Post**
  - Device A Driver
  - Device B Driver
  - Device C Driver

**OS Initial Update**

- Computer Reset
What Should You Prepare for OS-based Firmware Update?

• **Unique ESRT GUID**
  – Represents each individual hardware device or UEFI BIOS

• **Methods to update the ESRT entry**

• **UEFI Flash Update Driver**
  – Chose the standard UEFI communication interface
  – Separate vendor configurable data from update driver (ex: I2C Slave Address per hardware design)

• **New firmware image file**
User Experience
User Experience

Security
••Protects the capsule image file

Visual
••Helps to display UI about the flash update progress

Stability
••Reduces the possibility of damaging the device firmware during flash update
Security

• Why?
  – Avoid flashing the corrupted firmware
  – Avoid flashing the hostile or modified firmware
  – Avoid execute hostile or modified flash update driver

• How?
  – Sign the device firmware data (IHV)
  – Sign the UEFI device firmware update driver with UEFI db (IHV)
  – Sign the capsule with RSA2048 with SHA256 with OEM/ODM private key (OEM/ODM)
  – Sign the OS-based image content by OS’s signing algorithm (OS)
Security Signing Process

IHV

Updated Device Firmware
UEFI Drivers

OEM/ODM
Sign the capsule using RSA2048 private key

Windows Firmware Update Package
INF file
Catalog file
Capsule

Color marking:
- Microsoft-signed
- OEM-signed

Capsule
Updated System Firmware
Updated Device Firmware
UEFI Drivers

Sign

OEM-signed or IHV-signed (device firmware)
Signed with key in UEFI “db” (not applicable to ARM)
Better Security Signing Process

Color marking:
- **Microsoft-signed**
- **OEM-signed**
- **OEM-signed or IHV-signed (device firmware)**
- **Signed with key in UEFI “db” (not applicable to ARM)**
**OS**

- **OS verify the Windows Firmware Update Package**
- **Action ->** Call UEFI UpdateCapsule () and hand over the capsule to UEFI BIOS

**UEFI BIOS**

- **UEFI BIOS verify Capsule**
- **Action ->** Launch and process the content from the trusted Capsule

**UEFI BIOS**

- **UEFI BIOS verify device flash update driver**
- **Action ->** Execute the trusted device flash update driver

**Flash Update Driver**

- **Device flash update driver verify firmware image from capsule**
- **Action ->** Flash is the trusted firmware to the device
Visual Notification

• Windows UEFI Firmware Update (WUFU) request that the flash screen must have the Windows boot look and feel

• UEFI BIOS should display meaningful information on the screen
  – Indicates the update is still in process with animated appealing that the system has not locked up or crashed (Recommended)
  – Current update progress (Better)
Current Flash Update Screen

Windows OS → Prepare → UEFI BIOS

UEFI BIOS → Hand Over → Display

Device Flash Drivers

Prepare

Display

Execute

Device Flash Drivers

Capsule Device_GUID

Update Device Firmware

Device Flash Drivers

UX Capsule

{3b8c8162-188c-46a4-aec9-be43f1d65697}

Bitmap

Please wait while we install a system update
Current Flash Update Screen

• What have we noticed?
  – UEFI BIOS only accepts certain UX capsule GUID to display the bitmap on screen
  – UEFI BIOS only performs certain display methods on the flash screen
  – UEFI BIOS does not need those flash update display mechanisms in BIOS ROM
  – UEFI BIOS needs to re-flash the BIOS itself if OEM/DOM want a different look and feel for the flash update screen
How Can We Improve?

• Pack the flash screen display UEFI driver into capsule with necessary data like bitmap
• The flash screen display UEFI driver could be customized for each device firmware capsule
• OEM/ODM could decide if they want to include several UEFI drivers in capsule for different tasks
Flash Display UI Driver

OS

Prepare

UEFI BIOS

Hand Over

Execute

Device Flash Drivers

Prepare

Update Device Firmware

Flash Screen Display Driver

Device Flash Drivers

Capsule Device_GUID

Bitmap

Execute

Display

UEFI Plugfest – March 2017

www.uefi.org
Example

Please wait while we update touch
Stability

• Firmware update should perform a series of pre-installation checks before the update
• These checks avoid firmware update corruption by accident
• These checks may be platform specific
• Example:
  – Power Check (less then 25%)
  – Version Check
  – Power Button Lock (Lockup the power button)
  – Storage Check
Propose Capsule Format

Capsule Device_GUID

- Flash Screen Display Driver
- Power Check
- Storage Check
- Power Button Lock
- Device Flash Drivers

OEM Provide
- Update Device Firmware
- Device Vendor Configurable Data
- Bitmap

IHV Provide
- OEM Provide
- IHV Provide

Perform power check if it is less than 25% or A/C power is required

Perform storage check like whether or not the BIOS SPI ROM or HDD have enough space to backup current device firmware

Lock the power button to prevent user switch off the platform
Why?

• OEM/ODM would be able to add or change multiple pre-installation check drivers to the capsule before flash update begins, and the driver implementation could be changed at anytime and repack as the new capsule without needing to re-flash the UEFI BIOS.
Summary

• IHVs are encouraged to use UEFI standard protocols and unify ESRT update method for its own device firmware, and separate vendor specific data for different OEM/ODM for better management

• OEM/ODM needs to provide the necessary UEFI BIOS service for IHVs

• Device capsule with pre-installation check/action to enhance user experience
Reference

• Windows UEFI Firmware Update

• Linux Vendor Firmware Service
  – http://fwupd.org.s3-website-eu-west-1.amazonaws.com/
Questions?
Thanks for attending the Spring 2017 UEFI Seminar and Plugfest

For more information on the UEFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by

UEFI Plugfest – March 2017  www.uefi.org