

keySTREAM™ Firmware Over-The-Air Update

TPDS Usecase Guide

Table of Contents

Firmware Over-The-Air Update with keySTREAM-ECC608-TMNGTLS	3
Description	3
Training Video	4
Prerequisites	5
Setup	6
keySTREAM Account Setup and Requirements.....	9
Signing in on keySTREAM portal	11
Creating a Fleet Profile.....	11
Obtaining an API key	22
Creating Signing Key Pairs.....	24
ECC608-keySTREAM Firmware Over-The-Air Update Usecase	27
Generating device manifest	28
Provisioning Usecase Resources	32
Preparing new Application Image for FoTA update.....	39
Preparing a FOTA campaign in the KeyStream Cloud	42
Observing the result of the FOTA by keySTREAM in TeraTerm.....	45
Conclusion	47
Microchip Information	48
The Microchip Website	48
Product Change Notification Service	48
Customer Support.....	48
Microchip Devices Code Protection Feature	48
Legal Notice	49
Trademarks	49
Quality Management System.....	50

Firmware Over-The-Air Update with keySTREAM-ECC608-TMNGTLS

Description

Firmware Over The Air (FoTA) is a critical technology that enables remote updates of embedded device firmware without the need for physical access. This capability is essential for maintaining device security, deploying new features, and fixing bugs in a scalable and efficient manner. keySTREAM Connect is a secure, cloud-based platform designed to streamline and manage the FoTA process across a wide range of IoT devices.

Implementing FoTA with keySTREAM Connect and ECC608-TMNGTLS not only reduces operational costs and minimizes downtime but also enhances the overall reliability and security of connected devices. This use case demonstrates how organizations can efficiently manage large-scale device fleets and deliver continuous improvements to their products in the field.

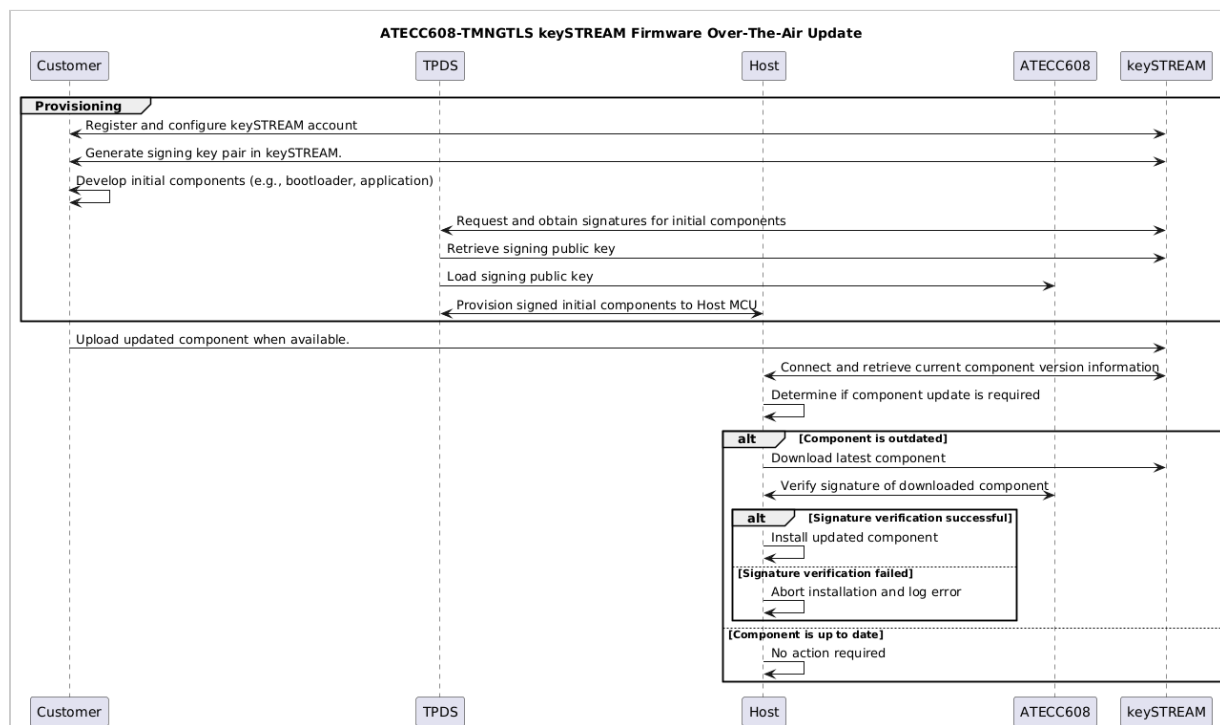


Figure-1

Training Video



Figure-2

Prerequisites

- Software
 - [Trust Platform Design Suite \(TPDS\)](#)
 - [MPLAB® X IDE](#)
 - Serial monitor program ("e.g. TeraTerm")
- Hardware
 - [CryptoAuth Pro Trust Platform Development Kit \(EV89U05A\)](#)
 - [WIFI WINCS02: EV68G27A](#)
 - Firmware version needs be 3.0.0 or higher
 - [Device Firmware Update \(DFU\) guide](#)
- Micro USB cable

Setup

- Connect WINCS02 on MicroBus2 (MB2) connector as shown below

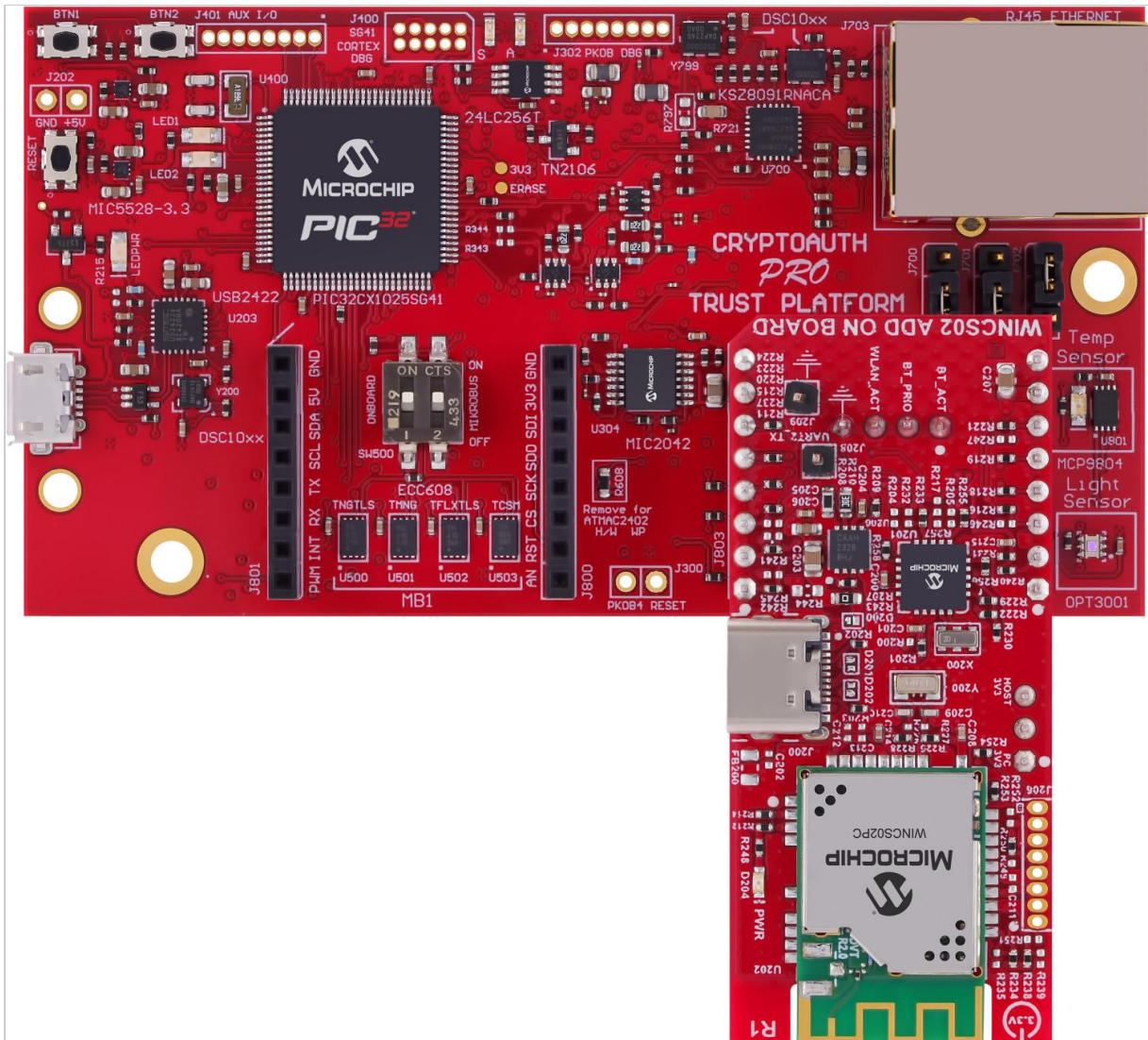


Figure-3

- Ensure both the ON and CTS switches are in the ON position in the Dual SPST DIP Switch (**SW500**) for selecting onboard ECC608 TRUSTMANAGER device. The I2C device address of ECC608 TRUSTMANAGER device is 0x70 - Connect the micro USB port on the board to the computer using a micro USB cable.
- Make sure the MPLAB X path is set in File -> Preferences -> MPLAB X path on TPDS.

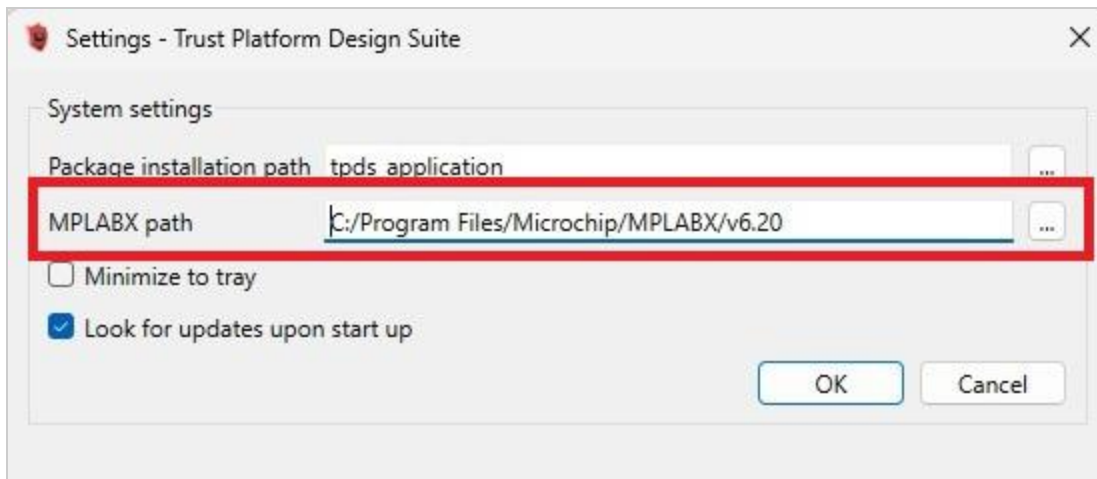


Figure-4

- Instructions for loading Factory program the CryptoAuth Pro Trust Platform kit (EV89U05A)
- Go to the TPDS "Utilities", click on "Device Interactions", select the EV89U05A board.

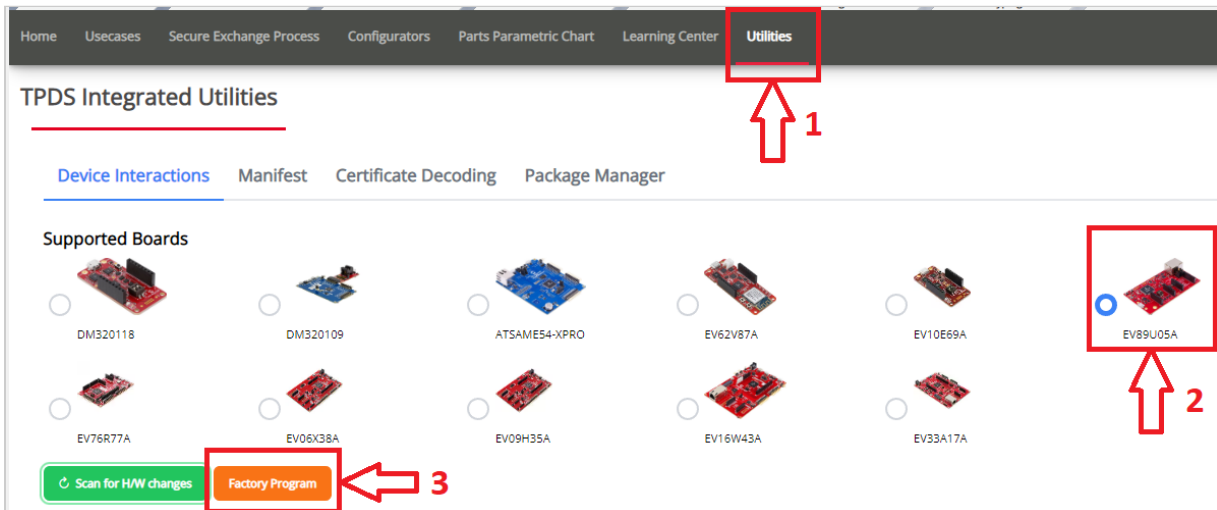


Figure-5

- click **Factory program** and wait until the process is completed, you will see a message when it is done

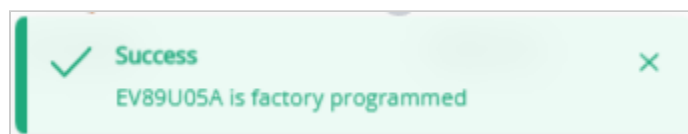


Figure-6

- After factory programming process is complete, launch the Terminal application (e.g., TeraTerm) on your computer.
- Connect to the Virtual COM port and configure the serial settings as follows:

- Baud : 115200
- Data : 8 Bits
- Parity : None
- Stop : 1 Bit
- Flow Control : None

- Press the Reset button on the CryptoAuth Pro Trust Platform Development Kit and observe a similar log:

```
-- CRYPTOAUTH PRO TRUST PLATFORM --  
-- Compiled: Jun  4 2025 17:17:21 v1.0.0 --  
-- Console log (115200-8-N-1) --  
  
KitParser Version: v3.2.1  
  
Device Discovery.....  
I2C ECC608C  C0  
I2C ECC608C  70  
I2C ECC608C  6C  
I2C ECC608C  6A  
Completed
```

Figure-7

keySTREAM Account Setup and Requirements

- Step 1: Creating a Kudelski keySTREAM account
- You must create an account on the Kudelski keySTREAM portal to proceed with the use case. Registration is free.
- Go to the following link for account creation: [Register link](#)
- Click on register.

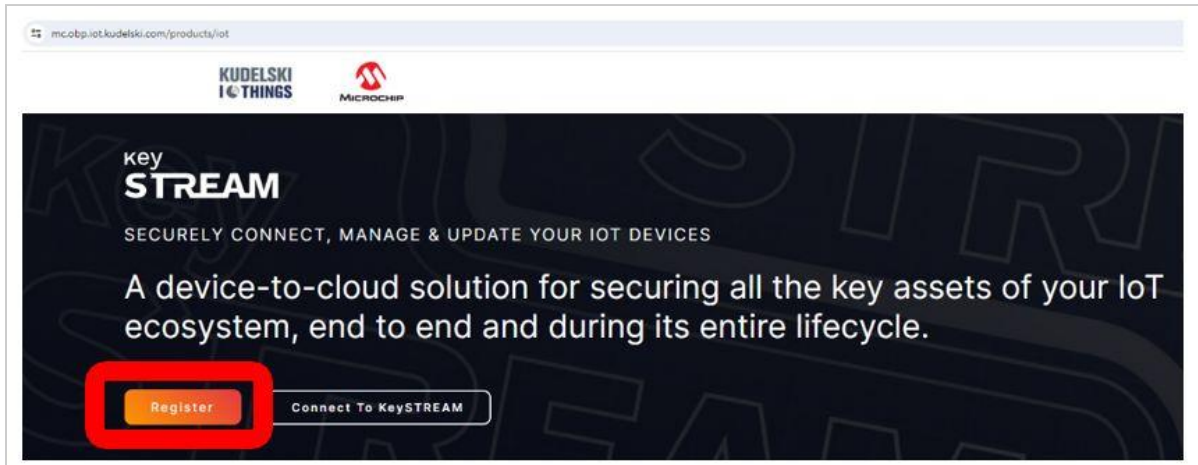


Figure-8

- Provide the information required for account creation. A verification email will be sent to the provided email address.

Figure-9

- Look for the verification code in the registered email. Enter the verification code once it is available to you. Once the correct code is entered, you should see a successful account creation message. - **WARNING:** No two accounts can have the same workspace similarly to email addresses. No two different user can create the same email addresses.

Figure-10

- **SUPPORT:** If you experience issues creating the account contact: iot.ops@nagra.com - Step 2: Creating a [fleet profile](#) under the keySTREAM portal.

- In this step, we will create a fleet profile under the Kudelski keySTREAM portal. A fleet profile could represent a product line or a subdivision under a product line. For example, you can set up a fleet profile for a line of internet-connected thermostats; this will let you remotely manage those specific devices. During the creation of the fleet profile, we will provide information required for creating a Root CA that will be used for this specific set of devices. All the devices under the fleet profile will be signed by this Root CA certificate. Each created fleet profile will always have a Root CA certificate associated with it.
- Log in to the Kudelski keySTREAM portal.

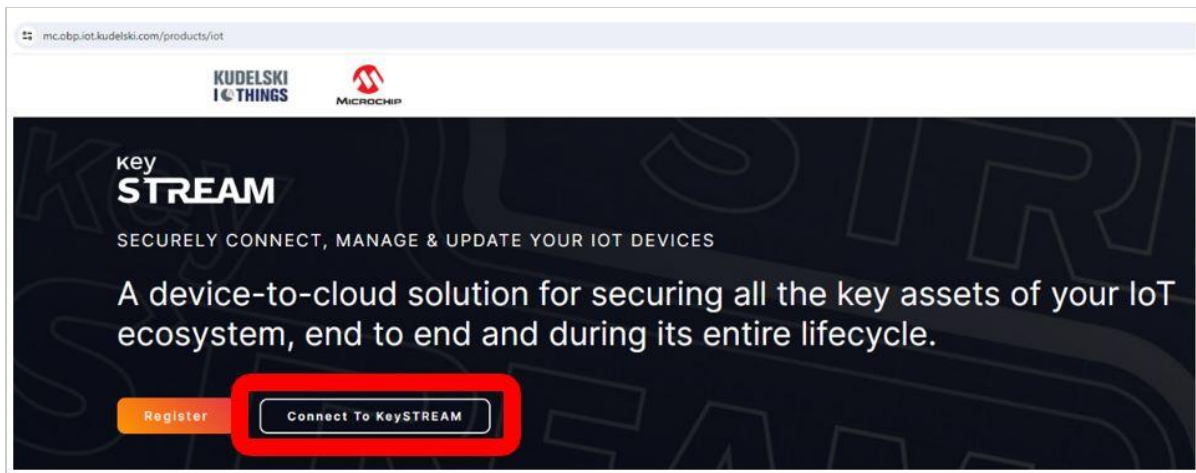


Figure-11

- Upon completing the registration process, you will gain access to your own tenant on the keySTREAM SaaS platform from Kudelski IoT.

Signing in on keySTREAM portal

- To sign in to the keySTREAM portal, please locate the link provided in the email from the **Kudelski IoT Onboarding Portal** confirming the creation of your keySTREAM tenant. This email includes comprehensive instructions, so ensure you review it carefully.
- Alternatively, go back to the keySTREAM portal via this link and click [Login to keySTREAM](#). Once logged in, you will be redirected to the home page.

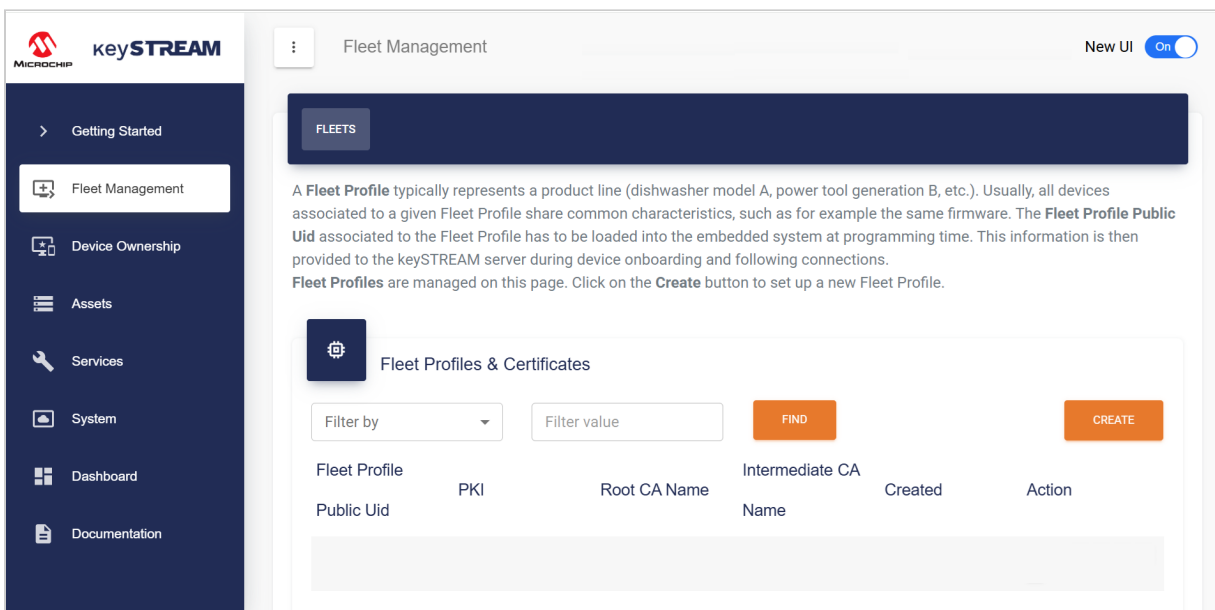


Figure-12

- For the upcoming steps, it's essential to be signed in to the keySTREAM portal.

Creating a Fleet Profile

- A Fleet profile:
- Is a label attached to a fleet of devices sharing the same configuration

- Serves as a link between you, the device manufacturer, and the devices you will have in the field.
- Will be referenced both on keySTREAM and in your device firmware.
- Is identified by a **Fleet Profile Public UID** that is easily readable by a human, as further explained below.
- the first 4-digits of the Fleet provide will be part of the ECC608 ordering part number.
- The keySTREAM portal allows you to create a Fleet Profile and attach a configuration to it:
- Click on Fleet Management from the left panel as shown in the figure below.

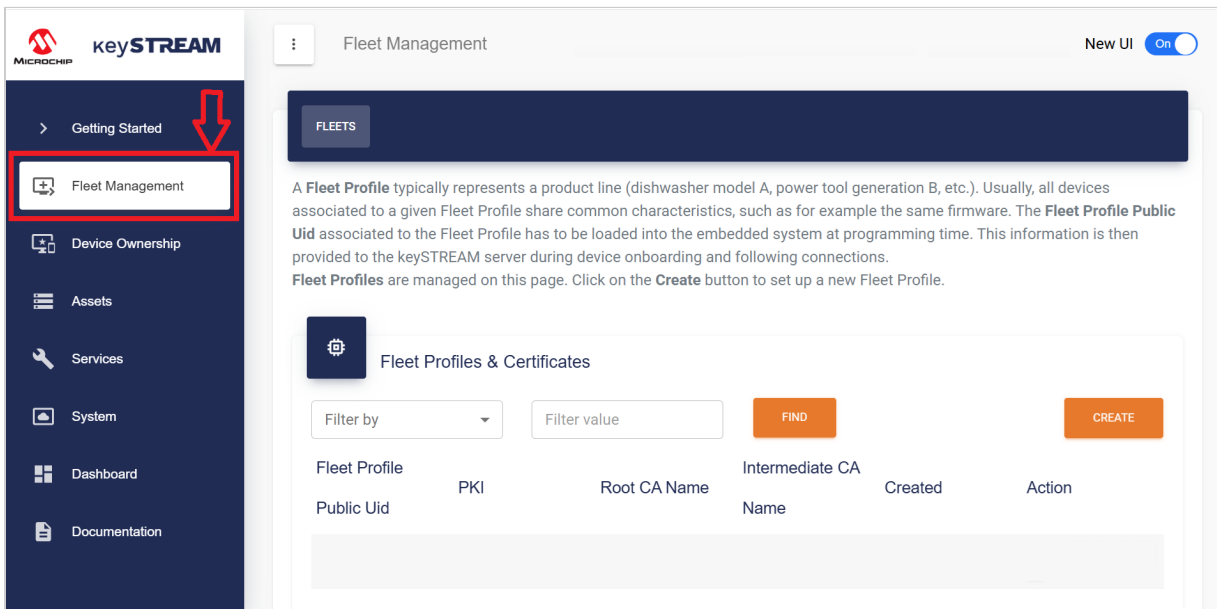


Figure-13

- Click on **FLEETS** and **CREATE** as shown in figure below.

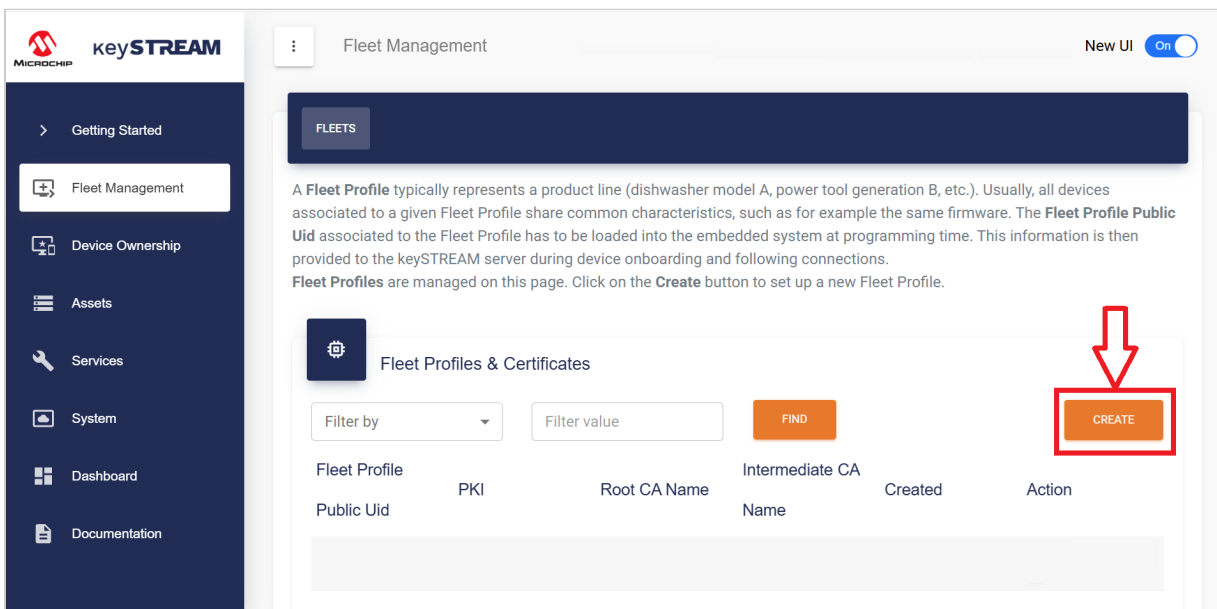
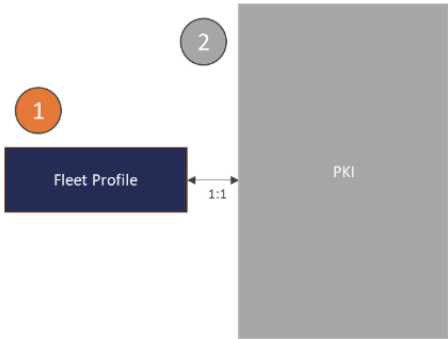


Figure-14

- A popup window appears:



+

Create Fleet Profile

Fleet Profile Public Uid *

?

Fleet Profile Public Uid BKFZ:

Model

Model

Brand

Brand

Manufacturer

Manufacturer

* is required

DISCARD

NEXT

Figure-15

- The following fields can be configured:
 - **Fleet Profile Public Uid (required):** A unique URI string identifying the Fleet Profile. It is advised to avoid creating two Fleet Profiles with the same name. The final Fleet Profile Public UID will be a concatenation of an auto-generated prefix (e.g., 9S4F:) and the Fleet Profile Public UID (example.org:dp:network:device) that you have defined. The final Fleet Profile Public UID would look like 9S4F:example.org:dp:network:device.
 - **Model, Brand and Manufacturer:** These are optional and purely informational, for your own use.
 - Upon clicking on **NEXT**, you can create your own 1-tier Root Certificate Authority (CA) associated with this new Fleet Profile:

Create your PKI

☒ 1-Tier PKI ☐ 2-Tier PKI

Chipset Models Compatibility *

Key Algorithm *

Root CA Common Name (CN) *

Root CA Organization (O) *

Root CA Certificate Validity (years) *

Device Operational Certificate Validity (years) *

Automatic Leaf Certificates Renewal ☐ Renew before (days)

* is required

Figure-16

- This is the step where you create your customized 1-tier Root Certificate Authority secured by Kudelski HSMs. The following fields will have to be configured:
 - **Root CA Common Name (CN):** This field is restricted to a fixed length of 16 bytes. If your input is shorter, it will be right-padded with spaces; a longer input will be truncated.
 - **Root CA Organization (O):** This field is restricted to a fixed length of 16 bytes. If your input is shorter, it will be right-padded with spaces; a longer input will be truncated. Note that the Organization Unit will be hardcoded to TrustMANAGER.
 - **Root CA Certificate Validity(Years):** Number of years of validity for the Root CA.
 - **Device Operational Certificate validity (Years):** The number of years of validity for the Device Operational Certificate. It must be shorter than the Root CA Certificate validity.
 - After you click on **COMMIT**, a newly configured Fleet Profile with its freshly created custom Root CA appears in the list below:

Profile **BKFZ:DEMO_PROFILE** created successfully

Filter by

Fleet Profile Public	PKI	Root CA Name	Intermediate CA	Created	Modified	Action
Uid			Name			
BKFZ:DEMO_PROFILE	1-Tier	DEMO	N/A	2024-07-18 14:50:52.000	2024-07-18 14:50:52.000	Get PoP

Figure-17

- All devices that are configured with this Fleet Profile will be provisioned with a unique device certificate signed by the certificate authority (CA) associated with this Fleet Profile when they register on keySTREAM.
- **Creating a 2-Tier Fleet Profile with a keySTREAM generated rootCA**

Some companies security policies requires the root CA to be separated from the device certificate. That's where the Intermediate Certificate Authority (ICA) brings its meaning. Here the rootCA, ICA, are generated and protected within Kudelski HSMS.

- If you need to create a 2-Tier Fleet Profile, follow the steps below:
- Root CA of your 2-Tier PKI: You can have only one 2-Tier Root CA, which will sign as many 2-Tier Intermediate CAs as needed.
- Click on **Fleet Management** from the left panel, then select **2-TIER ROOT CA**.

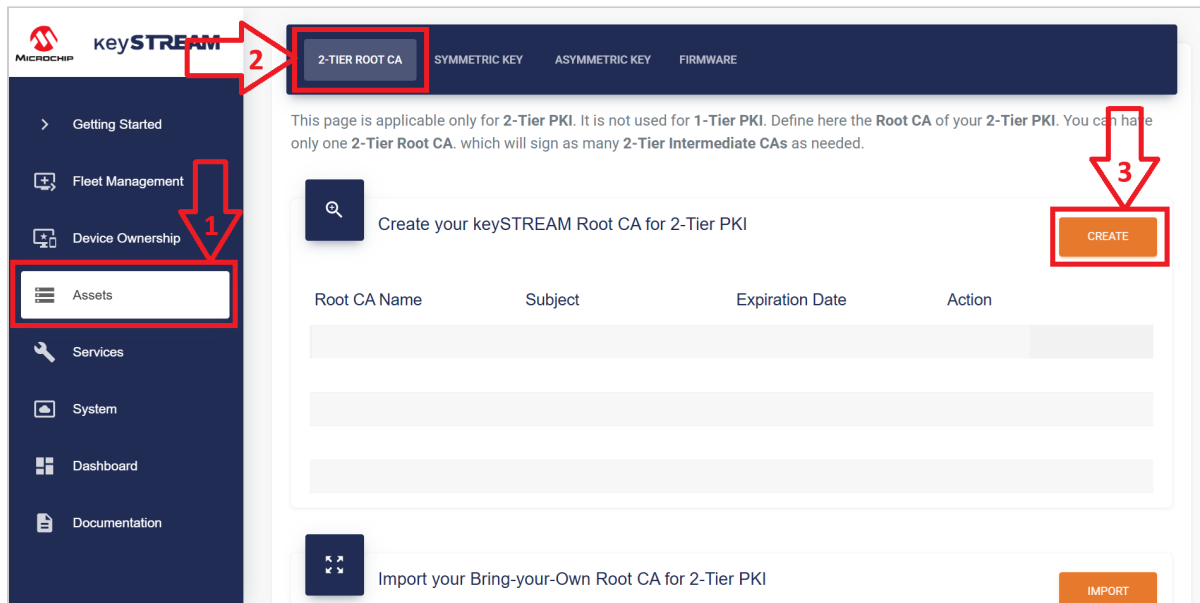


Figure-18

- A popup window appears : Select ECC608 as teh Chipset model, Feel free to enter Common name, Organization and Certificate validity to your liking. We recommend to set the Validity years to atleast 1 year.

Create your keySTREAM Root CA for 2-Tier PKI

Chipset Models Compatibility *

Chipset Models Compatibility
ECC608

Key Algorithm *

Key Algorithm
EC-PRIME256V1

Common Name (CN) *

?

Common Name (CN)

Organization (O) *

?

Organization (O)

Certificate Validity (years) *

?

Certificate Validity (years)

* is required

DISCARD

COMMIT

Figure-19

- After creating a ROOT CA, a pop-up will appear indicating that it has been successfully created, as shown below.

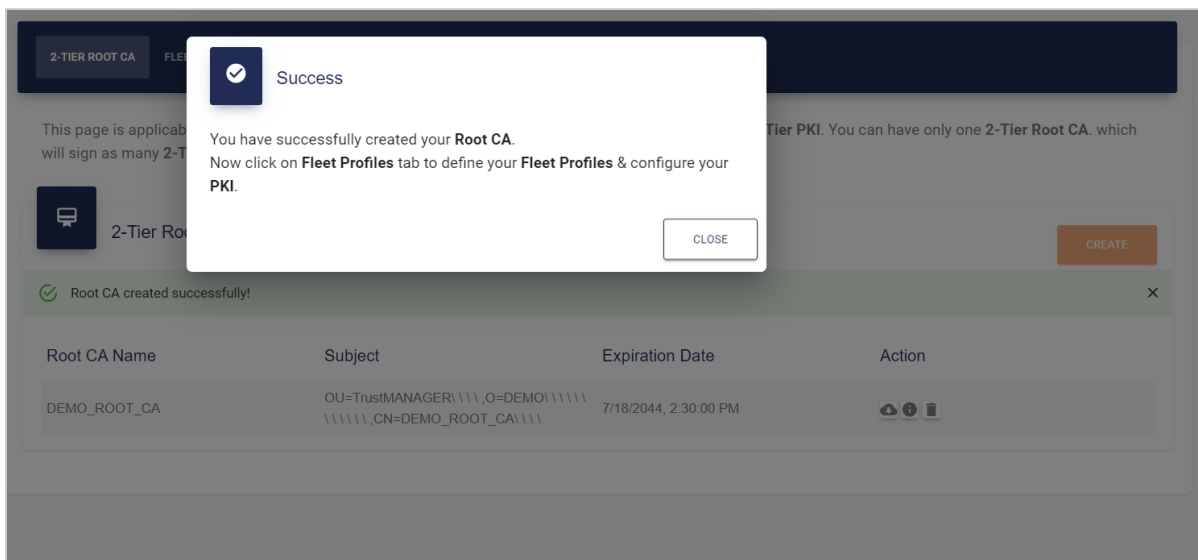


Figure-20

- Click on **FLEET Management** and **CREATE**, as shown in figure below.

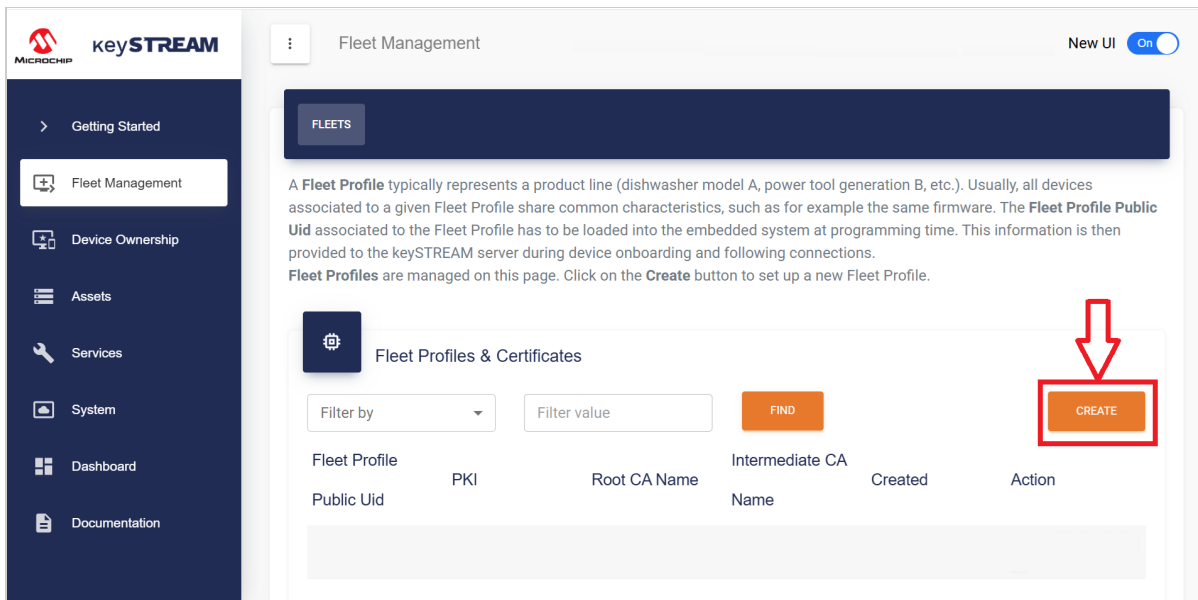


Figure-21

- A popup window appears :

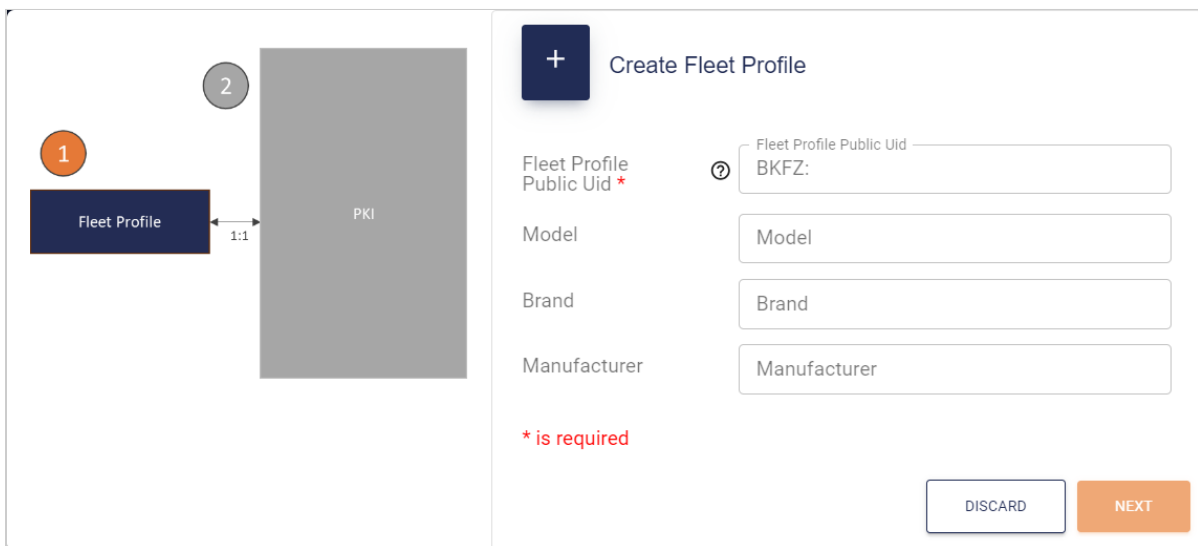


Figure-22

- Upon clicking on **NEXT**, you can choose your own Root Certificate Authority (CA) to associate with this new Fleet Profile:

- The device operational Certificate is the default device certificate validity, this refers to the end product device certificate.

Create your PKI

☐ 1-Tier PKI ☒ 2-Tier PKI

☒ keySTREAM Root CA ☐ Bring-your-Own Root CA

Root CA * Root CA

Intermediate CA Common Name (CN) * Intermediate CA Common Name (CN)

Intermediate CA Organization (O) * Intermediate CA Organization (O)

Intermediate CA Certificate Validity (years) * Intermediate CA Certificate Validity

Device Operational Certificate Validity (years) * Device Operational Certificate Validity

Automatic Leaf Certificates Renewal ☐ Renew before (days)

* is required

DISCARD COMMIT

Figure-23

- After you click on **COMMIT**, a newly configured 2-Tier Fleet Profile with its freshly created custom Root CA appears in the list below:

Profile BKFZ:DEMO_2TIER created successfully

Filter by Filter value FIND CREATE

Fleet Profile Public	PKI	Root CA Name	Intermediate CA Name	Created	Modified	Action
BKFZ:DEMO_2TIER	2-Tier	DEMO_ROOT_CA	IOT_NAGRA	2024-07-18 15:37:07.000	2024-07-18 15:37:07.000	Get PoP

Figure-24

Creating a 2-Tier Fleet Profile with your own rootCA

Some companies security policies requires the root CA to be separated from the device certificate. That's where the Intermediate Certificate Authority (ICA) brings its meaning. Here you are given the option to bring your own rootCA instead or using the keySTREAM generated rootCA

- If you need to create a 2-Tier Fleet Profile, follow the steps below:
- Under "Fleet & PKI" in the "import your Bring-your-own RootCA for 2-tier PKI" click on the "Import" button

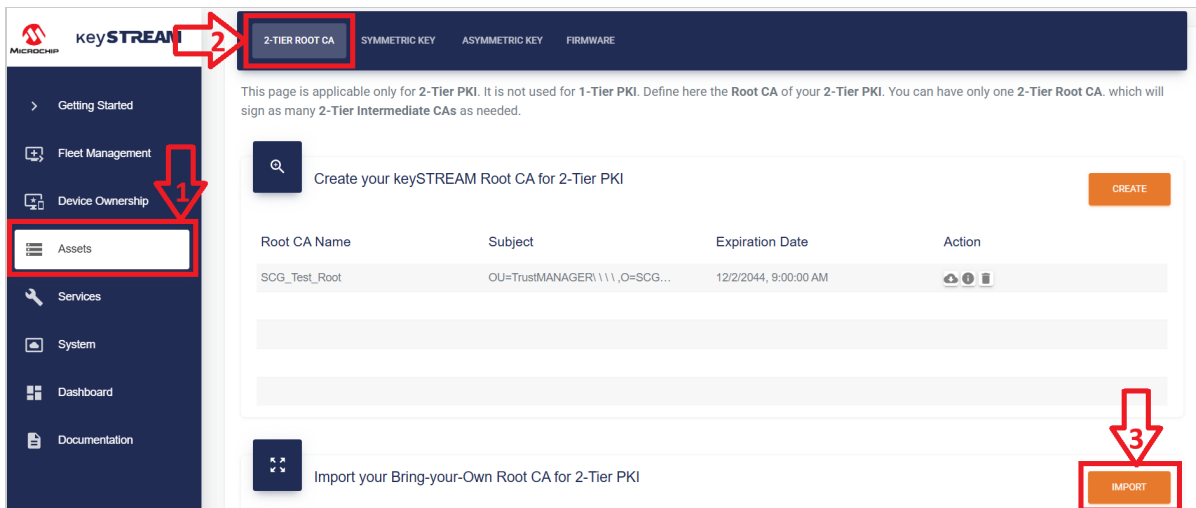


Figure-25

- Choose the .pem or .der file containing your rootCA

Import your Bring-your-Own Root CA for 2-Tier PKI

The following certificate formats are supported: *.pem and *.der.

Target embedded system limitations lead to several constraints on the Root CA. ?

Chipset Models Compatibility * ECC608

Certificate * Choose File No file chosen

* is required

DISCARD COMMIT

Figure-26

- [WARNING] Root CA of your 2-Tier PKI: You can have only one 2-Tier Root CA, which will sign as many 2-Tier Intermediate CAs as needed. - Click on **Assets** from the left panel, then select **2-TIER ROOT CA**

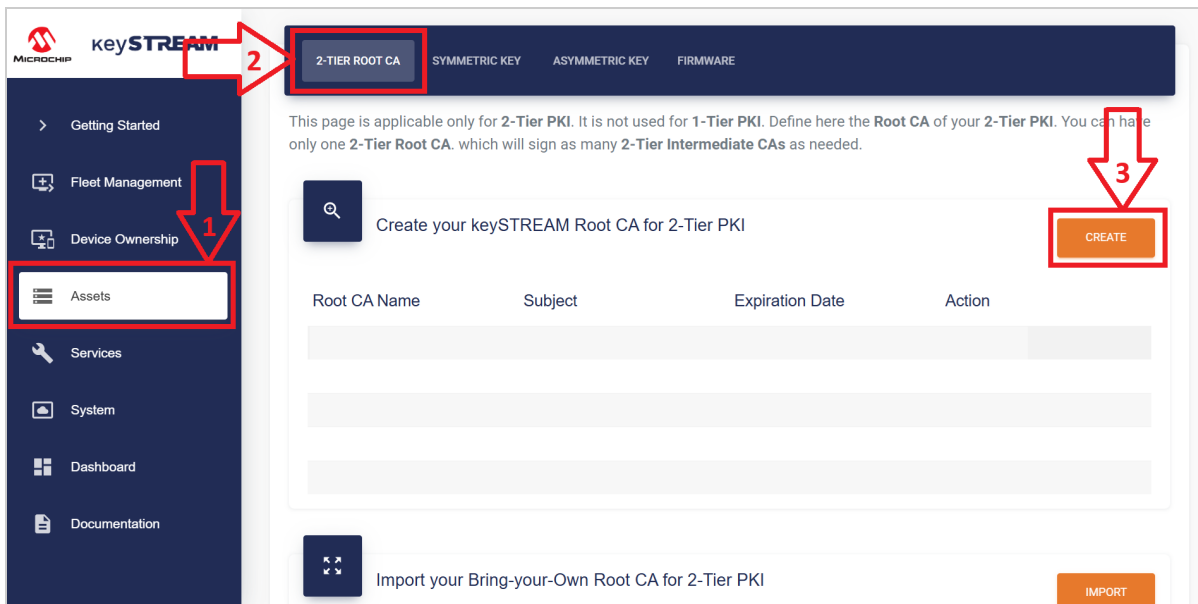


Figure-27

- Click on **Fleet Management** and **CREATE**, as shown in figure below.

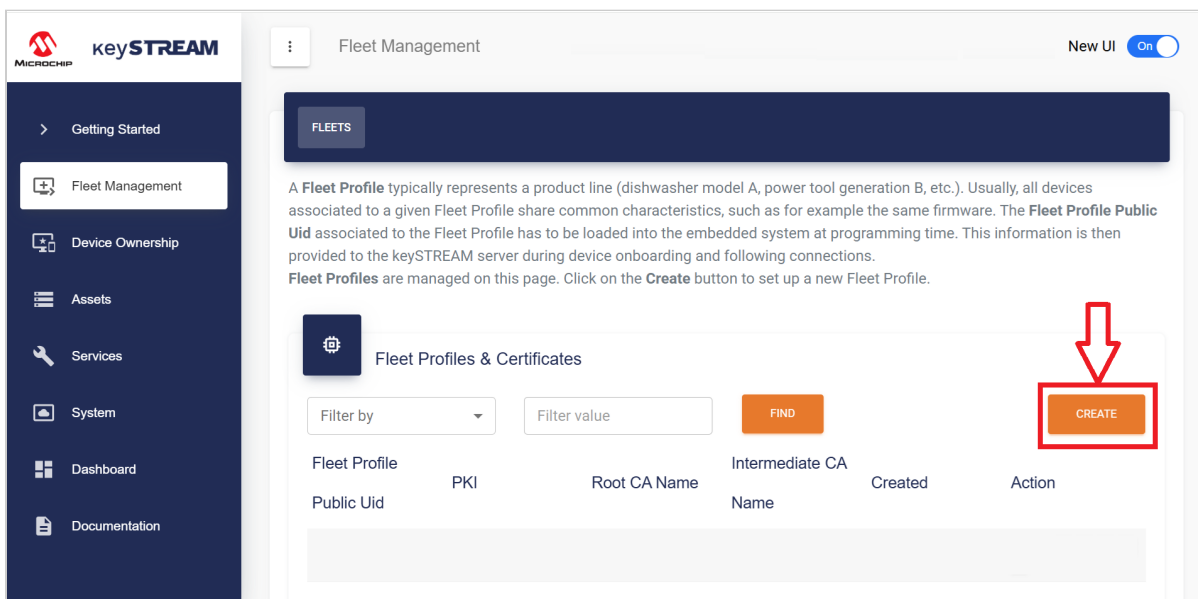
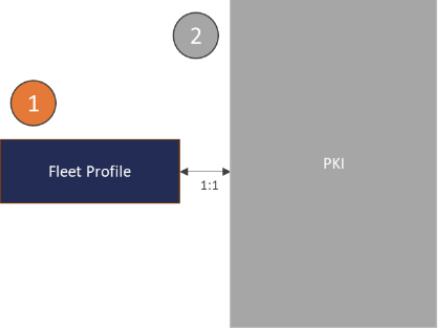


Figure-28

- A popup window appears :



Create Fleet Profile

Fleet Profile Public Uid * ②

Model

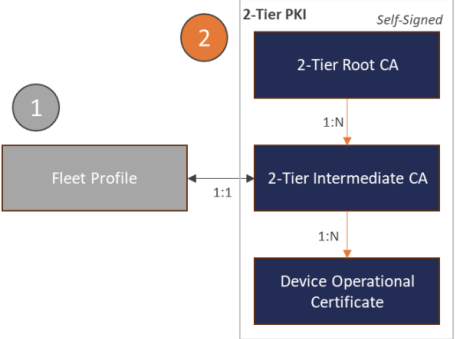
Brand

Manufacturer

* is required

Figure-29

- Upon clicking on **NEXT**, you can create your own Root Certificate Authority (CA) associated with this new Fleet Profile: - select the external rootCA previously imported



Create your PKI

☐ 1-Tier PKI ☒ 2-Tier PKI

☒ keySTREAM Root CA ☐ Bring-your-Own Root CA

Root CA *

Intermediate CA Common Name (CN) *

Intermediate CA Organization (O) *

Intermediate CA Certificate Validity (years) *

Device Operational Certificate Validity (years) *

Automatic Leaf Certificates Renewal ☐ Renew before (days)

* is required

Figure-30

- After you click on **COMMIT**, a newly configured 2-Tier Fleet Profile with its freshly created custom Root CA appears in the list below:

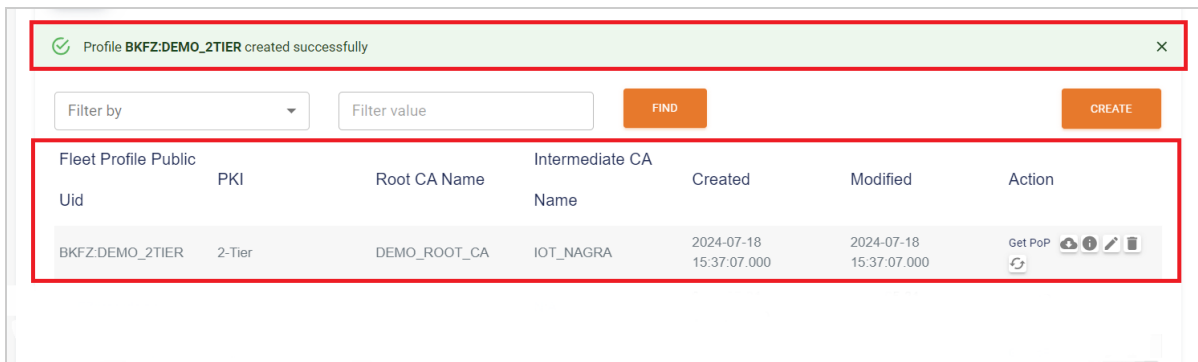


Figure-31

Obtaining an API key

- From the keySTREAM portal, you can generate an API key to be used in the TPDS UI. This key enables the TPDS backend to connect to your user account, access the signing key for its public key, and sign user components. Follow these steps to obtain your API key:
- On the left panel, click on **System**, then on the tab **API KEYS**.

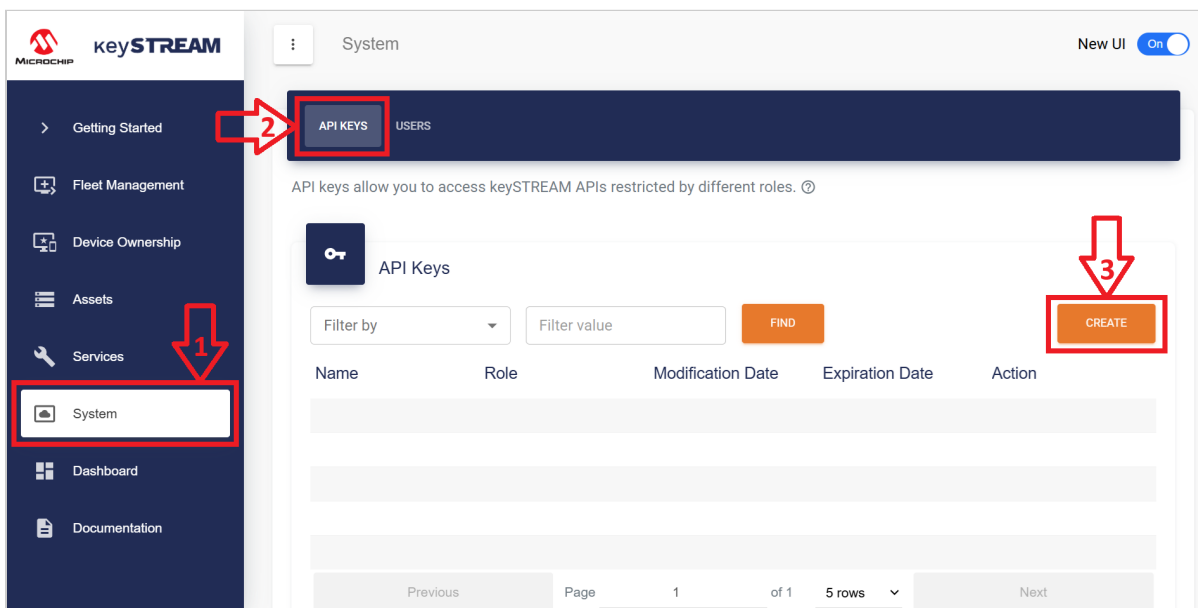
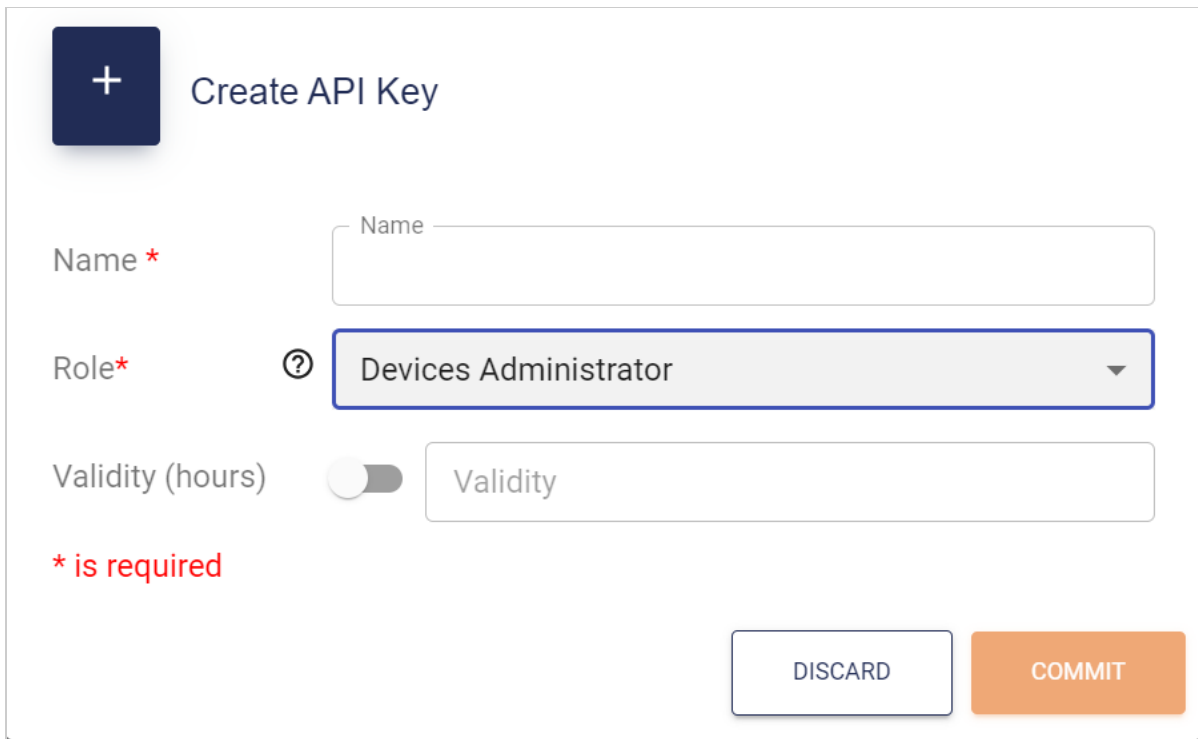


Figure-32

- On the right hand side, click on **CREATE**.
- The Create API Key popup window appears :



Create API Key

Name *

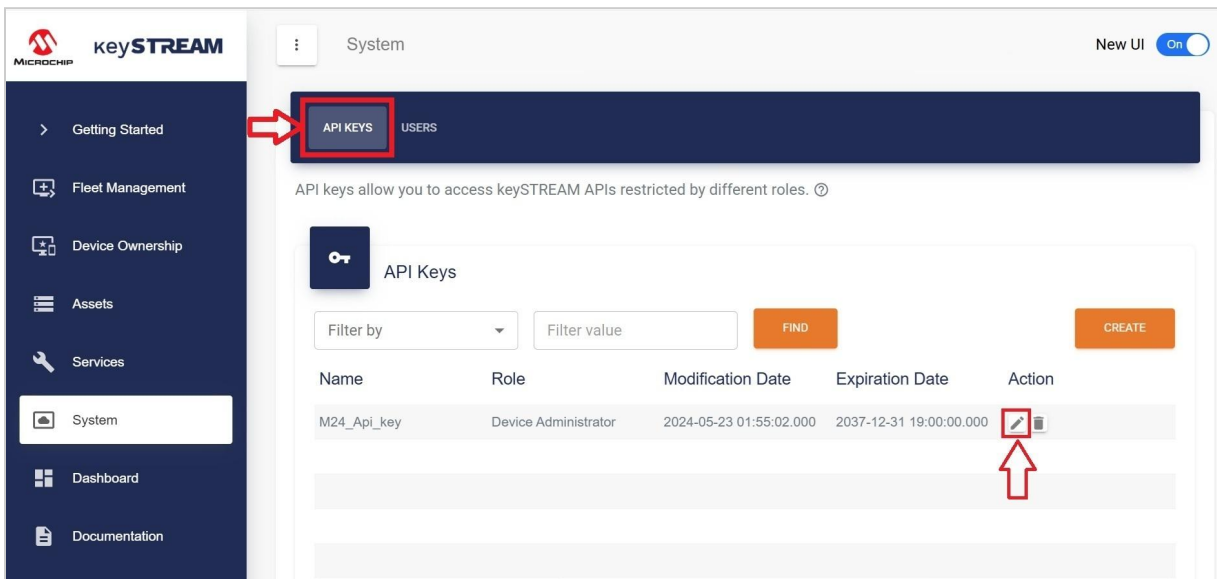
Role* ? Devices Administrator

Validity (hours) ☐

* is required

Figure-33

- The Name field can be freely chosen. - The Role to be selected here is **Devices Administrator** (aka **dmAdmin**) - The Validity can be specified in hours or left empty, in which case the API key will never expire. - Click **Commit** to create the API key, which will appear in the list of API keys. - Locate the API key that you just created, in the list, and on the Action column, click the pencil icon (tooltip: **Edit**):



keySTREAM

System

New UI ☒

API KEYS USERS

API keys allow you to access keySTREAM APIs restricted by different roles. ⓘ

API Keys

Filter by Filter value


Name	Role	Modification Date	Expiration Date	Action
M24_Api_key	Device Administrator	2024-05-23 01:55:02.000	2037-12-31 19:00:00.000	

Figure-34

- A popup window appears, **Edit Api Key**:

The screenshot shows the 'Edit Api Key' form with the following fields and buttons:

- Name:** M24_Api_key
- Secret:** A masked field with a copy icon and an eye icon.
- Role:** dmAdmin
- Basic Credentials (API Key):** A masked field with a copy icon and an eye icon. This field is highlighted with a red box, and a red arrow points from the 'Step 2' annotation to the copy icon.
- Expiration Date:** 2037-12-31 19:00:00.000
- Buttons:** CLOSE, DOWNLOAD CREDENTIALS, and RENEW.

Annotations:

- Step 1:** Hit button to see the API key (points to the eye icon in the Basic Credentials field).
- Step 2:** Hit the button to copy key to clipboard (points to the copy icon in the Basic Credentials field).

Figure-35

The API key that is expected in this use case is the value in the field "Authorization Token". Unmask it by clicking on the eye button on the right, then copy/paste it somewhere for later use. You can also click the **DOWNLOAD CREDENTIALS** button; the downloaded file contains the same "BasicCredentials" field, among other metadata.

Creating Signing Key Pairs

- Through the keySTREAM portal, you can generate or upload your own asymmetric keys, such as those used for signing firmware. The signing private key is securely protected within the keySTREAM HSM. This key will be used to retrieve the corresponding public key for verification purposes and to sign user components.
- On the left panel, click on **Services**, then on the right-hand side, click on **CREATE**.

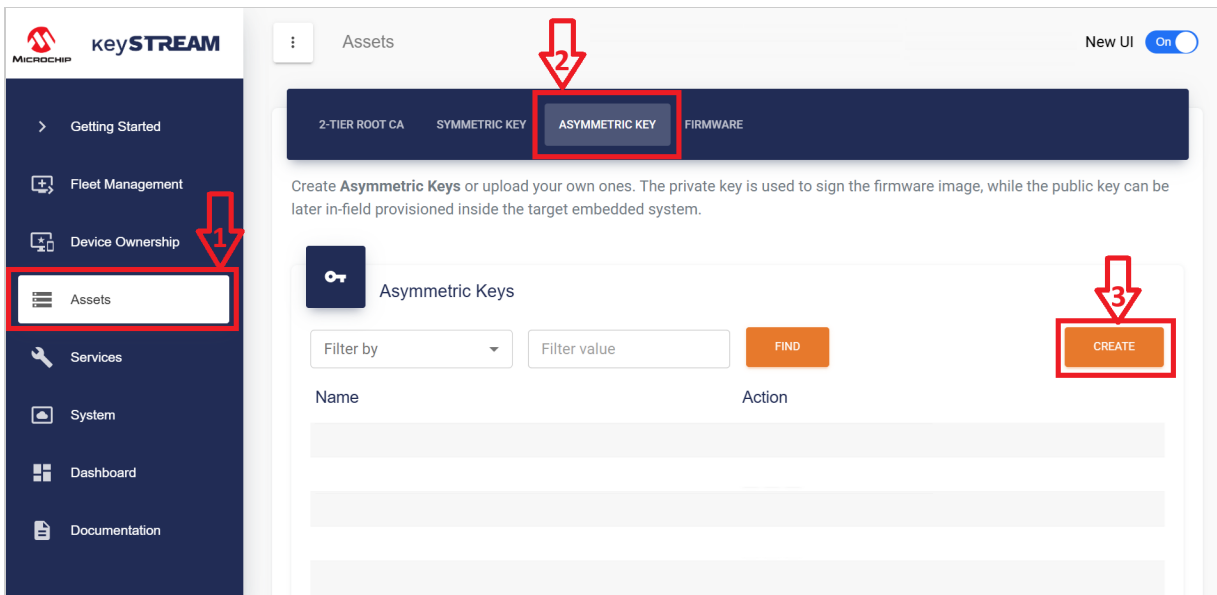


Figure-36

- A popup window appears:

- Select You ask keySTREAM to generate the entire Key Pair. You provide its name. You will have to sign with the keySTREAM sign API.
- Enter the Name for the Key and click on **COMMIT**

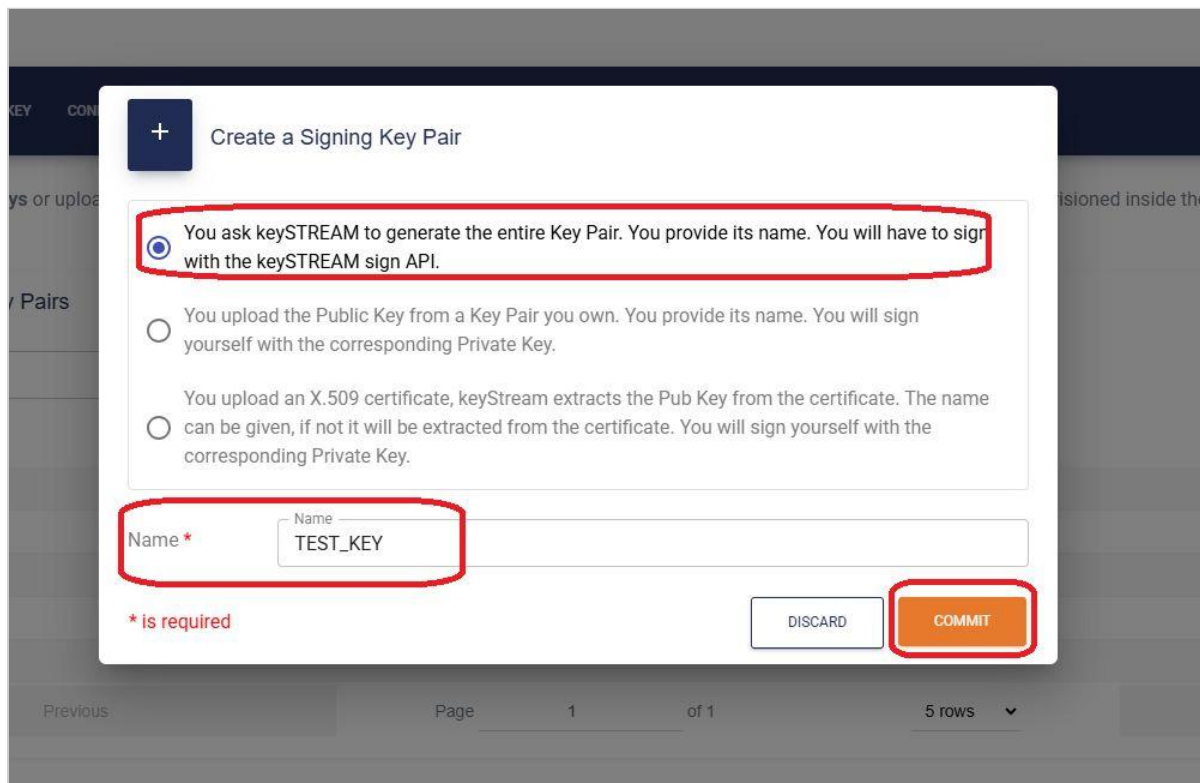


Figure-37

- keySTREAM will create an asymmetric key pair for signing operations, and this key can be used with the keySTREAM Sign REST API to sign your data.

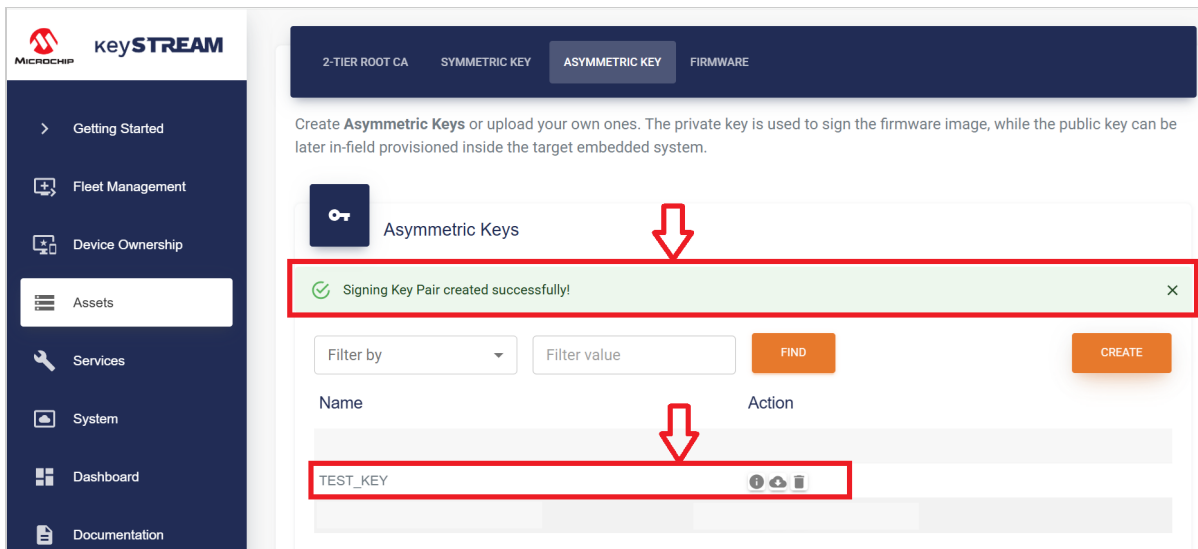


Figure-38

- Alternatively keySTREAM gives you the options to either upload your own public key or upload an entire X509 certificate containing the public key corresponding to the private you took the responsibility to generate securely in your own environment

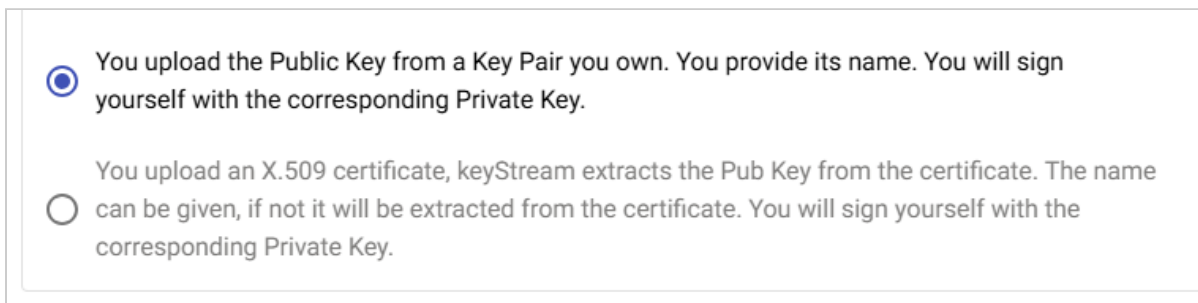


Figure-39

At this point in time of the setup, all your cryptographic assets needed to start the FOTA use case are ready.

ECC608-keySTREAM Firmware Over-The-Air Update Usecase

- Open TPDS and navigate to the Use Cases section.
- Select the kit as **CryptoAuth Pro Trust Platform**
- Select Usecase as **keySTREAM Firmware Over-The-Air Update** under ECC608-TMNGTLS

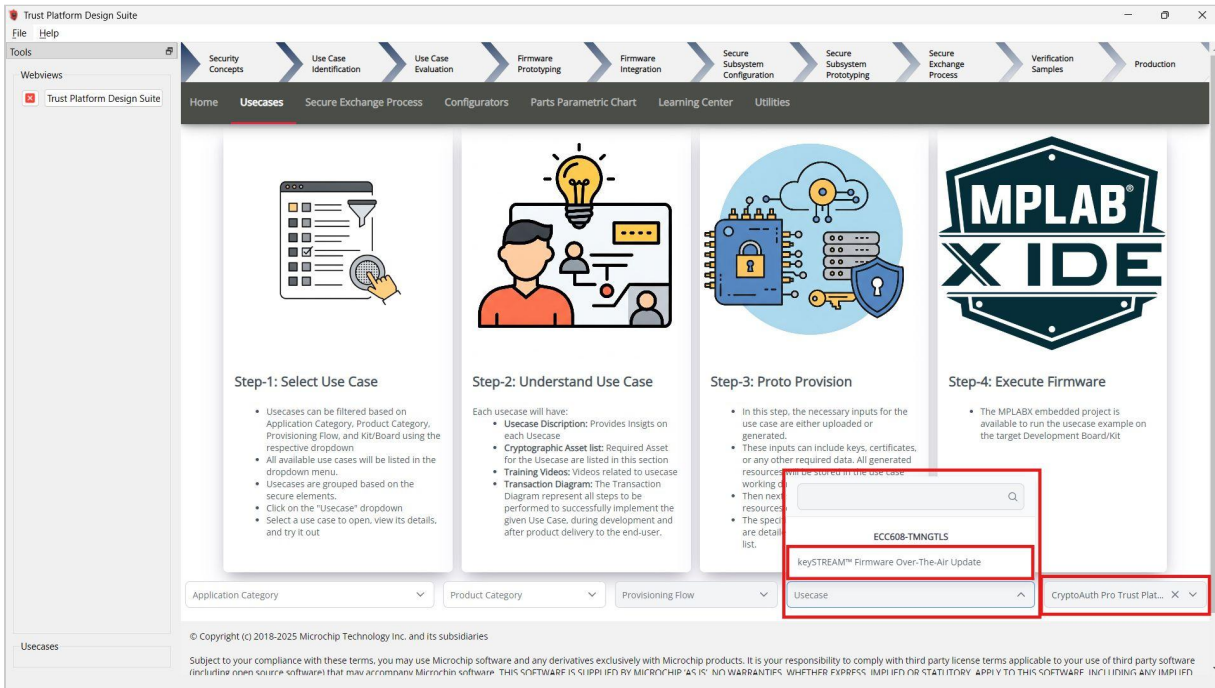


Figure-40

- The **keySTREAM™ Firmware Over-The-Air Update** Usecase will open as below:

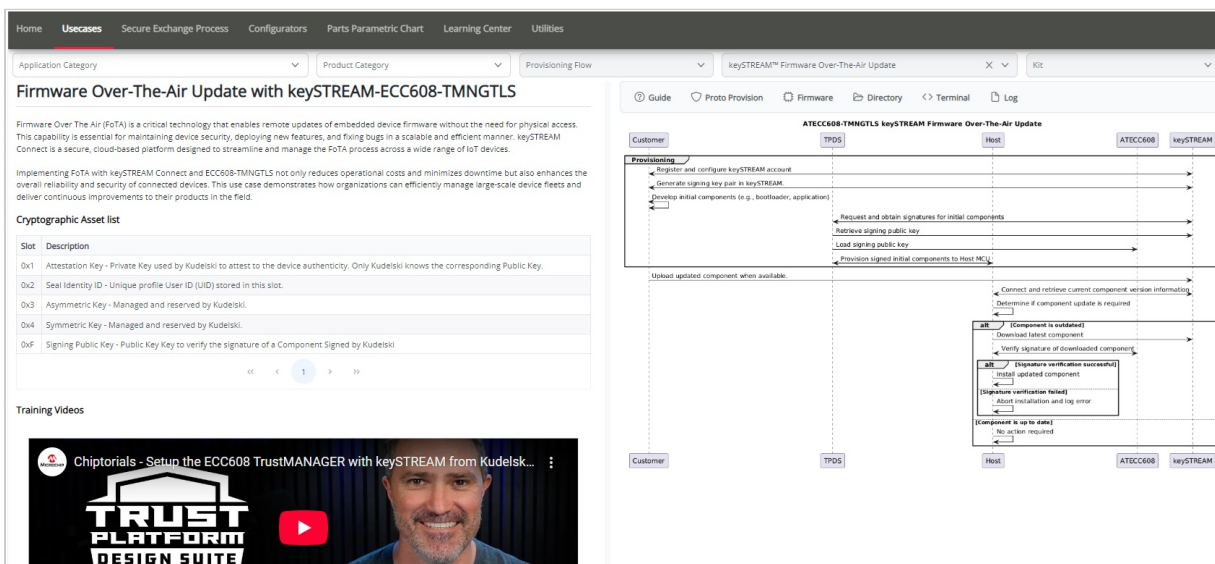


Figure-41

Generating device manifest

Device manifest is a file that describes a specific device(s). It contains public information about the device. This file is used in this Usecase for device claiming (Process through which you claim the device into your account).

- Device claiming is a one time operation. For specific device this step needs to be done only once, there is no need to redo this step for running the Usecase again. Though there are methods to change ownership of the device, we recommend claiming device on one keySTREAM account and using it on the same account.
- Factory program the CryptoAuth Pro Trust Platform kit (EV89U05A). This step loads firmware so TPDS can talk to the device.
- Go to the TPDS "Utilities", click on "Device Interactions", select the EV89U05A board.

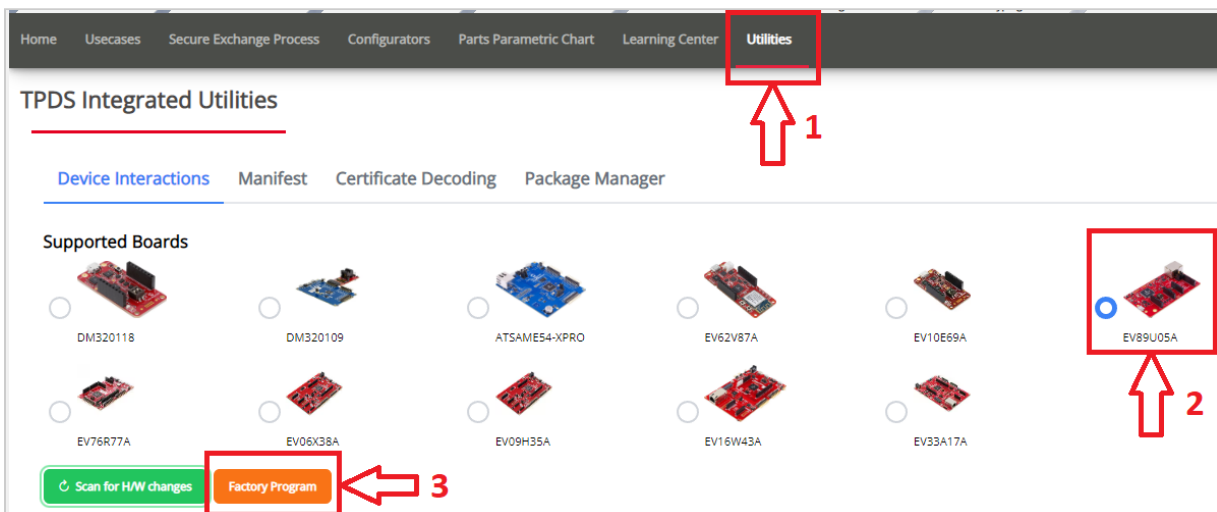


Figure-42

- Click **Factory program** and wait until the process is completed, you will see a message when it is done

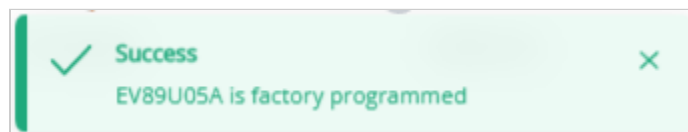


Figure-43

- Now go to "Utilities", click on "Manifest" and follow the instructions below

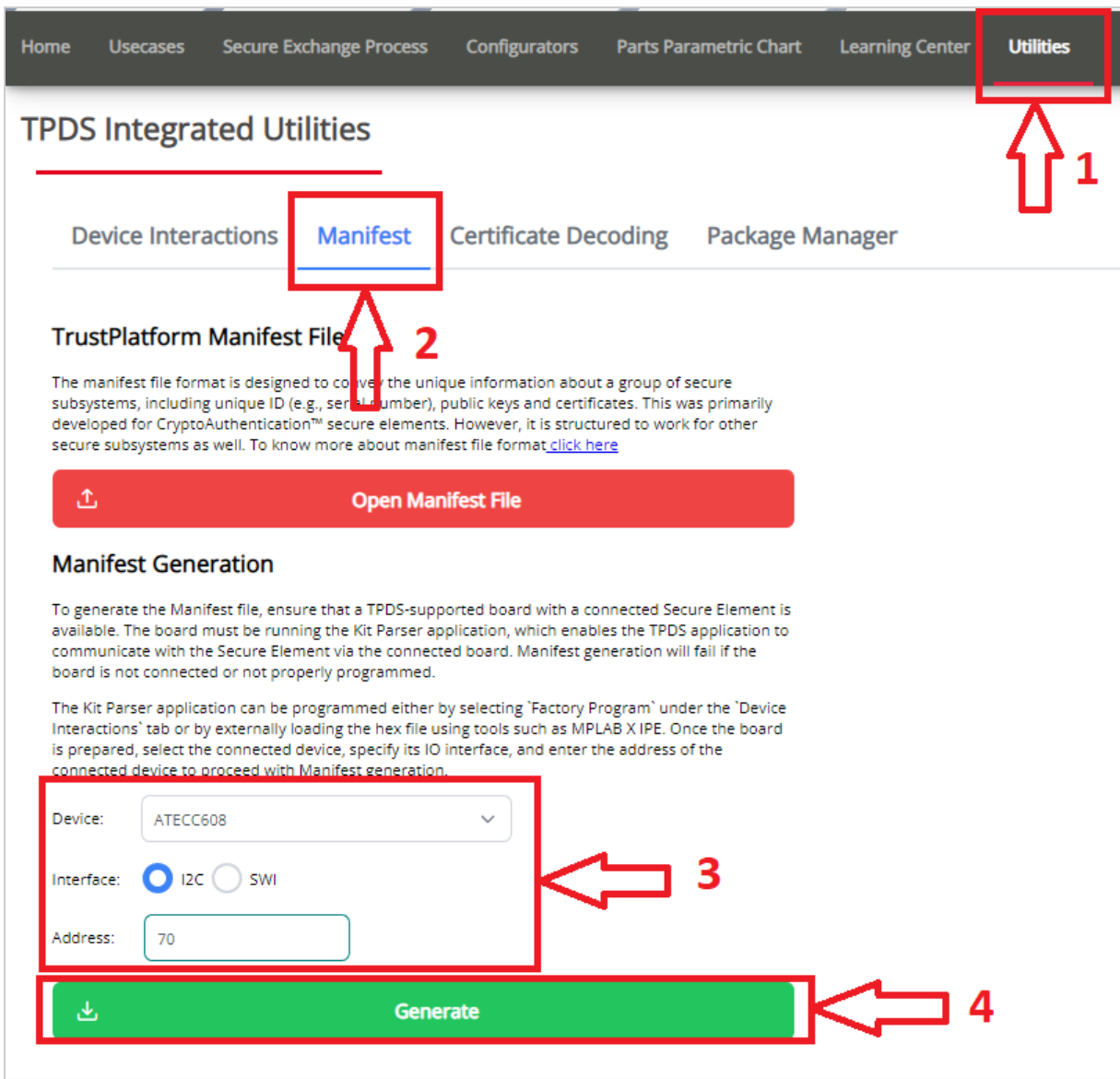


Figure-44

- Select **ATECC608** as **Device**, **I2C** as **Interface**, **70** as **Address** and click on **Generate** - Once manifest is generated, there will be a pop-up message with the generated manifest file location

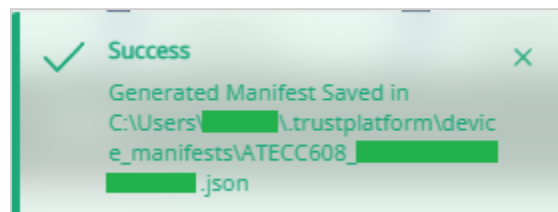


Figure-45

- Navigate to `~/trustplatform/device_manifest` and observe the created files. There is the JSON file with your device UID and other metadata. This is the file which we can use to upload to TPDS to obtain our device UID, also known as serial number.





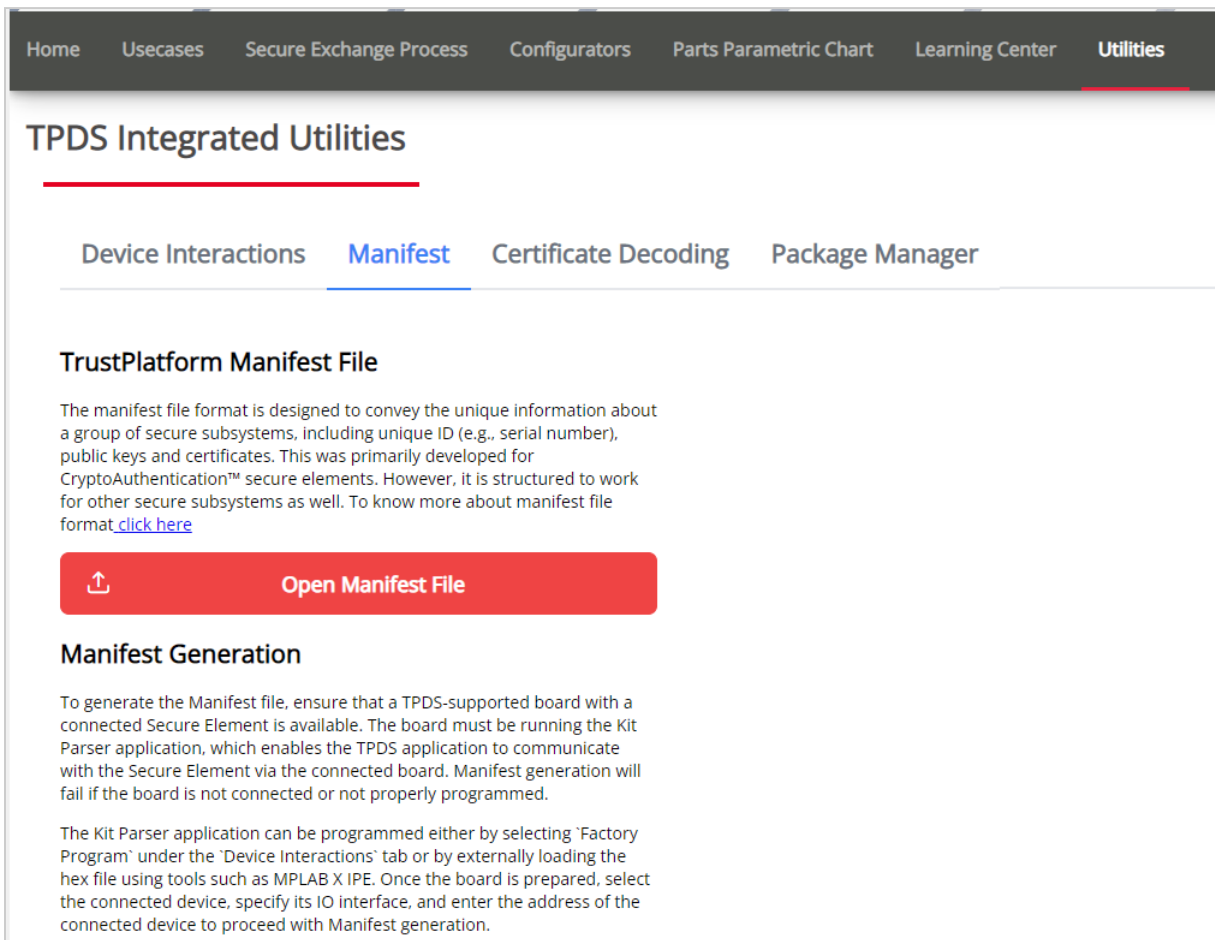
Name	Date modified	Type	Size
 ATECC608_0125D74C1E5D3C674034	6/25/2025 5:04 PM	JSON Source File	2 KB
 manifest_ca	6/25/2025 4:59 PM	Security Certificate	1 KB
 manifest_ca.key	6/25/2025 4:59 PM	KEY File	1 KB
 ATECC608_0125D74C1E5D3C674034	6/27/2025 2:36 PM	Compressed (zipped) Fo...	1 KB

Figure-46

- Open your TPDS tab, go to "Utilities" on the horizontal pane and then "Manifest"




TPDS Integrated Utilities

Device Interactions **Manifest** Certificate Decoding Package Manager

TrustPlatform Manifest File

The manifest file format is designed to convey the unique information about a group of secure subsystems, including unique ID (e.g., serial number), public keys and certificates. This was primarily developed for CryptoAuthentication™ secure elements. However, it is structured to work for other secure subsystems as well. To know more about manifest file format [click here](#)

 **Open Manifest File**

Manifest Generation

To generate the Manifest file, ensure that a TPDS-supported board with a connected Secure Element is available. The board must be running the Kit Parser application, which enables the TPDS application to communicate with the Secure Element via the connected board. Manifest generation will fail if the board is not connected or not properly programmed.

The Kit Parser application can be programmed either by selecting 'Factory Program' under the 'Device Interactions' tab or by externally loading the hex file using tools such as MPLAB X IDE. Once the board is prepared, select the connected device, specify its IO interface, and enter the address of the connected device to proceed with Manifest generation.

Figure-47

- Select "Open Manifest File" and then in the window, navigate to `~/trustplatform/device_manifest` directory and select your JSON file that TPDS generated for you. - Select that file and you will see a window pop up asking you to upload a custom Manifest Signer Certificate, select Yes.



Figure-48

- Select the "manifest_ca" certificate file and then you will see it display a new tab in TPDS showing the device UID and other metadata

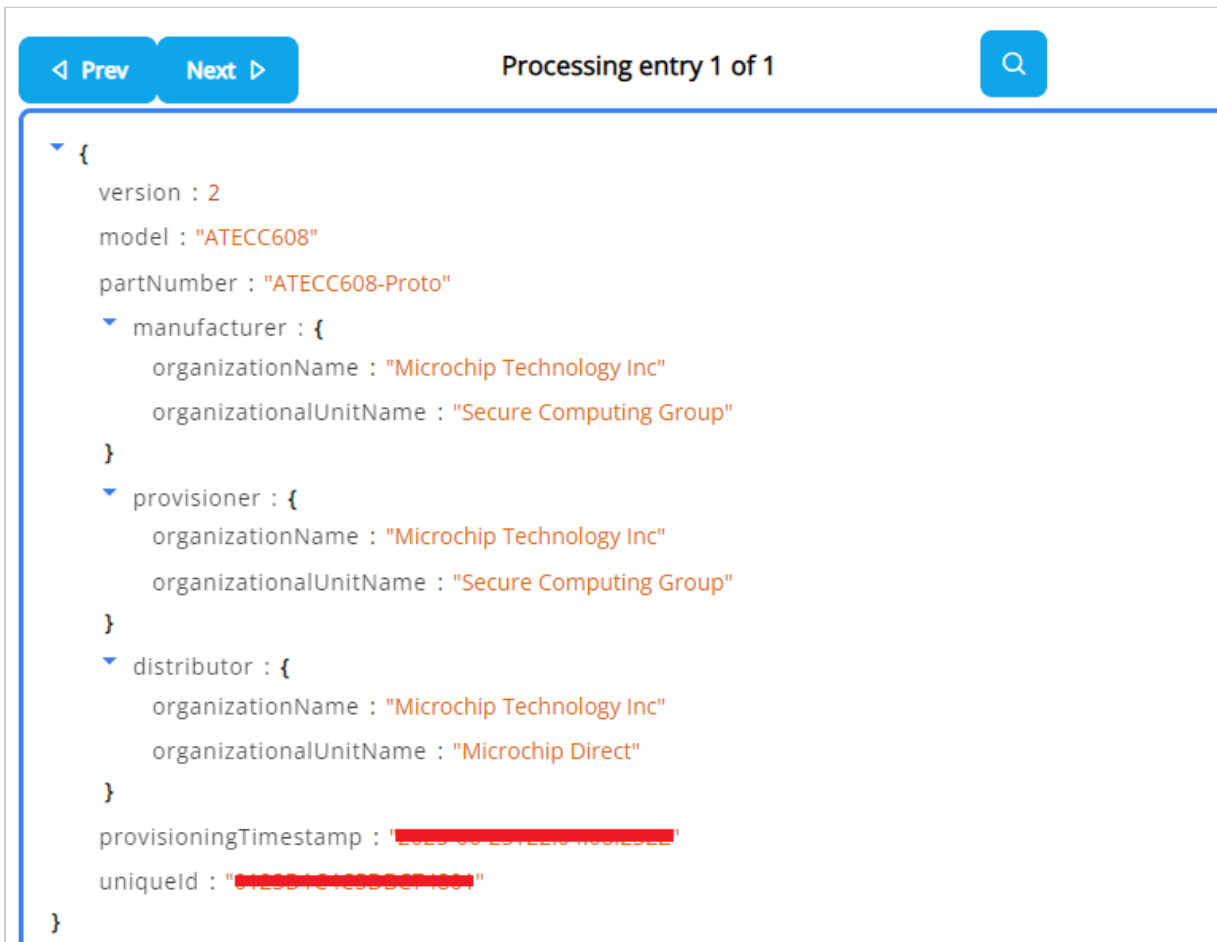


Figure-49

- Note the uniqueId field in your previously mentioned text document as we will need this in the next step when we provide it to keySTREAM. - Navigate to your keySTREAM tab and switch to the "Device Ownership" tab and then "Device Claiming". You may paste your Chip Unique ID into the field and then hit the "Commit" button.

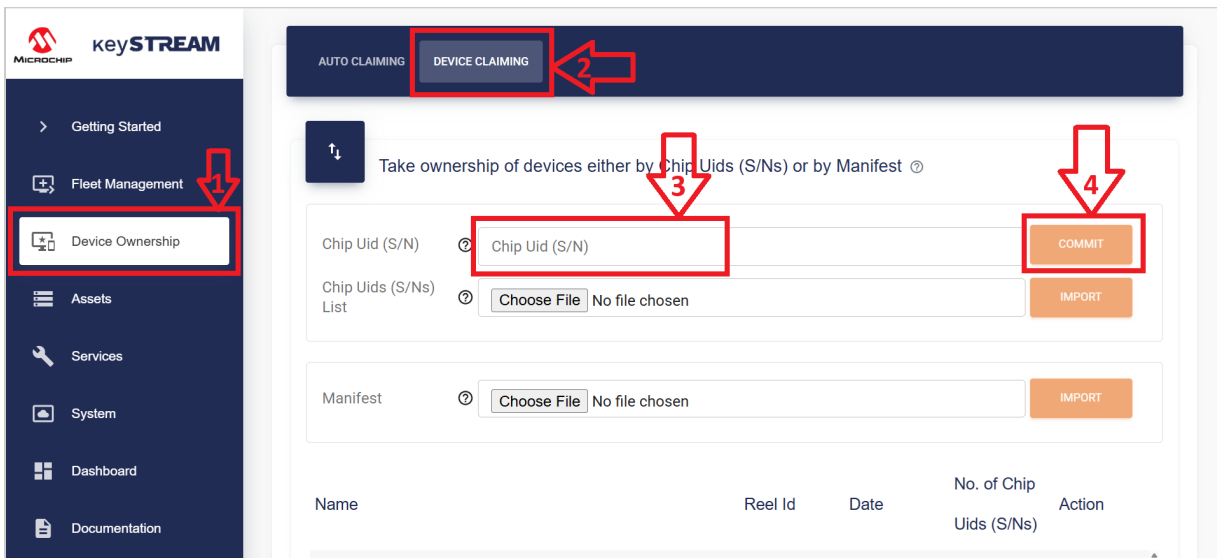


Figure-50

This completes the Device Claiming process on keySTREAM.

Provisioning Usecase Resources

This step prepares the embedded system by gathering the necessary resources, generates the firmware resources, and provisions the device accordingly.

- Memory layout of the Usecase firmware
 - **Bootloader:** 0x00000000 - 0x0000FC00
 - **Bootloader Metadata:** 0x0000FC00 - 0x00010000
 - **Application:** 0x00010000 - 0x0006FC00
 - **Application Metadata:** 0x0006FC00 - 0x00070000
 - **SmartEEPROM:** 0x00078000 - 0x0007A000
- Double-check that you selected the right target development kit.

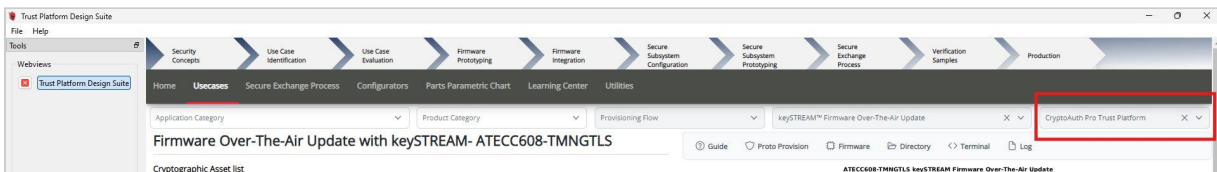


Figure-51

- Click on Proto Provision

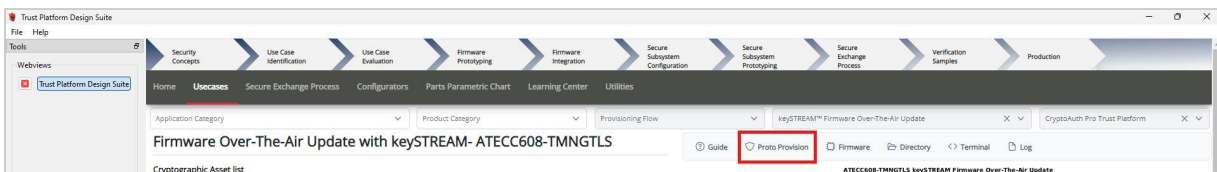


Figure-52

- Provide user Inputs:

Make sure there is **no space** before or after each character strings

- **Fleet Profile Public UID** - Enter the Fleet Profile Public UID Created in keySTREAM at [Creating a Fleet Profile](#)
- **Authorization Token** (API Key) - Provide keySTREAM Authorization Token which you have copied in [Obtaining an API key](#), refer this section for more details.
- **Signing Key Name** - Provide name of the Signing Key from keySTREAM created at [Creating Signing Key Pairs](#), refer this section for more details.
- **WiFi SSID** - Provide WiFi SSID to which device needs to connect
- **WiFi Password** - Provide Wifi Password for device to connect

The screenshot shows a 'User Inputs' dialog box. It has a title bar with a close button. The main area contains five input fields, each with a label to its left: 'Fleet Profile Public UID:', 'Authorization Token:', 'Signing Key Name:', 'WIFI SSID:', and 'WIFI Password:'. Each field has a corresponding text input box. These five fields are enclosed in a red rectangular border. Below these fields, there are two more sections. The first is 'Component 1 (Bootloader):' followed by a blue button with a plus sign and the text '+ Choose'. The second is 'Component 1 Info Address (hex):' followed by a text input box. At the bottom left of the dialog is a blue button with a circular arrow icon and the text 'reset'. At the bottom right is a blue button with a gear icon and the text 'Proto Provision'.

Figure-53

- Click on Proto Provision, this will generate the resources for firmware project.
- Click **No** in the pop-up asking to load generated resources into Secure Element. At this step we are only generating the resources to build the component binaries.

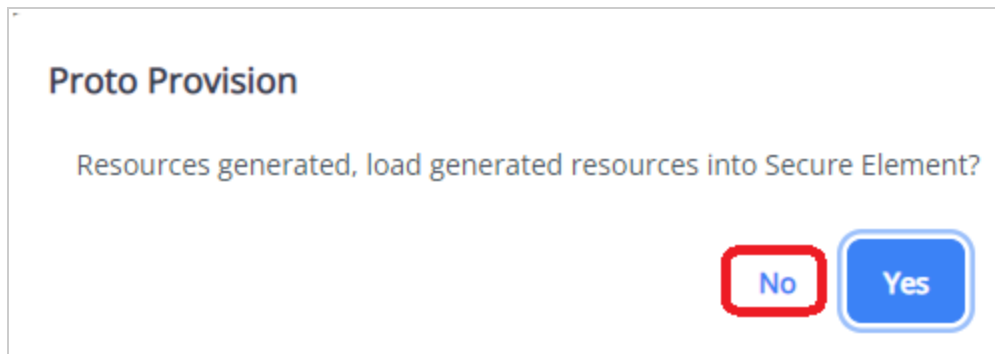


Figure-54

- The necessary resources will be created in the Usecase working directory `~/.trustplatform/keystream_fota`:

- **compname_version.hex/bin**: Processed Component along with its signature in hex and bin file format. Name and Version are fetched from the Component meta data. Refer to Component meta data in the firmware for more details.
- **combined_component.hex/bin**: If both components are provided, they will be merged to generate this combined binary.
- **KEY_NAME.pem**: Signing Public Key retrieved from keySTREAM.
- To open the use case working directory containing the use case resources click on the **Directory** button.

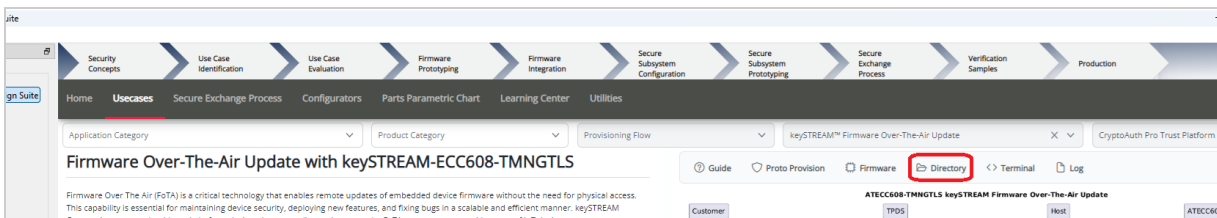


Figure-55

- Now open the firmware project by clicking on the **Firmware** button.

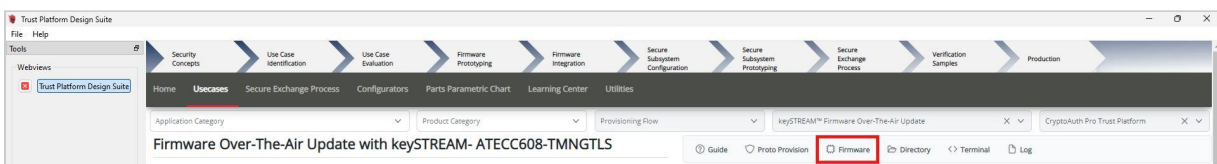


Figure-56

- MPLAB will open and will have the two open projects

- **sg41_sboot** - This is the bootloader firmware project
- **KTA_FOTA** - This is the application firmware project
- Build both the projects by right click on project and hit **Clean and Build** button to have the HEX images to be signed ready

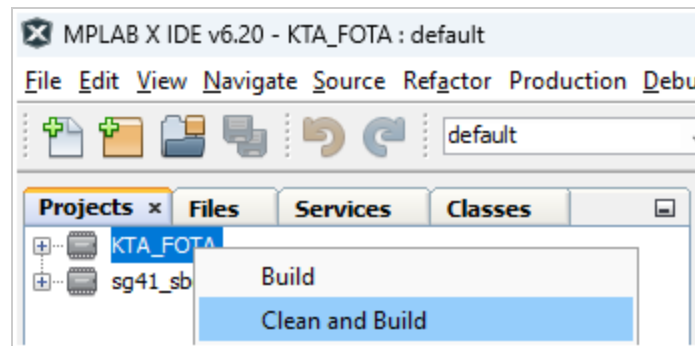


Figure-57

- Click again on **Proto Provision** to create signed components from the generated binaries

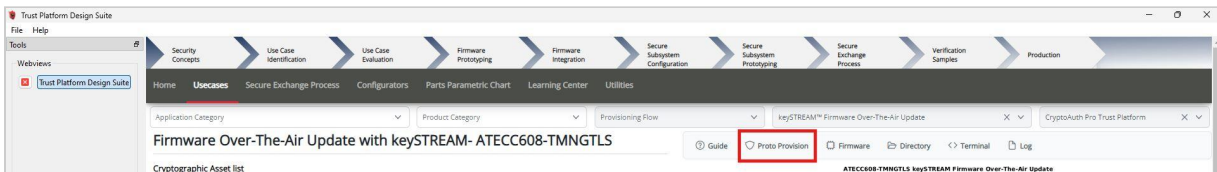


Figure-58

- Provide user Inputs:

- Leave Fleet Profile Public UID, Authorization Token, Signing Key Name, WiFi SSID, WiFi Password inputs blank to use from the previously generated resources.
- **Component1 (Bootloader):** - Upload the Component1 (sg41_sboot bootloader) hex file that was built in the previous step. Please note that the binaries are generated only after the project has been built, the default location will be: `~/.trustplatform/keystream_fota`.
- **Component1 Info Address (hex):** - Each component should include metadata detailing its version, image address, size, signature locations, and other relevant information. Please specify the location of this metadata. For this use case, the Bootloader metadata is located at FC00.
- **Component2 (Application):** - Upload the Component2 (KTA_FOTA application) hex file that was built in the previous step. Please note that the binaries are generated only after the project has been built, the default location will be: `~/.trustplatform/keystream_fota`.
- **Component2 Info Address (hex):** - Each component should include metadata detailing its version, image address, size, signature locations, and other relevant information. Please specify the location of this metadata. For this use case, the FoTA application metadata is located at 6FC00.
- **Notes**
 - Only one of the two components is required; providing either component is sufficient.
 - If both Component1 and Component2 are provided, they will undergo signing operations, then be combined into a single image and programmed during proto provisioning.
 - If only one component is provided, it will be processed for signing operations but will not be programmed.

User Inputs

Signing Key Name:

WIFI SSID:

WiFi Password:

Component 1 (Bootloader): + Choose sg41_sboot.X.production.hex

Component 1 Info Address (hex):

Component 2 (Application): + Choose KTA_FOTA.X.production.hex

Component 2 Info Address (hex):

reset Proto Provision

Figure-59

- Click on **Proto Provision** to sign the given components using keySTREAM and save in the Usecase working directory. - To program the components onto the board:
- Click Yes in the pop-up to load the keySTREAM signing public key onto the ATECC608-TMNGTLS and program the signed components onto the CryptoAuth Pro Trust Platform boards. This is typically done only for initial components.

Proto Provision

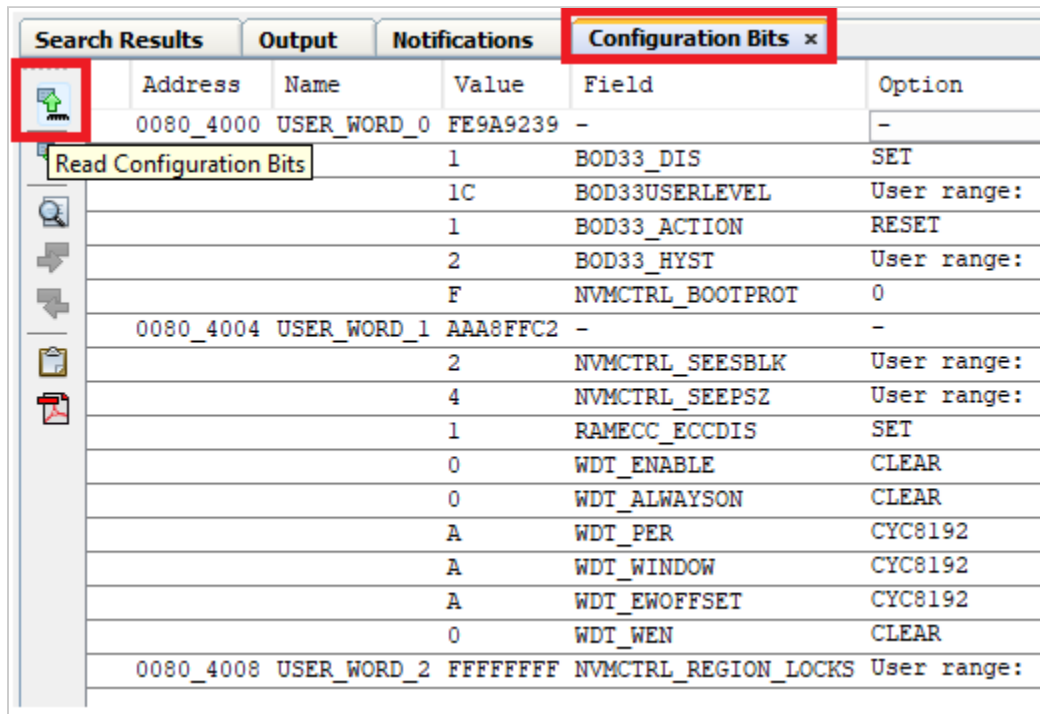
Resources generated, load generated resources into Secure Element?

No Yes

Figure-60

- Once the programming process is complete, please launch the Terminal application (e.g., TeraTerm) on your computer if it has not been set up initially.
- Connect to the Virtual COM port and configure the serial settings as follows:
 - Baud : 115200
 - Data : 8 Bits
 - Parity : None
 - Stop : 1 Bit
 - Flow Control : None
- Press the Reset button on CryptoAuth Pro Trust Platform Development Kit

- Set Configuration Bits for SmartEEPROM to manage FoTA Persistent data, Open Configuration Bits window on MPLAB X IDE by navigating to Window -> Target Memory Views -> Configuration Bits
- Read Configuration Bits from the connected device



Address	Name	Value	Field	Option
0080_4000	USER_WORD_0	FE9A9239	-	-
		1	BOD33_DIS	SET
		1C	BOD33USERLEVEL	User range:
		1	BOD33_ACTION	RESET
		2	BOD33_HYST	User range:
		F	NVMCTRL_BOOTPROT	0
0080_4004	USER_WORD_1	AAA8FFC2	-	-
		2	NVMCTRL_SEESBLK	User range:
		4	NVMCTRL_SEEPSZ	User range:
		1	RAMECC_ECCDIS	SET
		0	WDT_ENABLE	CLEAR
		0	WDT_ALWAYS ON	CLEAR
		A	WDT_PER	CYC8192
		A	WDT_WINDOW	CYC8192
		A	WDT_EWOFFSET	CYC8192
		0	WDT_WEN	CLEAR
0080_4008	USER_WORD_2	FFFFFFFF	NVMCTRL_REGION_LOCKS	User range:

Figure-61

- Set NVMCTRL_SEESBLK as 2 and NVMCTRL_SEEPSZ as 4. This allocates 8192 bytes of the flash for SmartEEPROM
- Program Configuration Bits to the connected device

Search Results					
Output					
Notifications					
Configuration Bits x 1					
Address	Name	Value	Field	Option	
0080_4000	USER_WORD_0	FE9A9239	-	-	
3	Program Configuration Bits	1	BOD33_DIS	SET	
		1C	BOD33USERLEVEL	User range: 0x0 - 0xFF	
		1	BOD33_ACTION	RESET	
		2	BOD33_HYST	User range: 0x0 - 0xF	
		F	NVMCTRL_BOOTPROT	0	
0080_4004	USER_WORD_1	AAA8FFC2	-	-	
2		2	NVMCTRL_SEESBLK	User range: 0x0 - 0xF	
		4	NVMCTRL_SEEPSZ	User range: 0x0 - 0x7	
		1	RAMECC_ECCDIS	SET	
		0	WDT_ENABLE	CLEAR	
		0	WDT_ALWAYS ON	CLEAR	

Figure-62

- Press the Reset button on CryptoAuth Pro Trust Platform Development Kit
 - Verify that Configuration bits are set correctly by checking SMARTEEPROM Size (SBLK = 2, PSZ = 4) - 8192 message in the console.
- Review the output message in the console:

```

COM20 - Tera Term VT
File Edit Setup Control Window Help

----- KeySTREAM FOTA Usecase - Bootloader -----
ECC608-TMNGTLS Initialization is successful
Serial Number: 01 23 E3 CA 04 E5 F1 9A 01
SHA256 calculation is started for 0x10000:0x5FC00
Digest:
A 27 D9 B2 AA AA E1 A1 24 07 3C F4 67 E6 54 FA
F1 9F B7 AB 04 E5 BB 12 E1 BA 0E D3 A7 0C 29 16
Signature location: 0x6FC30
Signature:
06 3B BB 94 F1 07 A6 51 B4 2B E9 A5 E9 FE BA 84
50 E2 00 CF 12 71 85 A2 1C A8 F9 63 3A 27 A0 AC
70 7E 3E F1 6C 2C B7 15 79 A6 B6 68 D9 60 3A D8
B1 2C 0E AB A6 FA 91 06 04 CB C4 E6 C5 F9 40 FA
Applications verified successfully.
Jumping to the application at location 0x10000

=====
keySTREAM Connect with WINCS02 - Demo
  Built: Nov 9 2025 04:54:31
  KTA Version: 1.4.1
  Application Version: 1.2.0
  Active Flash - BANK A
  SMARTEEPROM Size (SBLK = 2, PSZ = 4) - 8192
=====
IAPP_WINC1: Starting Cryptoauthlib
IAPP_WINC1: atcab_init is successful
IAPP_WINC1: Reading device serial number
IAPP_WINC1: Device serial number: 01 23 E3 CA 04 E5 F1 9A 01
IAPP_WINC1: Read KTA Fleet Profile ID
IAPP_WINC1: Sealed Fleet Profile ID in Slot2: 4H6E:uid
WiFil: WINC: Device ID = 29c70053
WiFil: 0: Seq No = fffffffe0, Version = 04030000, Source Address
= 60000000
WiFil: Firmware Version: 3.0.0 [17:52:21 Jun 17 2025]
WiFil: Driver Version: 3.0.0
IAPP_WINC1: Setting REG domain to GEN
IAPP_WINC1: Set Reg Domain -> SUCCESS

```

Figure-63

Note: The [BOOT <293>] error shown in the log is expected during initial setup because the Secure Boot public key resides in Slot 15. Any subsequent in-field key rotations performed by keySTREAM will update the public key in Slot 14.

Preparing new Application Image for FoTA update

- Navigate back to MPLAB X IDE and in the "KTA_FOTA" Project, expand drop-down for the "app_winc_s02.h" to increment APP_COMPONENT_VERSION

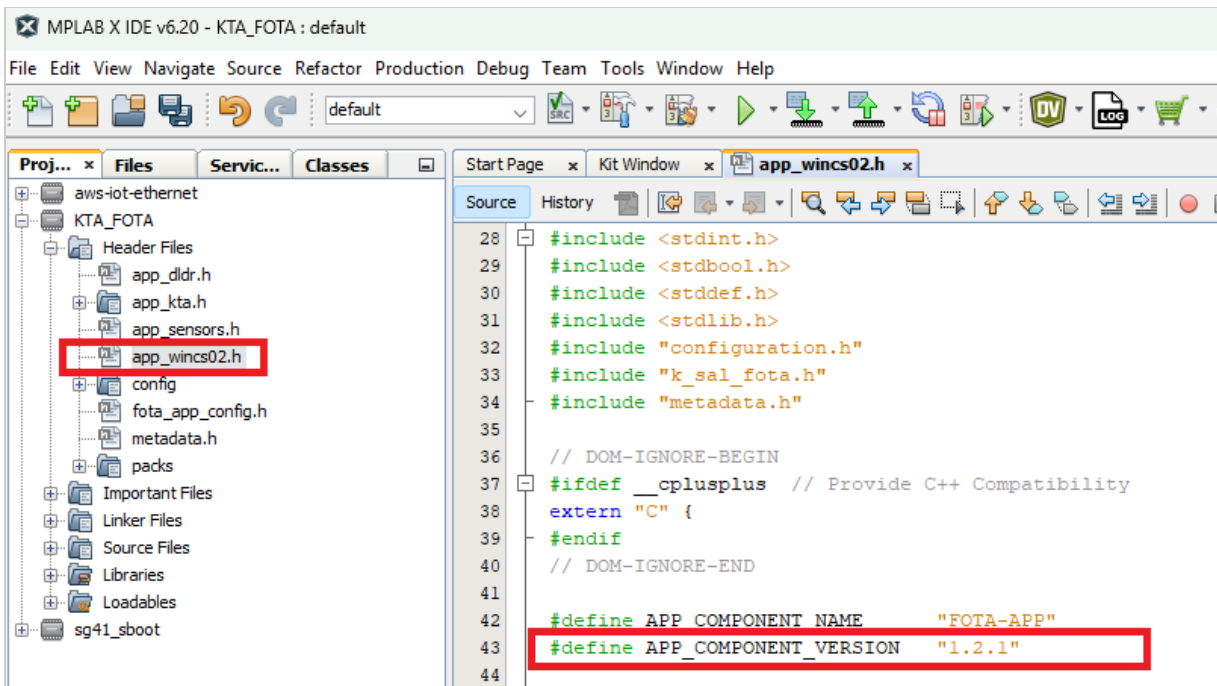


Figure-64

- Clean and Build KTA_FOTA application with new version... Wait for the build process to complete and generate the binaries.

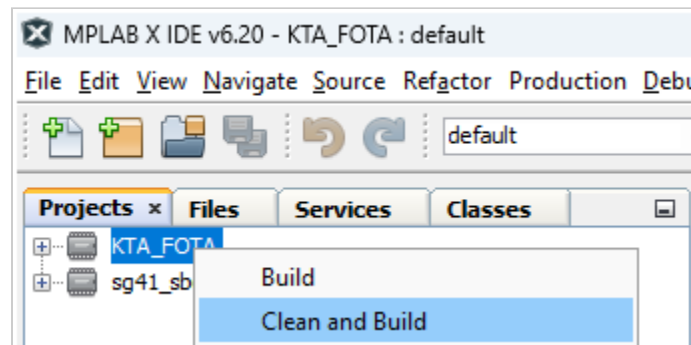


Figure-65

- Click on **Proto Provision** to create signed components from the generated binary

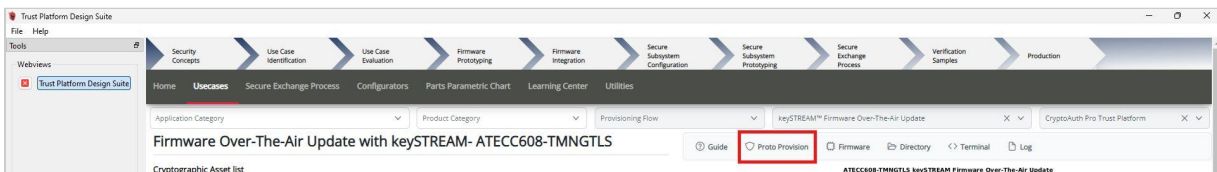


Figure-66

- Click on **reset** to clear previous Component selection if exists

Figure-67

- Leave Fleet Profile Public UID, Authorization Token, Signing Key Name, WiFi SSID, WiFi Password inputs blank to use from the previously generated resources.
- **Component2 (Application):** - Upload the Component2 (KTA_FOTA application) hex file that was built in the previous step. Please note that the binaries are generated only after the project has been built, the default location will be: `~/trustplatform/keystream_fota`.
- **Component2 Info Address (hex):** - Each component should include metadata detailing its version, image address, size, signature locations, and other relevant information. Please specify the location of this metadata. For this use case, the FoTA application metadata is located at 6FC00.
- Click on `Proto Provision` button, this will generate the signed Component for the new binary with incremented APP_COMPONENT_VERSION.
- Click **No** in the pop-up asking to load generated resources into Secure Element.

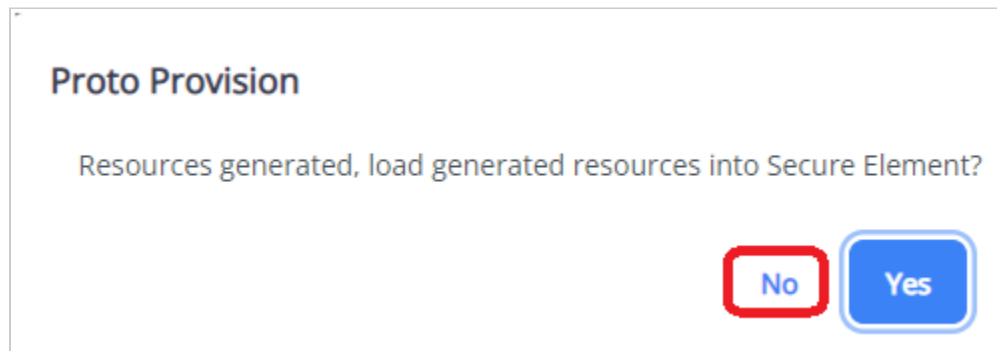


Figure-68

- The signed FOTA-APP_1.2.1.bin (may vary based on the VERSION values) and hex will be created in the Usecase working directory `~/trustplatform/keystream_fota`.

Preparing a FOTA campaign in the KeyStream Cloud

- Go back to the keySTREAM UI under the **Assets** option -> Then click on **Firmware** -> Then click on the component

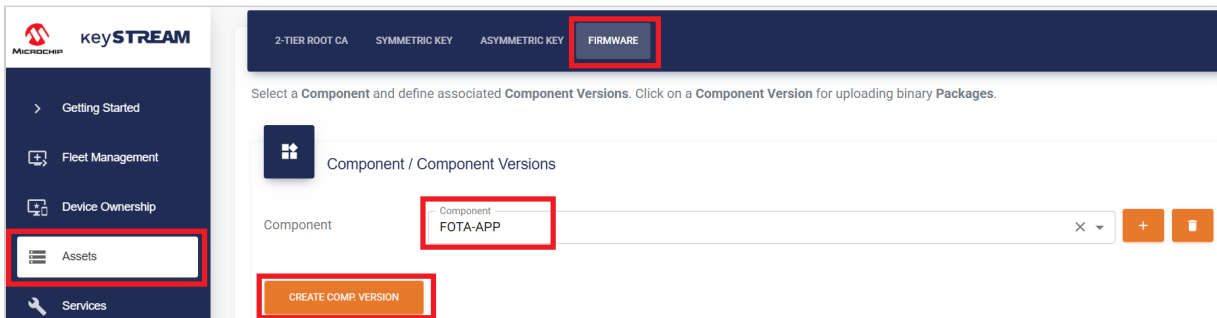


Figure-69

- If you don't see any components listed in the dropdown. Click **CREATE COMPONENT** and Name your component as "FOTA-APP". This is fixed in the Usecase.

 The screenshot shows the 'Create Component' form. It has a dark blue header with a plus icon and the text 'Create Component'. Below this are two input fields: 'Component Name' and 'Component Type'. The 'Component Type' field has 'BINARY' selected. At the bottom right are two buttons: 'CANCEL' and 'COMMIT'.


Figure-70

- Press **CREATE COMP. VERSION** and type "1.2.1" and then press "Commit"

 The screenshot shows the 'Create Component Version' form. It has a dark blue header with a plus icon and the text 'Create Component Version'. Below this is a single input field labeled 'Component Version' with the text '1.2.1' entered. At the bottom right are two buttons: 'CANCEL' and 'COMMIT'.

Figure-71

- Select the paper clip under "Packages" column for "1.2.1" to see a window appear to add your package/firmware. Click on "ADD PACKAGE" button



Edit Target Component Version 1.2.1


Component

Creation Date

Package Name	Source Comp. Version	Action
No records available.		

Figure-72

- Upload the updated firmware image, FOTA-APP_1.2.1.bin. Leave **Source Component Version** field blank, unless you want the firmware update to occur only for specific versions. This field is used to set conditional updates based on existing firmware versions. Once completed, click **COMMIT**.



Upload Package

Conditional Dependencies:

The Package you upload represents the **Target Component Version**, which will replace the existing **Source Component Version** on devices in the field.

- If you specify a **Source Component Version** below, the Package will be deployed only to devices currently running this specified **Source Component Version**.
- You can define multiple Packages, each targeting a specific **Source Component Version**. This allows precise control over updates based on the current software version of each device.
- If the **Source Component Version** field is left empty, the Package will be deployed to all devices *except* those explicitly targeted by other Packages. This functions as the **default Package**.

File * No file chosen

Source Component Version (for conditional dependency)

* is required

Figure-73

- To push the component version you just created go to Fleet Management-> Click on your Fleet

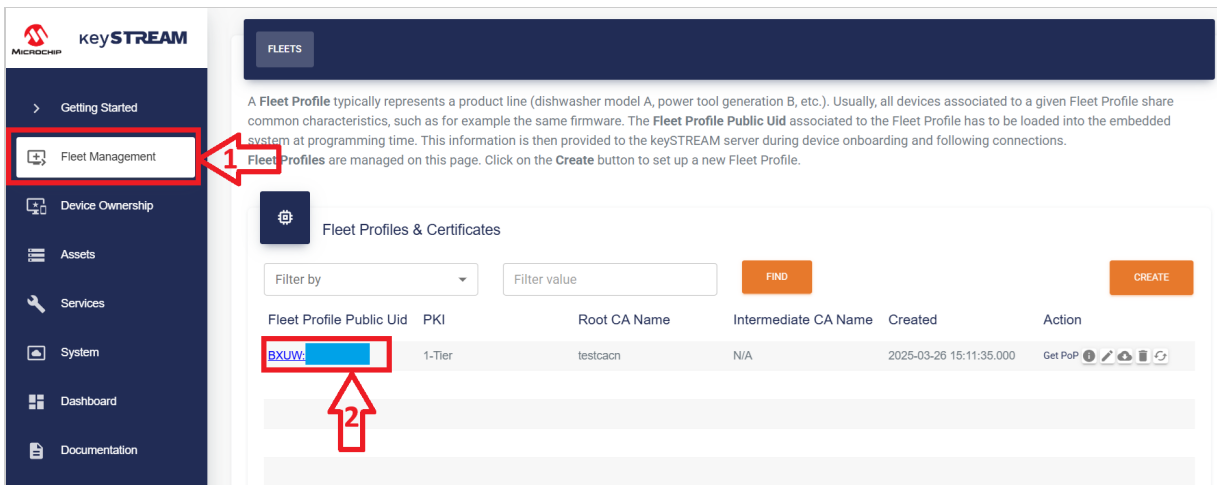


Figure-74

- See if the component you created is Associated to your fleet profile, if not associate it by clicking the associate button.

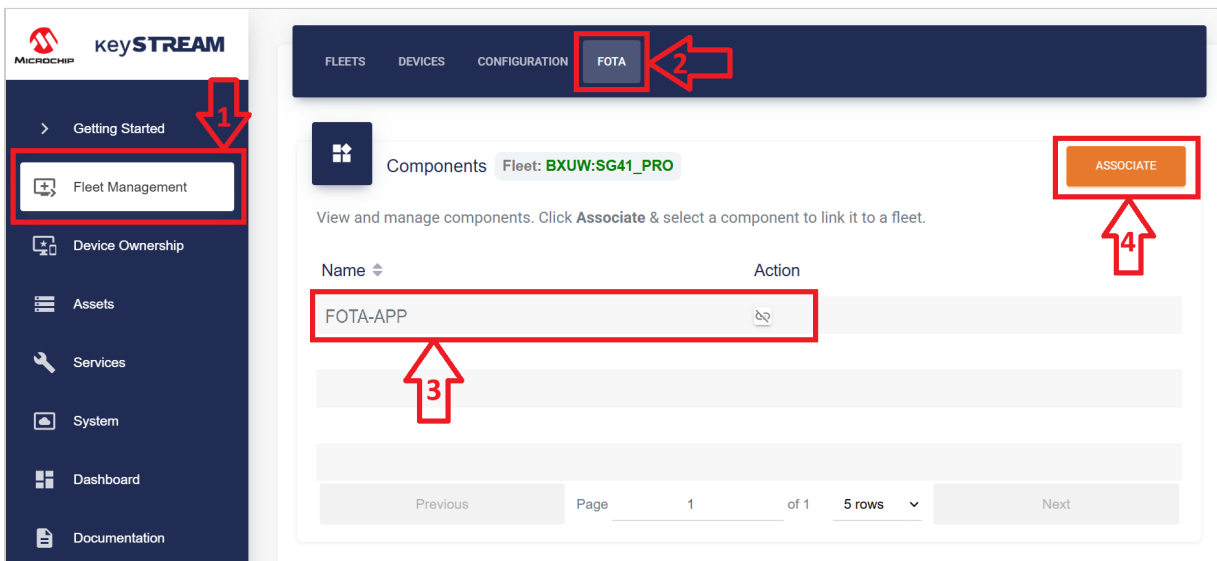


Figure-75

- Create the campaign by clicking **Create Campaign** and click commit once done.

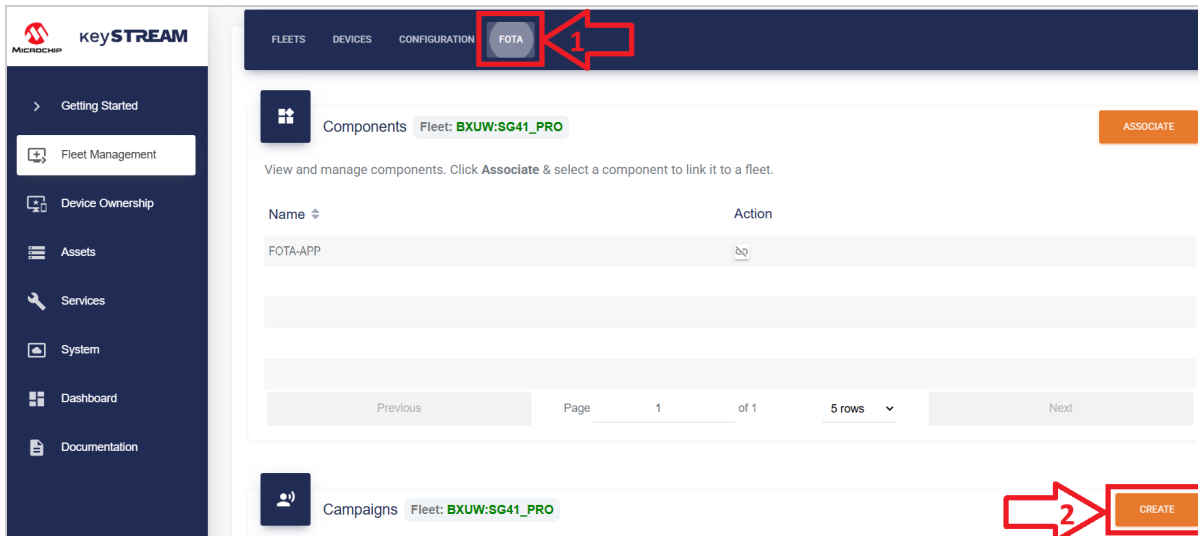


Figure-76

Create Campaign

Name *

Fleet Profile & Device Group * **Should be User's Fleet Profile ID**

Component *

Component Version *

Percentage

State *

Metadata (base64-encoded binary buffer)

* is required

Figure-77

- The percentage represent the percentage of devices of a given fleet you want to push an update for. It's generally a good practice to update at first a small percentage to ensure there is no incident and grow the percentage as the confidence level rises. For this Usecase, it can be left blank.
- Once you have pressed commit, the firmware over-the-air update is initiated by keySTREAM.

Observing the result of the FOTA by keySTREAM in TeraTerm

- We will now observe the result of the firmware over-the-air update by keySTREAM in TeraTerm.

- Observe in the bootloader window it will show our FOTA-APP component and the version we are downloading, 1.2.1:

```
[INFO] KTA msg generated successfully.
```

```
[INFO] Sending [69] bytes to KS
```

```
3021691088846ba65e550017000801c9d61b0100302f879ffbb348c5dd5bd36b
```

```
23870520d0cdfc0b93ee6c9832001ace54094af4f647a8993369131a0b4148d
```

```
2201a4h160
```

Figure-78

- You will observe it erase other bank on the PIC32SG41 and then it will download this new version to that bank and validate it before executing it.
- On the application TeraTerm output, you will observe it is no longer outputting light sensor readings, but rather the temperature sensor readings.
- You may also verify your FOTA was successfully deployed by navigating to the campaign window in the FOTA tab and then after clicking on your campaign, observing the "updated" field saying "1"

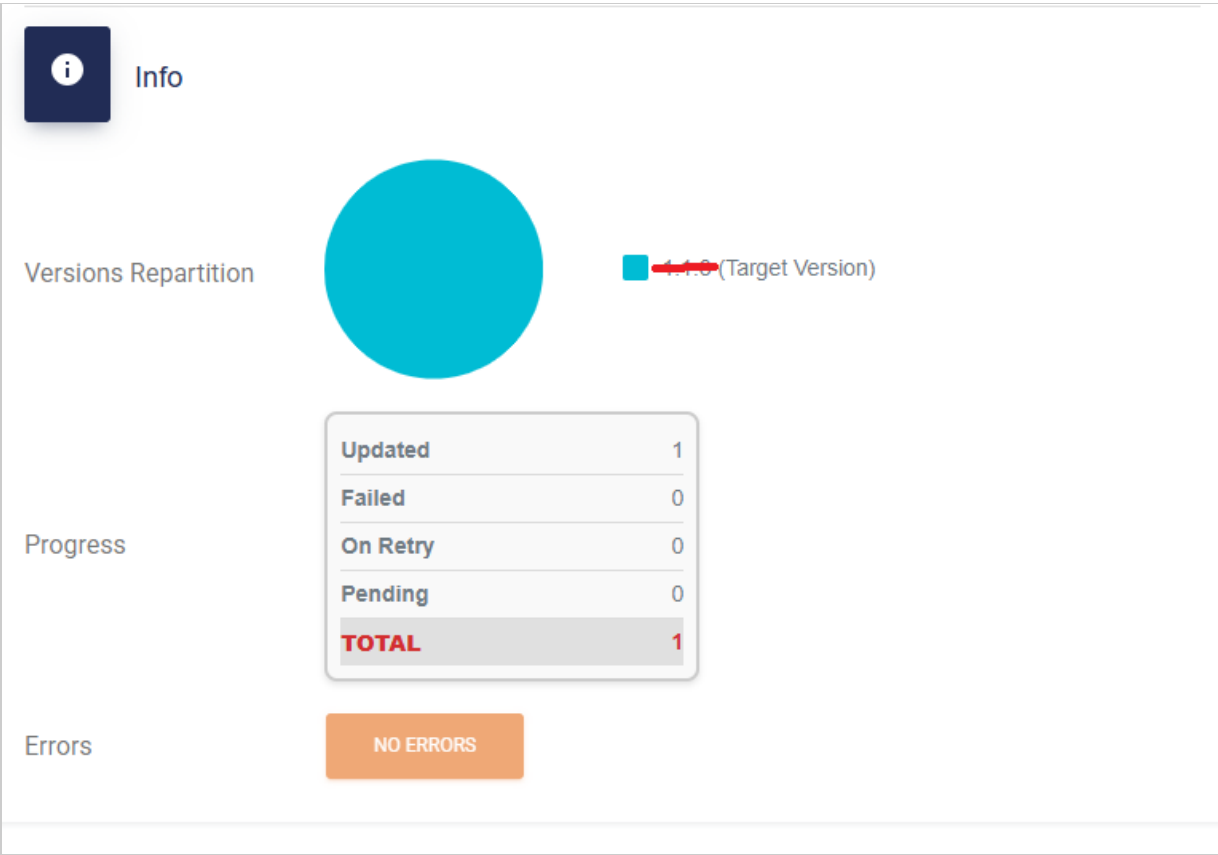


Figure-79

- This completes the FoTA Usecase steps.

Conclusion

The outlined use case demonstrates the keySTREAM Firmware Over-The-Air Update Usecase with ATECC608-TMNGTLS. This comprehensive guide covers the setup of the CryptoAuth Pro Trust Platform Development Kit, the provisioning of an ATECC608-TMNGTLS device, and the generation of necessary resources. It concludes with the steps to build and program the firmware, ultimately verifying the successful implementation of keySTREAM Firmware Over-The-Air Update through the ATECC608-TMNGTLS secure element.

Microchip Information

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** - Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** - Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** - Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.

- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge,

ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, TimeCesium, TimeHub, TimePictra, TimeProvider, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, EyeOpen, GridTime, IdealBridge, IGaT, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, MarginLink, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mSiC, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICKit, PICtail, Power MOS IV, Power MOS 7, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, Trusted Time, TSHARC, Turing, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.
© 2024, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved.
ISBN: 978-1-6683-0382-5

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.