Trusted Firmware-M

Firmware Update Service

Sherry Zhang
Arm
Agenda

- PSA Firmware Update APIs
- Firmware Update (FWU) service in TF-M
- FWU Service Integration with FreeRTOS OTA
- Future Plan
PSA Firmware Update APIs

- Firmware update support is an essential property of a PSA device.

- Arm Published the [PSA Firmware Update API Beta 0 version](https://www.arm.com/en/resources/psa-firmware-update-api) in February 2021.

- It defines a standard firmware interface for firmware updates.
Trusted Firmware M(TF-M)

- **Trusted Firmware-M** (TF-M) implements a Secure Processing Environment (SPE) for Armv8-M architecture (e.g. the Cortex-M55, Cortex-M33 and Cortex-M23 processors) and dual-core Cortex-M devices.
- It is a PSA reference implementation aligning with **PSA Certified guidelines**, enabling chips, Real Time Operating Systems, and devices to become PSA Certified.
- An Open Source project hosted in Trusted Firmware Open Governance community project.
Firmware Update (FWU) partition in TF-M
FWU Service APIs

- **Query image information**
  - `psa_fwu_query` ---- Returns information for an image of a particular image ID.

- **Store image**
  - `psa_fwu_write` ---- Writes an image to its staging area.

- **Verify image**
  - `psa_fwu_install` ---- Starts the installation of an image.
  - `psa_fwu_accept` ---- Indicates that the upgrade is successful.

- **Trigger reboot**
  - `psa_fwu_request_reboot` ---- Requests the platform to reboot.

Refer to PSA Firmware Update API for details of all the APIs:
[https://developer.arm.com/documentation/ihi0093/0000/](https://developer.arm.com/documentation/ihi0093/0000/)
Image State Transition in FWU Partition

- **UNDEFINED**
  - \texttt{psa\_fwu\_write()}

- **CANDIDATE**
  - \texttt{psa\_fwu\_write()}
  - \texttt{psa\_fwu\_install()}
    - Returns PSA\_SUCCESS\_REBOOT
  - \texttt{psa\_fwu\_install()}
    - Returns PSA\_ERROR\_DEPENDENCY\_NEEDED

- **RESTART NEEDED**
  - \texttt{psa\_fwu\_request\_reboot()}
    - and installation fails

- **INSTALLED**
  - \texttt{psa\_fwu\_accept()}
  - \texttt{psa\_fwu\_install()}
    - Returns PSA\_SUCCESS

- **PENDING INSTALL**
  - \texttt{psa\_fwu\_request\_reboot()}
    - and installation succeeds
Bootloader Shim Layer APIs

- Initialization
  - `fwu_bootloader_init` ---- Initialization at FWU partition initialization, such as read necessary data
- Staging area init
  - `fwu_bootloader_staging_area_init` ---- Prepare the staging area of the image with the given ID for image download.
- Load image to staging area
  - `fwu_bootloader_load_image` ---- Load a block data to its staging area
- Install image in staging area
  - `fwu_bootloader_install_image` ---- Starts the installation of an image.
- Accept image
  - `fwu_bootloader_mark_image_accepted` ---- Mark running image as installed.
TF-M integration with FreeRTOS

- TF-M integration with FreeRTOS kernel based on Armv8-M has been supported in its official repo. https://github.com/FreeRTOS/FreeRTOS-Kernel/tree/main/portable/ThirdParty/GCC/ARM_CM33_TFM
- TF-M integration with PKCS11 library is a standalone repo https://github.com/Linaro/freertos-pkcs11-psa and it is cloned into FreeRTOS by submodule
- TF-M integration with FreeRTOS OTA is in progress https://github.com/Linaro/amazon-freertos/pull/5
FWU Service Integration with FreeRTOS OTA

AWS OTA Server

Non-Secure Processing Environment

OTA Application

OTA Agent

OTA PAL

MQTT

HTTPS

TLS

Secure Processing Environment

Firmware Update Partition

Crypto Partition

Internal Trusted Storage Partition

TF-M

OTA Library

Application

Library

Component

Secure Partition
## Mapping of the TF-M FWU Service APIs and FreeRTOS OTA PAL APIs

<table>
<thead>
<tr>
<th>OTA PAL API</th>
<th>PSA Functional APIs</th>
</tr>
</thead>
<tbody>
<tr>
<td>prvPAL_Abort</td>
<td>psa_fwu_abort</td>
</tr>
<tr>
<td>prvPAL_CreateFileForRx</td>
<td>None</td>
</tr>
<tr>
<td>prvPAL_CloseFile</td>
<td>psa_fwu_query</td>
</tr>
<tr>
<td></td>
<td>psa_asymmetric_verify</td>
</tr>
<tr>
<td>prvPAL_WriteBlock</td>
<td>psa_fwu_write</td>
</tr>
<tr>
<td>prvPAL_ActivateNewImage</td>
<td>psa_fwu_install</td>
</tr>
<tr>
<td></td>
<td>psa_fwu_request_reboot</td>
</tr>
<tr>
<td>prvPAL_ResetDevice</td>
<td>psa_fwu_request_reboot</td>
</tr>
<tr>
<td>prvPAL_SetPlatformImageState</td>
<td>psa_fwu_accept</td>
</tr>
<tr>
<td>prvPAL_GetPlatformImageState</td>
<td>None</td>
</tr>
</tbody>
</table>
TF-M protected FreeRTOS OTA Process

- Image Download Process
- Signature Check Process
- Activate Image Process

AWS OTA Server

OTA Library

MQTT Library

TLS Library

Firmware Update Service

Crypto Service

Internal Trusted Storage Service

- psa_fwu_request_reboot
- psa_hash_XXX
- psa_asymmetric_verify
- psa_open_key
- psa_asymmetric_sign
TF-M protected FreeRTOS OTA Process

- Image Version Check Process
- Image Accept Process

OTA Server

TLS Connection
TLS Connected
Self Test

MQTT Connection
MQTT Connected

OTA Library

MQTT Library

TLS Library

Firmware Update Service

Crypto Service

Internal Trusted Storage Service

psa_fwu_queue

psa_open_key
psa_asymmetric_sign
Test with AWS cloud
What’s Protected in whole OTA Process?

- TLS connection process
  - The private key is protected by PSA Crypto Service and PSA Internal Trusted Storage Service.

- OTA process
  - protects the image in the passive or staging area from being tampered with by the NSPE
  - protects the active image from being manipulated by NSPE
Future Plan

- Add FWU secure partition to TF-M v1.3.0 release
- Complete upstreaming the integration of TF-M Firmware Update service with the OTA PAL of FreeRTOS
- Align the implementation with PSA FWU specification future update
Thank you

Accelerating deployment in the Arm Ecosystem