EASY INTEGRATION FOR FIRMWARE UPDATES SOLUTION BY RENESAS MCUS

~ TO REDUCE CUSTOMERS' CYBERSECURITY RISK ~

2026 APR REV.5.00

EMBEDDED PROCESSOR & CONTROLLER SOLUTION MARKETING DEPT
EMBEDDED PROCESSING MARKETING DIVISION
EMBEDDED PROCESSING PRODUCT GROUP
(EP/EPMD/EPMSM)

RENESAS ELECTRONICS CORPORATION

CONTENTS

Market Needs

“Firmware Update” is critical for rapidly growing IoT devices and applications

Vulnerabilities for products

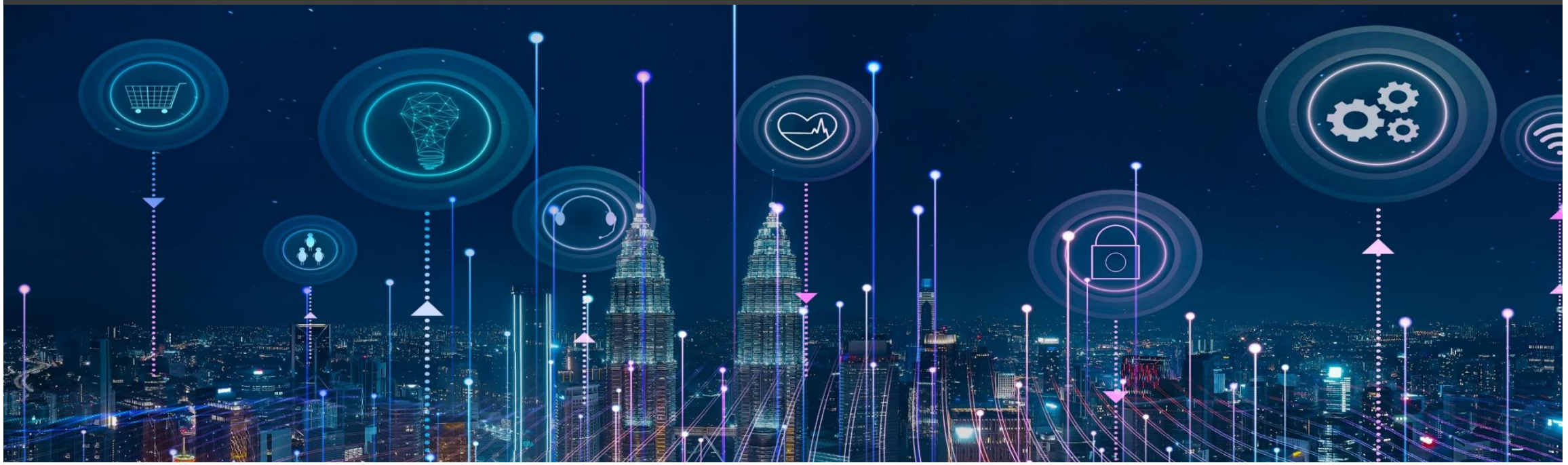
Common Technology for various application
(IA, FA, HA, BA, HC, Power Equipment, Social Infra…etc.)

Continuous Service

Maintain latest firmware updates OTA (Over-the-Air) via Cloud or Local network path

Cybersecurity Regulations

“Security Regulations and Legislation Requirements” such as CRA, U.S. Cyber Trust Mark, Other IoT Guideline



NECESSITY OF FIRMWARE UPDATE FEATURE IN ALL-DIGITAL PRODUCTS

As cybersecurity regulations tighten, **implementing Firmware Update capabilities** becomes a mandatory requirement for seamless lifecycle management of IoT and embedded products from Launch to End-of-Life.

Market Needs

- **Comply with Cybersecurity Regulatory Compliance**
  Urgent need to meet emerging global cybersecurity regulations such as CRA, U.S. Cyber Trust Mark and others.

- **Maintain Trust and Brand Value**
  Long-life IoT/embedded digital products require ongoing updates to ensure reliability and address vulnerabilities. Sustained updates preserve customer trust and brand value.

- **Enhancing the Update Foundation for Long-Term Use**
  As lifecycles extension,
  Products without update capabilities can no longer stand out.
  Cybersecurity readiness is now one of core technology to competitive advantage.

Role of Firmware Update Integration

- **Ensuring Safe Firmware**
  Signature verification ensures firmware **integrity** during updates and prevents tampering.

- **Secure Boot & Firmware Updates**
  Secure boot at power-on and post-launch firmware updates for new features and vulnerability fixes minimize cyberattack risks and maintain product **availability**.

- **Key Management & Encryption**
  Combining key management with encryption preserves the **confidentiality** of critical data.

FIRMWARE UPDATE* INFRASTRUCTURE EXAMPLES

*: Include **Secure Boot**

Some use cases of firmware update implementations as common applications on IoT devices and digital products

: New Firmware Image

Application e.g.

Social infrastructure

Industry Application (FA/IA/OA)

HA (Home Appliances/Healthcare)

FW Update Options for IoT Products

IT side

Examples of IoT device communication

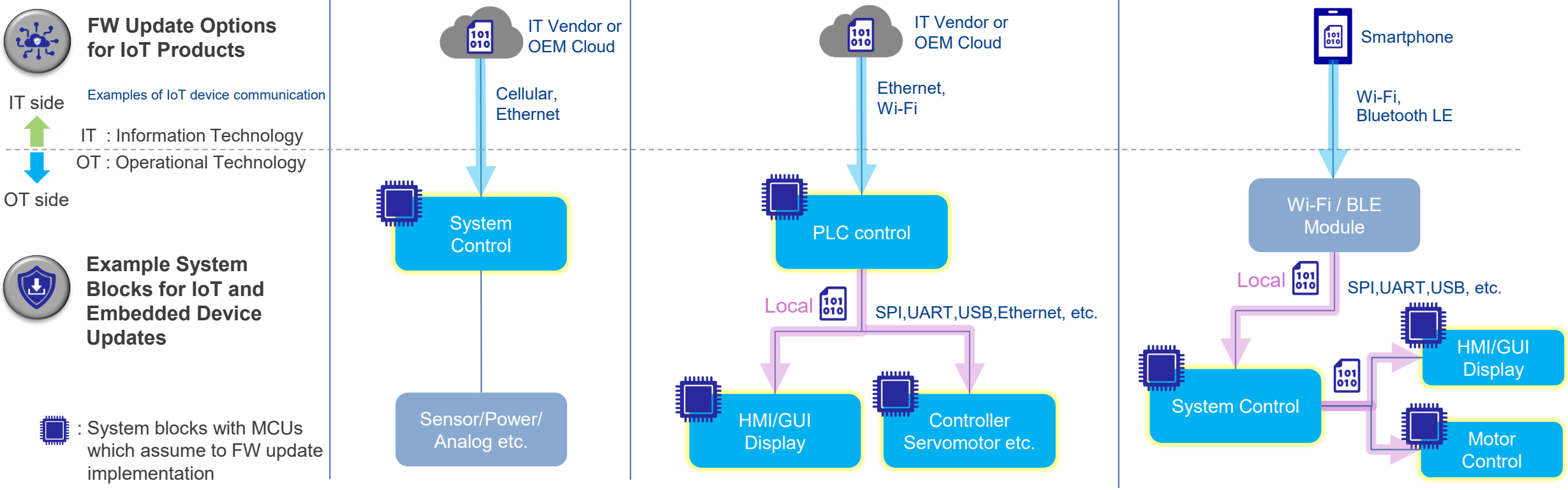
IT : Information Technology

OT : Operational Technology

OT side

Example System Blocks for IoT and Embedded Device Updates

: System blocks with MCUs which assume to FW update implementation



FIRMWARE UPDATE SOLUTIONS BY UTILIZING RENESAS MCU

FW UP : Firmware Update
OSS : Open-Source Software
SKMT : Security Key Management Tool
TSIP : Trusted Secure IP
RSIP : Renesas Secure IP
SCE : Secure Crypto Engine

Provide optimal FW Update methods by leveraging Spec and Security Engine on Renesas MCU family

“ FW UP Module ” Solution		Contents	“ MCUboot ” Solution		
RX Family	**RL78** Family	Target MCU Family	**RA** Family without Security Engine	**RA** Family with Security Engine	**RX** Family with Security Engine
FW UP Module		Driver name	**MCUboot Module**		
Software Cryptographic Library		Signature Verification by…	Software Cryptographic Library	Security Engine（by TSIP / RSIP / SCE）	
ECDSA P-256, SHA256		Signature Verification Method	ECDSA P-256/P-384, RSA 2048/3072, ML-DSA-44/65/87	ECDSA P-256/P-384, RSA 2048/3072	ECDSA P-256, RSA 2048
Plaintext format		Store method for a “Public Key” for Signature Verification	**Plaintext** format	**Wrap** format* or **Plaintext** format	**Wrap** format*
Linear Mode - **Partial Update** Method Linear Mode - **Full Update** Method Dual Mode - **Dual-bank** Method		Selectable Update method for Flash ROM type	Linear Mode - **Swap** Method Linear Mode - **Overwrite** Method Linear or Dual Mode - **DirectXIP** Method		
Refer to FW UP Soution Section		Details	Refer to MCUboot Solution Section		

* : Create a public key for signature verification wrapped in SKMT and store it using a sample for key installation

Note : Implementation details (e.g., signature verification and update methods) vary depending on the presence of an MCU or security engine.
For more information, please refer to the respective solution sections and Application notes.

RX & RL78 Family
“Firmware Update（FW UP）Module ”
Solution

FIRMWARE UPDATE ROLES AND IMPLEMENTATION STEPS

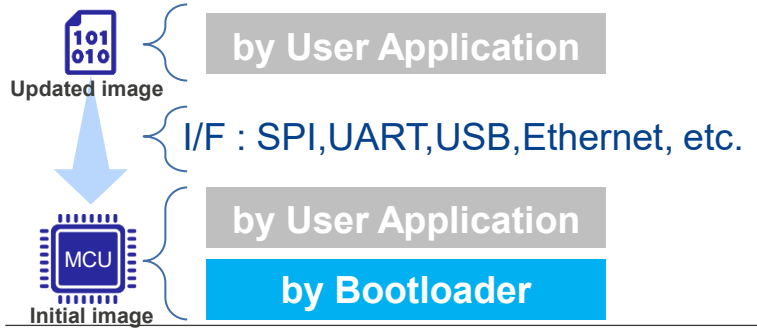
Key pairs are prepared on the customer side,
while Renesas supports secure signing and firmware updates through tools and drivers.

Firmware Update

Required "Steps" and "Implementations" for realizing Firmware Update by Renesas Solution

Steps : Firmware Update

1. Import the update image to the MCU via a communication interface
2. Program the update image to the on-chip flash memory of the MCU
3. Validate the update image
4. Activate to the update image



Implementations : Firmware Update

Color coding of blocks: The process prepared by the customer / Processes provided by Renesas

1. Create a Firmware Image

- ✓ Generate keys for Code Signing
- ✓ Integrate both the Bootloader & User application
- ✓ Generate the Code signature verification data

2. Execute Firmware Update

- ✓ Communication control to import the update image to the MCU
- ✓ Program the update/new image to internal flash memory on the MCU
- ✓ Validity verification of the update image by the Code signature verification
- ✓ Activate to the update/new image

The code signature verification algorithm is based on Elliptic Curve Cryptography (ECDSA).It supports ECDSA NIST P-256 and SHA-256 (hash only).

Key Generator Tool
(Sample keys are generated using the OpenSSL (OSS).)

Renesas Provide

Image Generator for Updating FW
Renesas Image Generator

+

FW UP Module

+

Bootloader

+

User Application

FEATURES OF “FW UP MODULE”

Firmware Update

Your Firmware Update function can be easily implemented by using the “FW UP module” !

Renesas Provide

Image Generator for Updating FW
Renesas Image Generator

+

Bootloader
Sample

+

FW UP Module

+

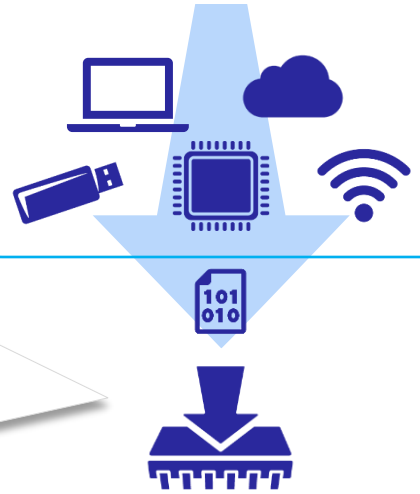
User Application

Preparation: Create the program's mot file and key pair

Generates signed Update FW Image* by the Renesas Image Generator

* OpenSSL generates Key Pairs and a Code Signature

Input the new Update FW Image via various Methods



Easy Firmware update in various interfaces

(UART / SPI / USB / Ethernet, etc., via serial communication, etc.)

A new firmware image can be acquired by **various communication interfaces** in user applications to meet the customer requirements.

“3” selectable modes of FW Update Methods by Flash ROM type on MCUs

1. Dual Mode : **Dual-Bank Method**
2. Linear Mode : **Partial Update Method**
   - ➢ Easy recovery even if FW updating fails
3. Linear Mode : **Full Update Method**
   - ➢ Preferred FW update method for a small memory footprint MCU
   - ➢ Supports the method of using an external flash as a buffer

Secure Firmware Update by validating Code signature

Supports firmware validation by **ECDSA NIST P-256 and SHA256** as Secure Boot feature.

SAMPLE SYSTEM CONFIGURATION USING THE FW UP MODULE

The FW UP Module is ideal for realizing your firmware update over various “ **Primary Devices** ”

Compatible with various use cases

: New Firmware Image

Stand-alone
(Products that run on a single device)

Home appliances

Measuring instruments

Air conditioners, etc.
such as “ **Paired**” systems

Outdoor unit (ODU)

Indoor unit (IDU)

Industrial system
(System consisting of control devices and connected multiple secondary devices)

Control device
(primary device)

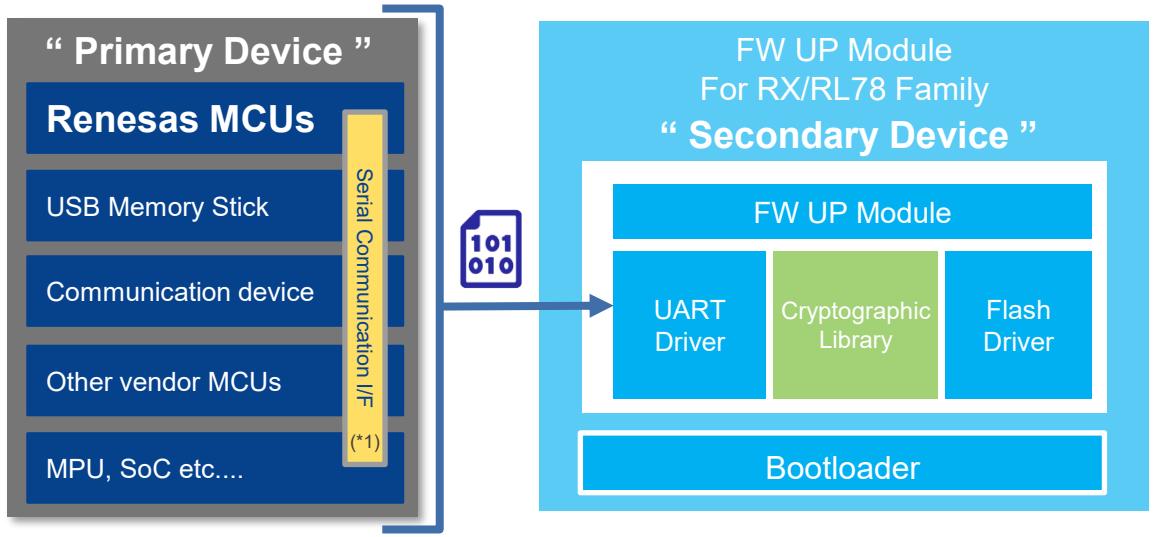
Secondary Device

Secondary Device

Secondary Device

MCUs supported by FW UP Module : **RX Family** - All RX Family,
RL78 Family - RL78/G22, RL78/G23, RL78/G24, RL78/L23

Get an updated FW via serial communication



*1 : Firmware Updating Communications Module : (RX, RL78)

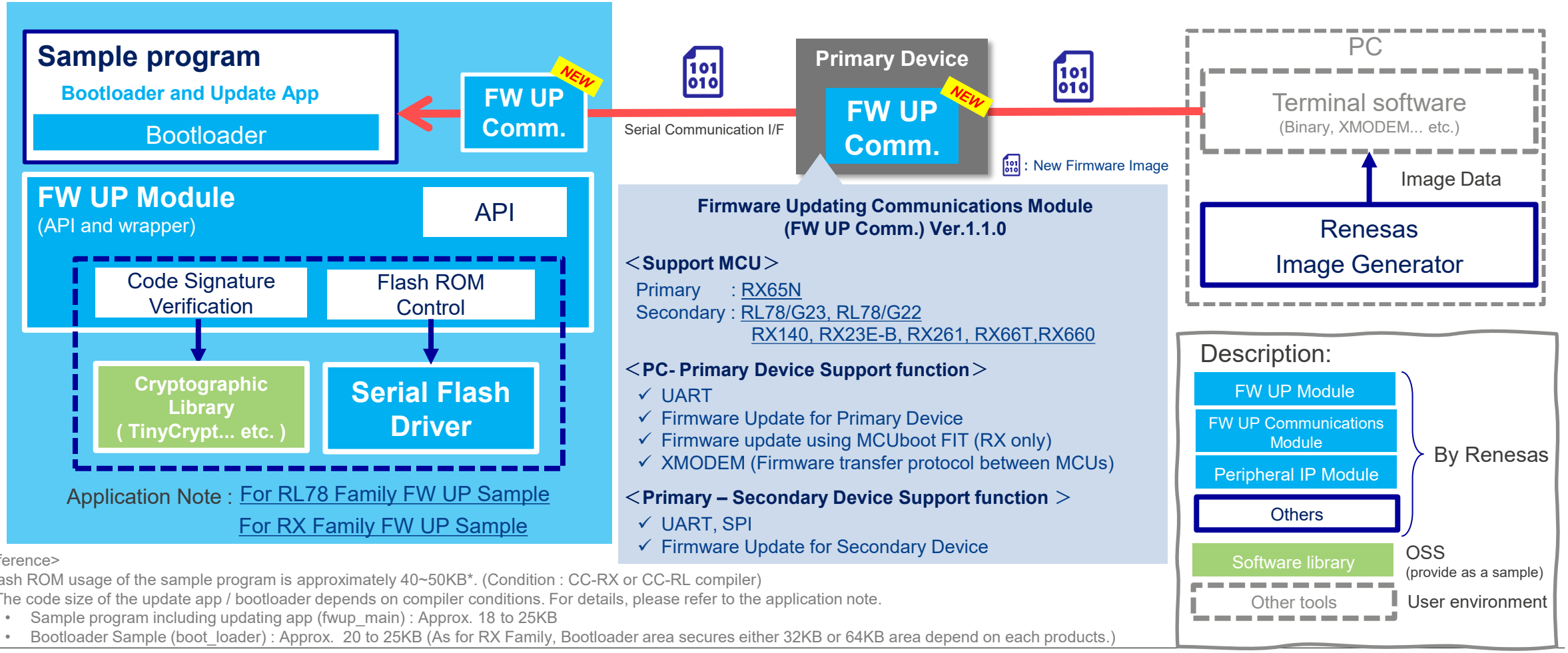
Application Notes

FW UP Module APN (Link) for RX Family

FW UP Module APN (Link) for RL78 Family

SYSTEM CONFIGURATION EXAMPLE FOR “SECONDARY DEVICE” UPDATE BY THE “FW UP MODULE”

➢ “ Secondary Device ” firmware update



<Reference>
Flash ROM usage of the sample program is approximately 40~50KB*. (Condition : CC-RX or CC-RL compiler)
* The code size of the update app / bootloader depends on compiler conditions. For details, please refer to the application note.
- Sample program including updating app (fwup_main) : Approx. 18 to 25KB
- Bootloader Sample (boot_loader) : Approx. 20 to 25KB (As for RX Family, Bootloader area secures either 32KB or 64KB area depend on each products.)

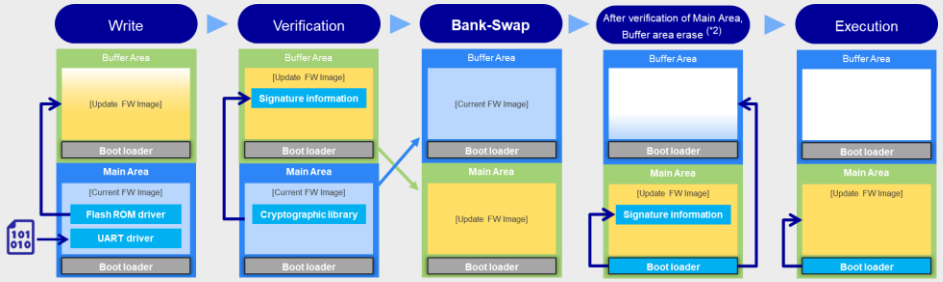
SELECTABLE UPDATE METHOD ACCORDING TO YOUR MCU FLASH ROM PRODUCT

"3" selectable modes of FW Update Methods by Flash ROM type on MCUs

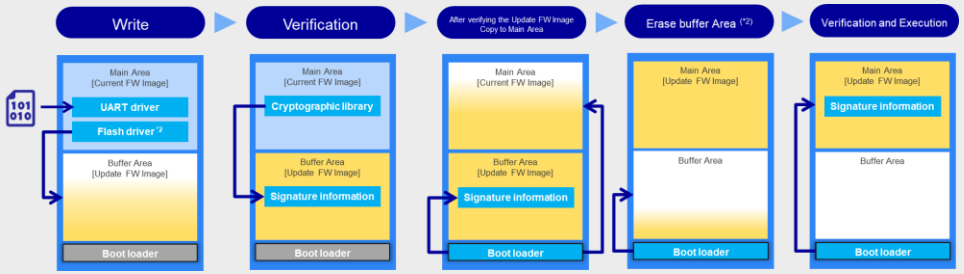
~ SUPPORT FOR DEVELOPMENT WITH SAMPLE PROGRAMS FOR EACH METHOD ~

<Sample program>
RX : #R01AN6850 / Sample Zip
RL78 : #R01AN6374 / Sample Zip

1. Dual Mode: Dual-Bank* update operation

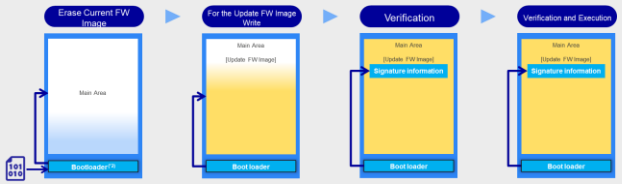


2. Linear Mode: Partial Update Method* update operation

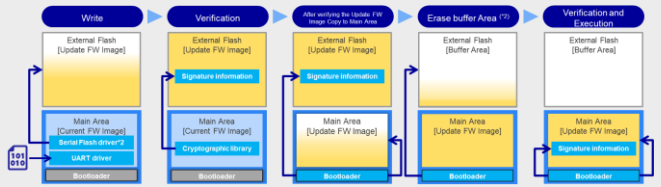


3. Linear Mode: Full Update Method* update operation

without buffer plane



with buffer plane (External Flash)



Modes and Device Correspondence Table

Flash Product	4MB	2MB	1.5MB	1MB	768KB	512KB	384KB	256KB	128KB	96KB	64KB
RX130	-	-	-	-	-	L	L	L	L	-	-
RX140	-	-	-	-	-	-	-	L	L	-	-
RX231/230	-	-	-	-	-	L	L	L	L	-	-
RX23E-A	-	-	-	-	-	-	-	L	L	-	-
RX23E-B	-	-	-	-	-	-	-	L	L	-	-
RX24T	-	-	-	-	-	L	L	L	L	-	-
RX261	-	-	-	-	-	L	L	L	-	-	-
RX26T	-	-	-	-	-	DB/L	-	L	L	-	-
RX65N/651	-	DB/L	DB/L	L	L	L	-	-	-	-	-
RX66N	DB/L	L	-	-	-	-	-	-	-	-	-
RX66T	-	-	-	L	-	L	-	L	-	-	-
RX660	-	-	-	L	-	L	-	-	-	-	-
RX671	-	DB/L	DB/L	L	-	-	-	-	-	-	-
RX72M	DB/L	L	-	-	-	-	-	-	-	-	-
RX72N	DB/L	L	-	-	-	-	-	-	-	-	-
RX72T	-	-	-	L	-	L	-	-	-	-	-
RL78/G22	-	-	-	-	-	-	-	-	-	-	L
RL78/G23	-	-	-	-	L	L	L	L	L	L	-
RL78/G24	-	-	-	-	-	-	-	-	L	-	L
RL78/L23	-	-	-	-	-	DB/L	-	DB/L	L	L	L

DB : Products supporting Dual Mode (1. Dual Bank Method)
L : Products supporting Linear mode (2. Partial Update / 3. Full Update Method)
- : Not supported
red text: Products supporting the sample program(Sample provided in zip)

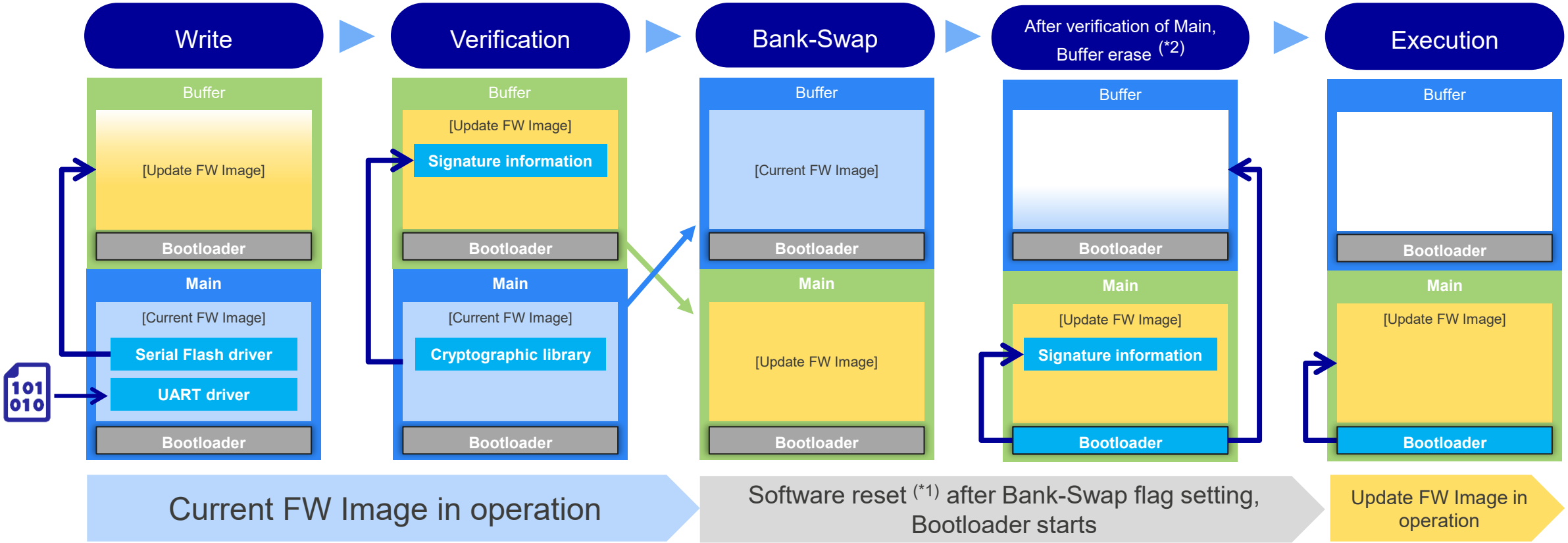
Note: RX Family demo project for linear mode full update method (with buffer plane using external flash) is not provided for dual-bank compatible products. However, implementation is possible. If needed, please consider using a demo project for a product with a similar memory layout as a reference.

* : The updating Method name is partially different from the application note for RL78.
"1." means "Dual-Bank(2 bank) method",
"2." means "Partial update method (buffer plane is internal flash)", and
"3." means "Full update method (without buffer plane)" "Full update method (buffer plane is external flash)"

1. DUAL MODE
DUAL-BANK METHOD: NORMAL OPERATION

For use case that doesn‘t want to stop user applications, Dual-bank supported MCUs are recommended !

For **Dual-Bank** supported MCUs, Update FW Image can be written while the current program on the main is executing!
Address placement management of programs is also *not necessary* to utilize the **Bank-Swap** function.



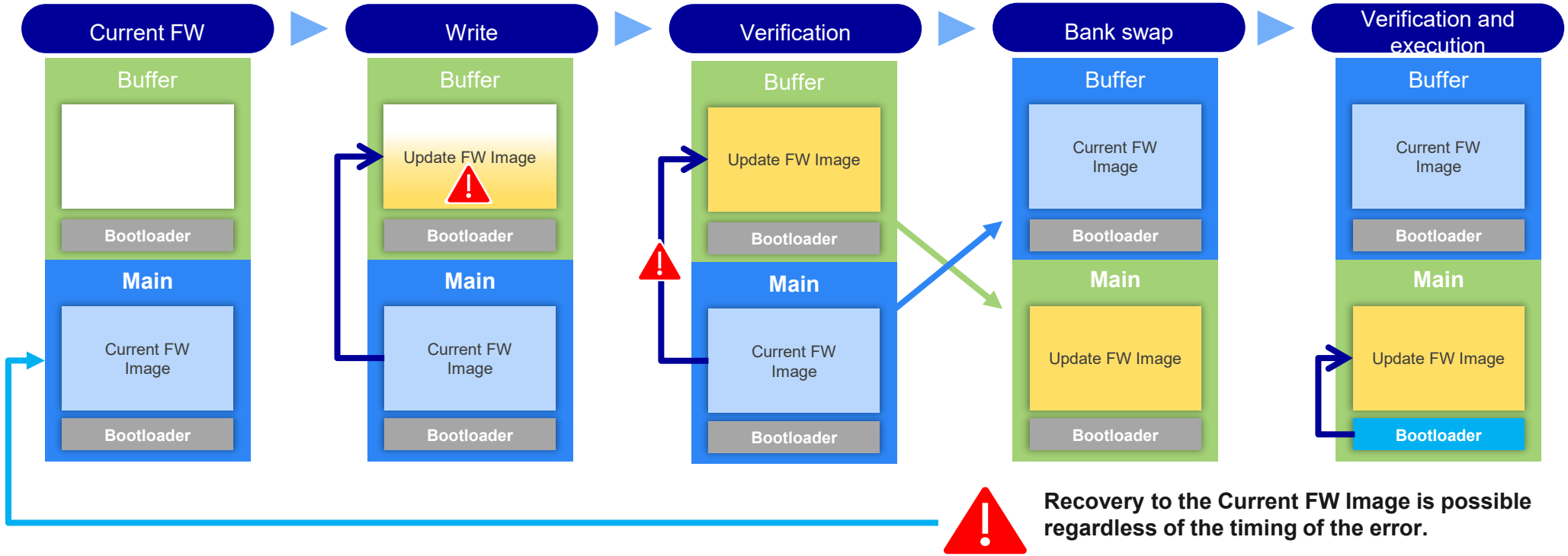
*1 : For initialization by a software reset, please see the "Reset Chapter" in the hardware manual of each MCU.
*2 : The demonstration program of the FW UP Module does not erase the Buffer. If it is necessary to erase the current(previous) Image before updating, users need to add the Image erase process of located the buffer side to prevent rollback.

1. DUAL MODE
DUAL-BANK METHOD: OPERATION WITH ERROR OCCURS

If the firmware update process fails due to a power-cut/signature verification failure as abnormal process, Enables to boot the Current FW Image and redo the firmware update,again.

Buffer	Disabled	Disabled	Enable	Enable
Main	Enable	Enable	Enable	Enable

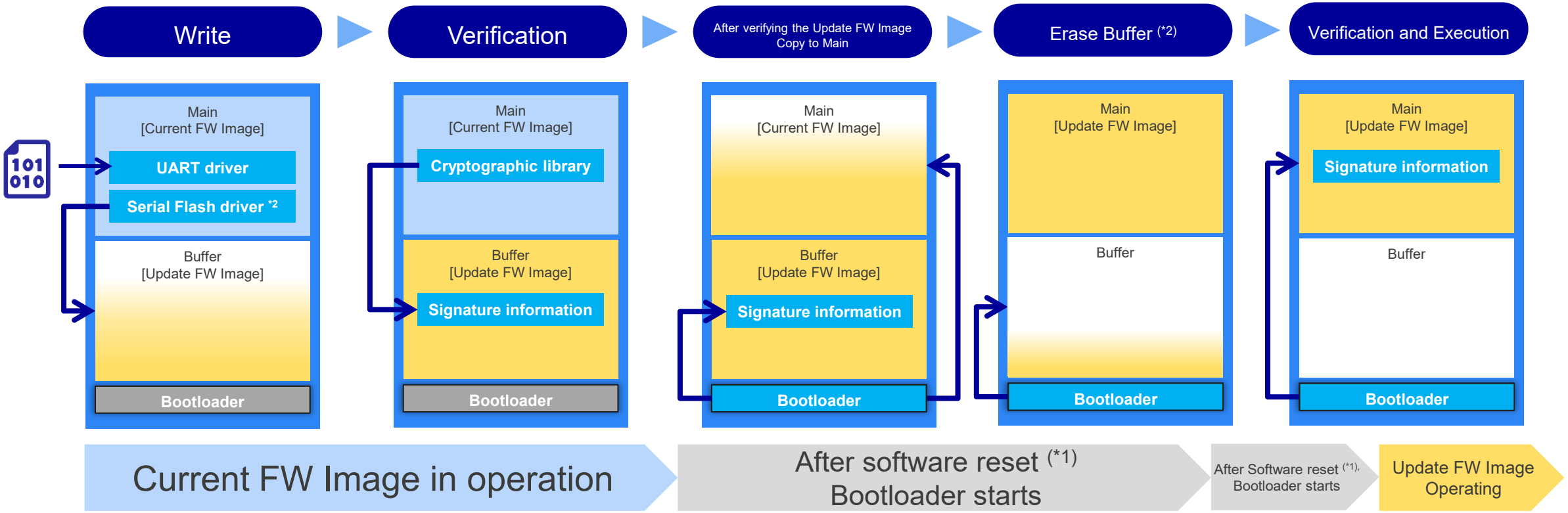
Green text: Startable state / **Black text:** Unbootable state



Recovery to the Current FW Image is possible regardless of the timing of the error.

2. LINEAR MODE
PARTIAL UPDATE METHOD: NORMAL OPERATION

Even if MCU does not support Dual-Bank, a firmware can be Update with the previous FW Image retained by using the Partial Update Method.



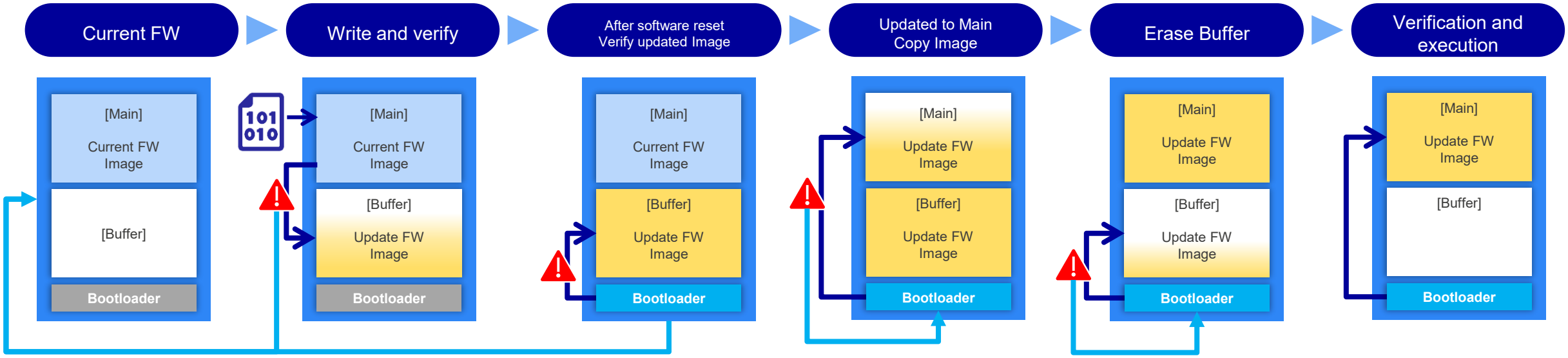
*1 : For initialization by a software reset, please see the "Reset Chapter" in the hardware manual of each MCU.
*2 : Flash write operations by the Serial Frash driver run in RAM. See the Serial NOR Flash Memory Control Module application notes for each family (RX, RL78) for details.

2. LINEAR MODE
PARTIAL UPDATE METHOD (WITH BUFFER PLANE) : OPERATION WHEN AN ERROR OCCURS

If the firmware update fails due to a power-off or signature verification failure,
You can boot with the Image before the update and start the firmware update again.

Main	**Enable**	**Enable**	**Disabled**	**Enable**	**Enable**
Buffer	**Disabled**	**Disabled**	**Enable**	**Disabled**	**Disabled**

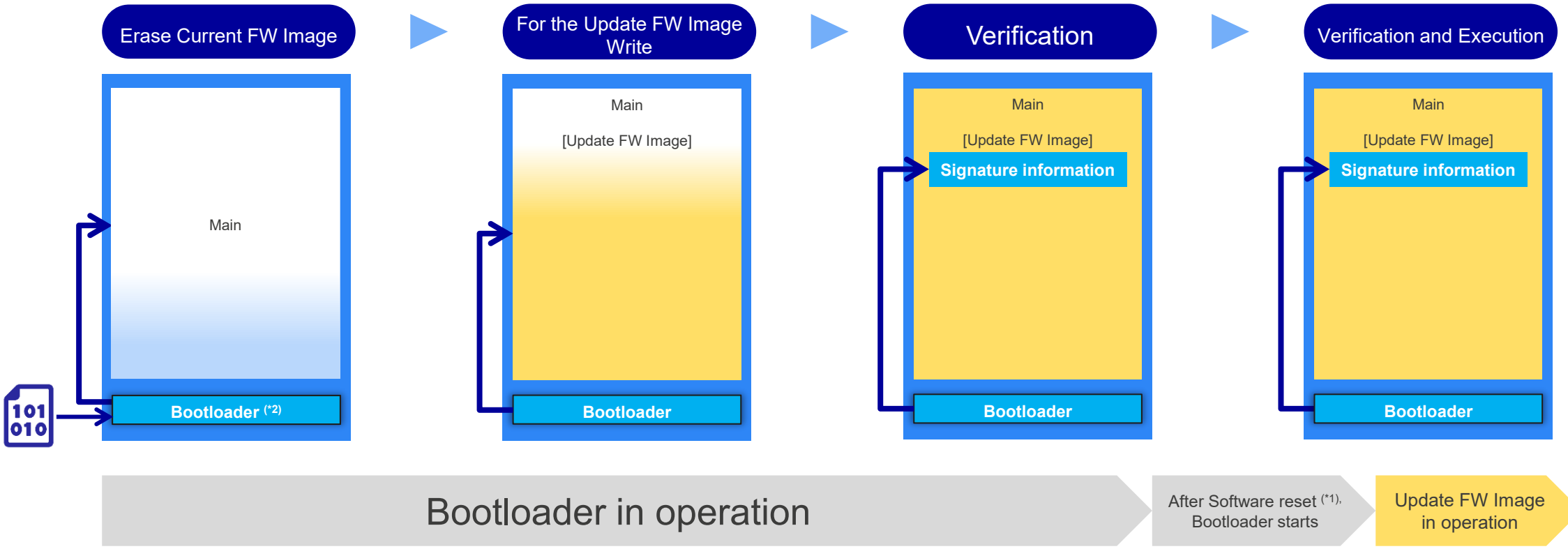
Green text: Startable state / **Black text:** Unbootable state



**There is always a valid Image on either side,
And recovery is possible in case of an error.**

3. LINEAR MODE
FULL UPDATE METHOD (WITHOUT BUFFER PLANE): NORMAL OPERATION

Firmware updates can also be implemented for small ROM footprinted products using the Full Update Method.



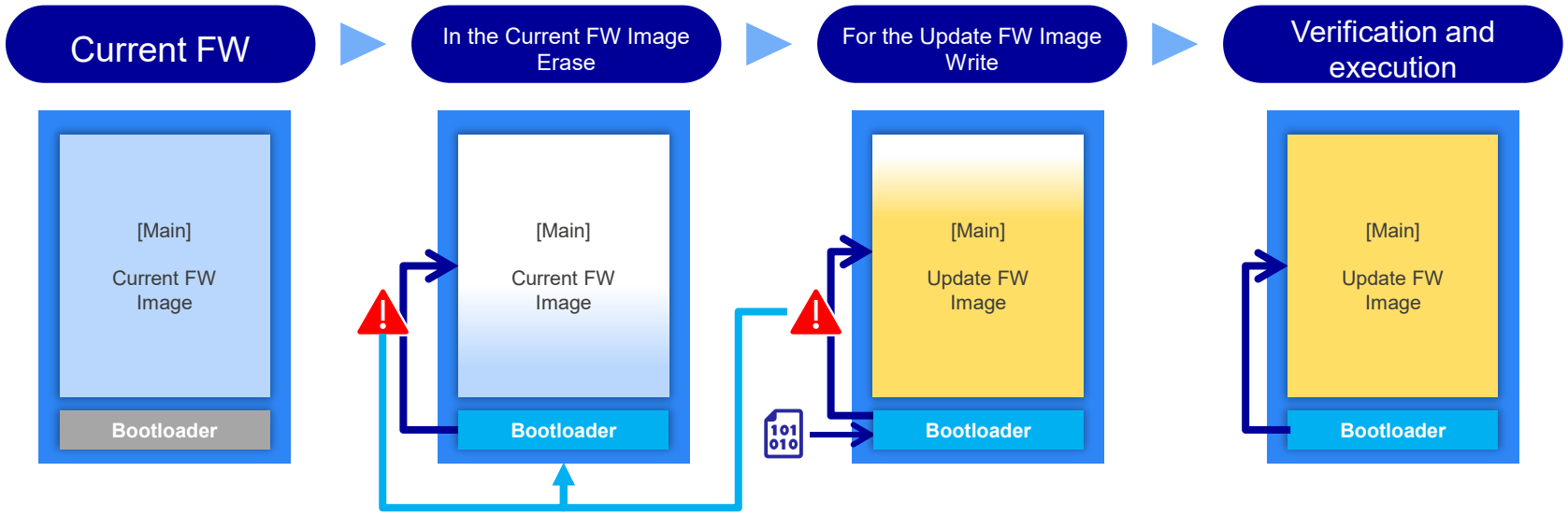
*1 : For initialization by a software reset, please see the "Reset Chapter" in the hardware manual of each MCU.
*2 : The demonstration program bootloader uses UART communication to obtain an updated Image. It is necessary to change it according to the communication Method customers want to use.
*3 : Flash write operations by the Serial Frash driver run in RAM. See the Serial NOR Flash Memory Control Module application notes for each family (RX, RL78) for details.

3. LINEAR MODE FULL UPDATE METHOD(WITHOUT BUFFER PLANE): OPERATION WHEN AN ERROR OCCURS

With the Full Update Method, if the firmware update fails, use the bootloader function to perform the firmware update again.

Main	Enable	Disabled	Disabled	Enable

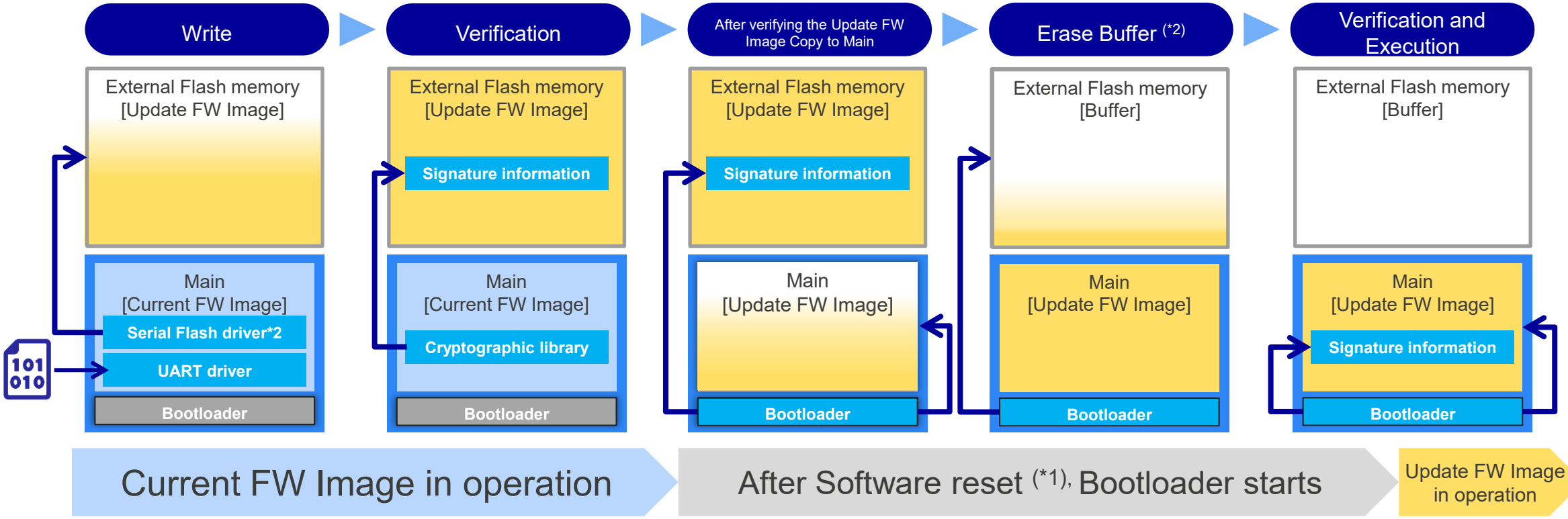
Green text: Startable state / **Black text:** Unbootable state



Repeat until the firmware update is successful.

3. LINEAR MODE
FULL UPDATE METHOD (WITH BUFFER PLANE): NORMAL OPERATION

Using an external flash memory enables the system to get the Update FW Image while maintaining the Current FW Image.



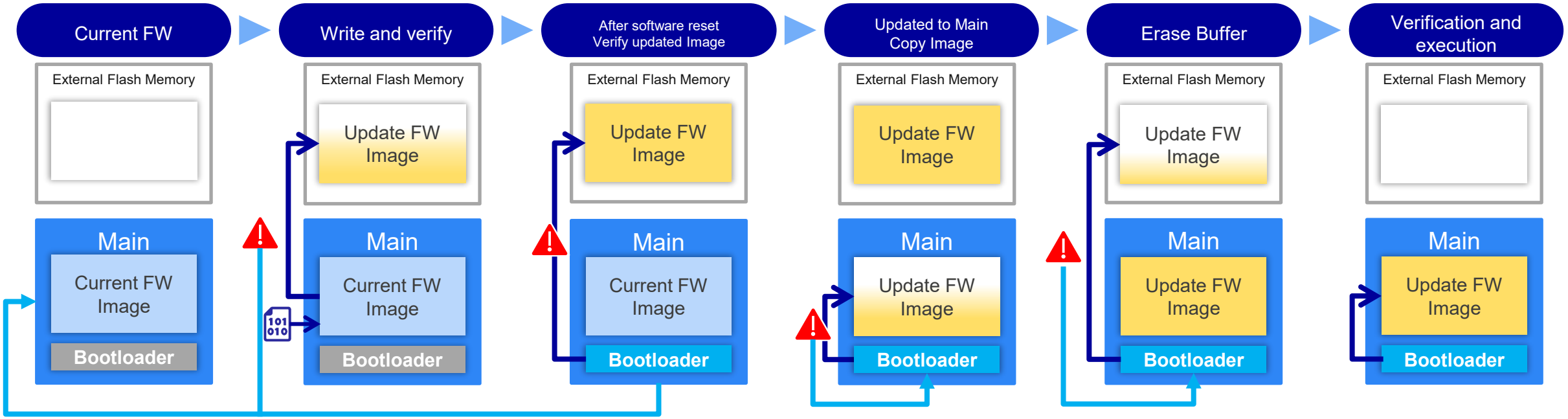
*1 : For initialization by a software reset, please see the "Reset Chapter" in the hardware manual of each MCU.
*2 : For details on the write process to External Flash, please refer to the application notes for the Serial NOR Flash Memory Control Module of each device family (RX, RL78).

3. LINEAR MODE FULL UPDATE METHOD(WITH BUFFER PLANE): OPERATION WHEN AN ERROR OCCURS

If the firmware update fails due to a power-off or signature verification failure,
You can boot with the Image before the update and start the firmware update again.

External Flash Memory (Buffer)	**Enable**	**Enable**	**Disabled**	**Enable**	**Enable**
Main	**Disabled**	**Disabled**	**Enable**	**Disabled**	**Disabled**

Green text: Startable state / **Black text:** Unbootable state



There is always a valid Image on either side,
And recovery is possible in case of an error.

Renesas Image Generator Capabilities: Signed Initial and Update Image Generation

Renesas Image Generator can be used for both "signing to firmware" and "binding with bootloader and user firmware "

1. Preparation

Signature Verification Method: **ECDSA NIST P-256**
The following data is generated on the customer side.

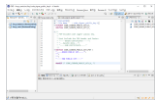
Generate Key and Signature Information

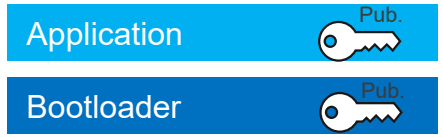
OpenSSL

Generating **Public Key** and **Private Key** Key Pairs.
The sample key pair provided by Renesas is generated using OpenSSL (OSS).

Create a firmware

e² studio





Including a public key for signature verification in the **Application** and **Bootloader** source code.

Generate mot Files

Generate each .mot file with e² studio

Application.mot
Bootloader.mot

Builde and Generate mot file

2. Execute in the Renesas Image Generator

By simply inputting the pre-prepared files, the initial image can be generated easily.

Create RSU※ Header Addresses

Renesas Image Generator creates header address information as RSU using a parameter file that includes address information for the target MCU device.

RSU Header-Address Information

※ RSU (Renesas Secure Update) is a proprietary Renesas data format that is smaller than MOT files and helps reduce communication traffic.

Grant RSU Header-Signing Information

Signing RSU Header Addresses Information and Applications with Private Key

RSU Header-Address Information

Application.mot

Provide code signing information

3. Complete of generating the initial Image

Renesas Image Generator **integrates RSU, the application, and the bootloader** to generate the initial image (.mot).

➢ **Initial Image(.mot)**

RSU Header-Signing Information (0x200 turning tool)
RSU Header-Address Information (0x100 turning tool)
Application Program data
Bootloader (Code flash data)

4. From this step onward, generate the update image (.rsu).

Can also be generated using **Renesas Image Generator.**
For detailed procedures and practical usage, refer to the application notes （RX, RL78）

RA & RX family:

“ MCUboot “ Solution

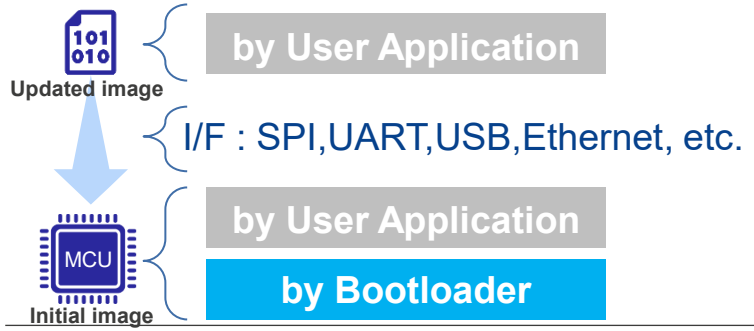
FIRMWARE UPDATE ROLES AND IMPLEMENTATION STEPS

Key pairs are prepared on the customer side,
while Renesas supports secure signing and firmware updates through tools and drivers.

Required "Steps" and "Implementations" for realizing Firmware Update by Renesas Solution

Steps : Firmware Update

1. Import the update image to the MCU via a communication interface
2. Program the update image to the on-chip flash memory of the MCU
3. Validate the update image
4. Activate to the update image



Implementations : Firmware Update

Color coding of blocks: The process prepared by the customer / Processes provided by Renesas

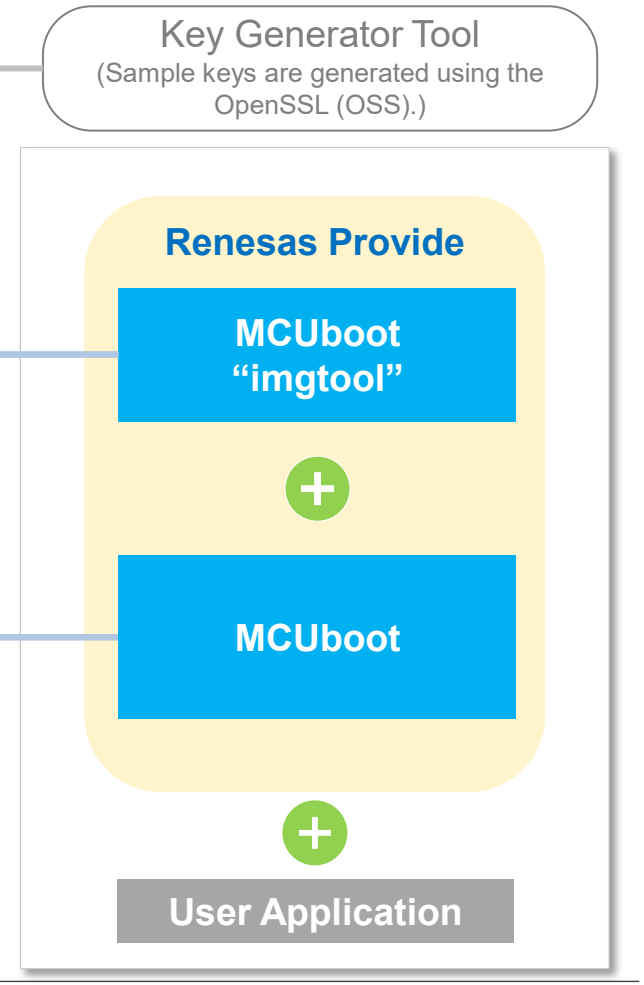
1. Create a Firmware Image

- ✓ Generate keys for Code signing
- ✓ Generate keys for image encryption
- ✓ Integrate both the bootloader & user application
- ✓ Encrypt the image, and generate the Code signature verification data

2. Execute Firmware Update

- ✓ Communication control to import the update/new image to the MCU
- ✓ Program the update/new image to internal flash memory on the MCU after decrypting the image
- ✓ Validity verification of the update image by the Code signature verification
- ✓ Activate to the update/new image

* The code signature verification algorithm varies depending on the MCUs.



“ MCUboot “ solution for realizing secure Firmware update by Renesas MCUs with Security Engine*

* : TSIP/RSIP/SCE

The Renesas Security Engine* can realize encryption performance & secure key management !

■ MCUboot Feature

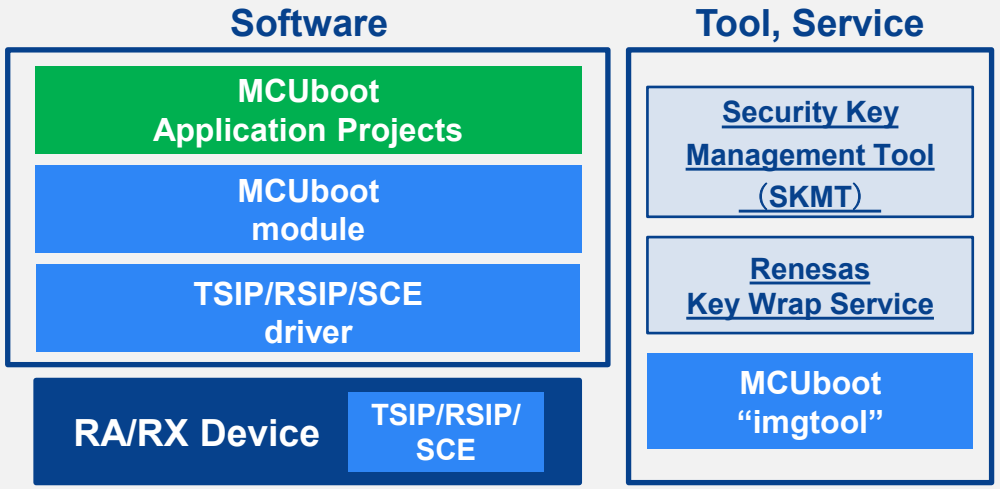
- ✓ **Secure bootloader** for 32-bit MCUs
- ✓ **Open-Source Software(OSS)** for the purpose of standardizing the bootloader and system flash infrastructure.
- ✓ Flexible design that can be integrated with a variety of hardware and operating systems
- ✓ This is a primary bootloader solution for embedded MCU systems

Reference：MCUboot | mcuboot
GitHub - mcu-tools/mcuboot: Secure boot for 32-bit Microcontrollers!

■ Features of the integration of TSIP/RSIP/SCE into MCUboot

- ✓ **Secure FW update with signature verification**
- ✓ **Improved encryption processing performance**
  Using security engine for firmware signature verification and decryption enables **faster processing, lower power consumption**, and **reduced load**.
- ✓ **Protection of Key data**
  Keys used for signature verification and image decryption are stored in a **wrap format**.
- ✓ **Secure key handling**
  Secure key injection is supported through tools such as **SKMT** and the **Renesas Key Wrap Service**.

Renesas Deliverables



TSIP : Trusted Secure IP, **RSIP** : Renesas Secure IP, **SCE** : Secure Crypto Engine

Supported Devices

RA Family：RA4, RA6 and RA8 with Security Engine
RX Family：RX261, RX65N, RX651, RX66N, RX671, RX72M, RX72N

FEATURES AND CONFIGURATION OF MCUBOOT SOLUTION FOR RA & RX FAMILY

Realize Secure Firmware Updates by the MCUboot Module integrated with Security Engine*1

*1 : TSIP/RSIP/SCE

Signature Verification: Security Engine (TSIP / RSIP / SCE)

Firmware Signature Verification at Startup

Verify the authenticity of the firmware when MCU starts up.

Firmware Signature Verification at Firmware update

Verify the authenticity of the firmware after receiving the entire update image *.

* In case of updated firmware image is encrypted, the verification process includes decryption.

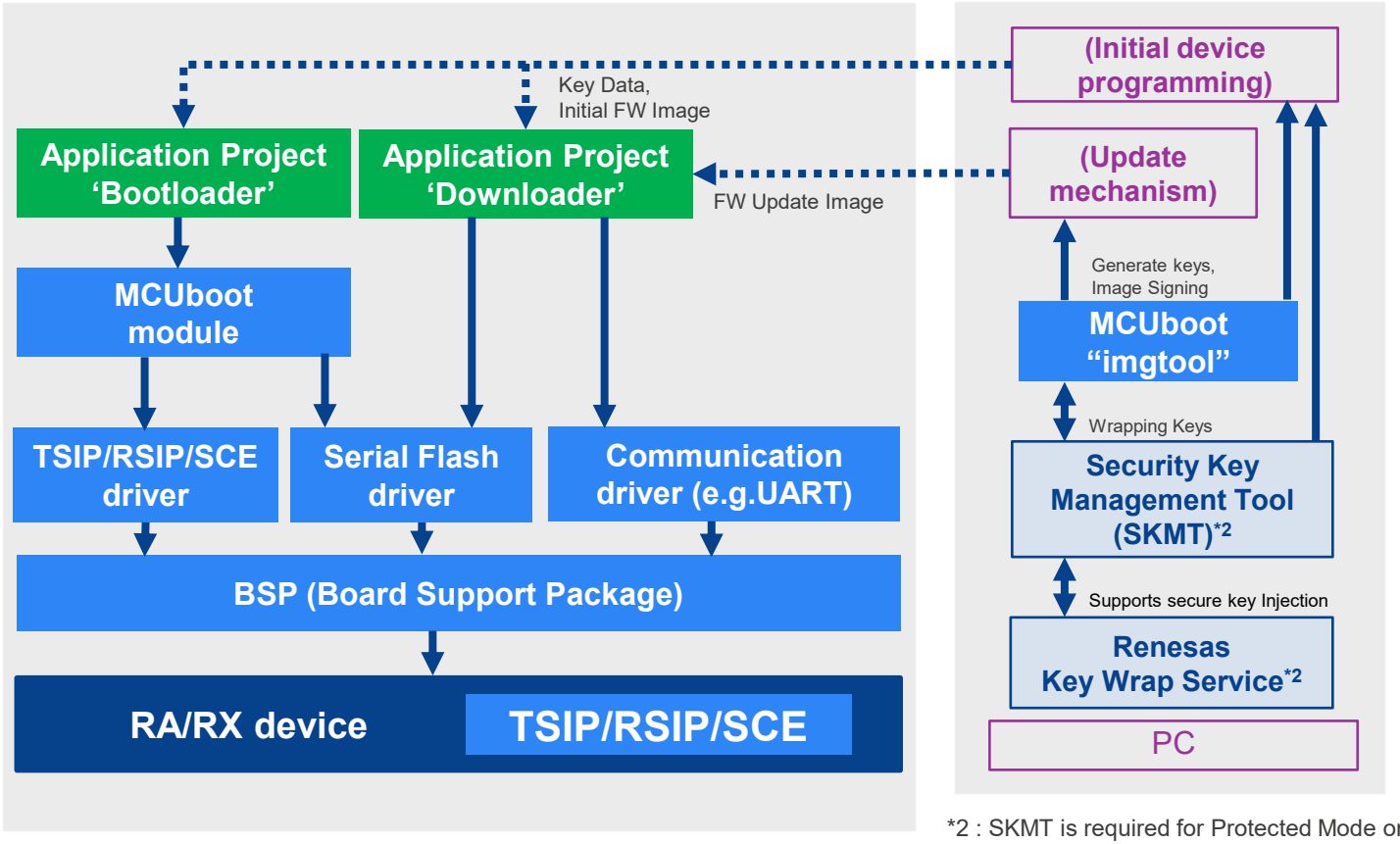
Secure firmware updates with signature verification

RA	RX
[With **SCE/RSIP**] **ECDSA NIST p256 + SHA256** **ECDSA NIST p384 + SHA384** **RSA 2048 + SHA256** **RSA 3072 + SHA256**	[With **RSIP**] **ECDSA NIST p256 + SHA256** [With **TSIP**] **ECDSA NIST p256 + SHA256** **RSA 2048 + SHA256**

Getting Started Guide and Sample Program

- MCUboot Module Guide （RA FSP / RX FIT）
- Sample Program (Published on the Renesas web page)

System configuration of the bootloader and demo application



*2 : SKMT is required for Protected Mode only

This page shows for RA family.

MCUboot solution for secure FW UP

For MCUs without the Renesas Security Engine,
Secure Firmware Update is enabled by Software Cryptographic Library of FSP's MCUboot module

■ MCUboot Feature

- ✓ **Secure bootloader** for 32-bit MCUs
- ✓ **Open-Source Software(OSS)** for the purpose of standardizing the bootloader and system flash infrastructure.
- ✓ Flexible design that can be integrated with a variety of hardware and operating systems
- ✓ This is a primary bootloader solution for embedded MCU systems

Reference : MCUboot | mcuboot
GitHub - mcu-tools/mcuboot: Secure boot for 32-bit Microcontrollers!

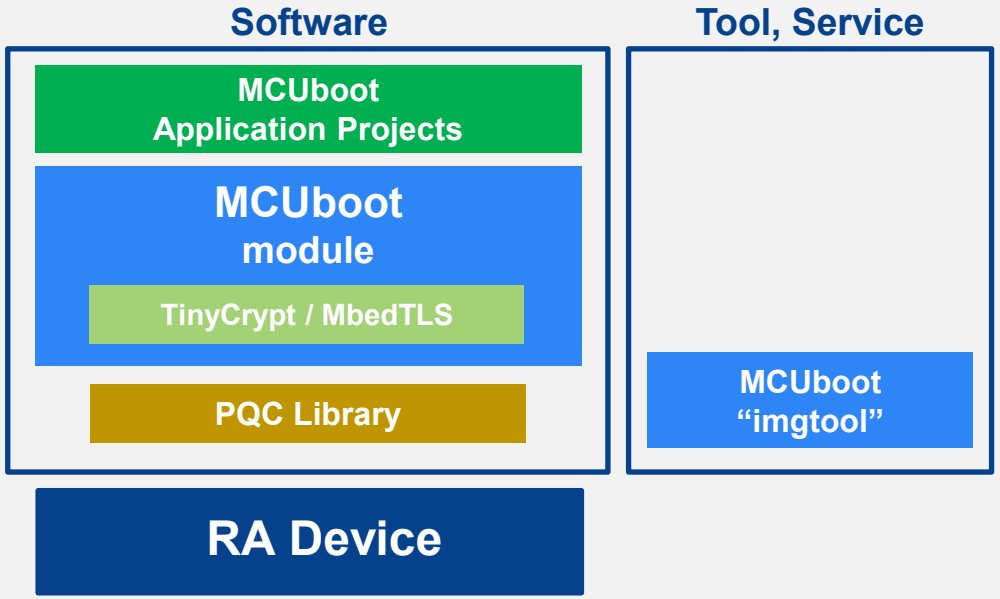
■ Software Cryptographic Library (FSP MCUboot module)

- ✓ **Secure FW update with signature verification**
  In RA Family MCUs **without** the Renesas Security Engine, secure firmware updates can be enabled by using the MCUboot module of FSP Software Cryptographic Library.
- ✓ **Supports PQC (Post-Quantum Cryptography)** NEW
  NIST (National Institute of Standards and Technology) recommends that digital products released after in 2035 be PQC compliant.
  RA/FSP supports PQC feature with MCUboot from FSP v6.4.0 onwards.

< Software Cryptographic Libraries >

- RA0, RA2 Series : Uses ' **TinyCrypt** ' in MCUboot
- RA4, RA6, RA8 Series : Uses ' **MbedTLS** ' in MCUboot
- RA Family : Uses Renesas own **PQC Library**

Renesas Deliverables



Supported Devices

RA Family : All RA0, RA2, RA4, RA6, RA8 series
RX Family : - (For products without a hardware security engine, Please kindly refer to the Firmware Update Module.)

FEATURES AND CONFIGURATION OF MCUBOOT SOLUTION FOR RA FAMILY

This page shows for RA family.

Realize Secure Firmware Updates with Software Cryptographic Library of FSP's MCUboot module

Signature Verification: Software

Firmware Signature Verification at Startup

Verify the authenticity of the firmware when MCU starts up.

Secure firmware update

Verify the authenticity of the firmware after receiving the entire update image.

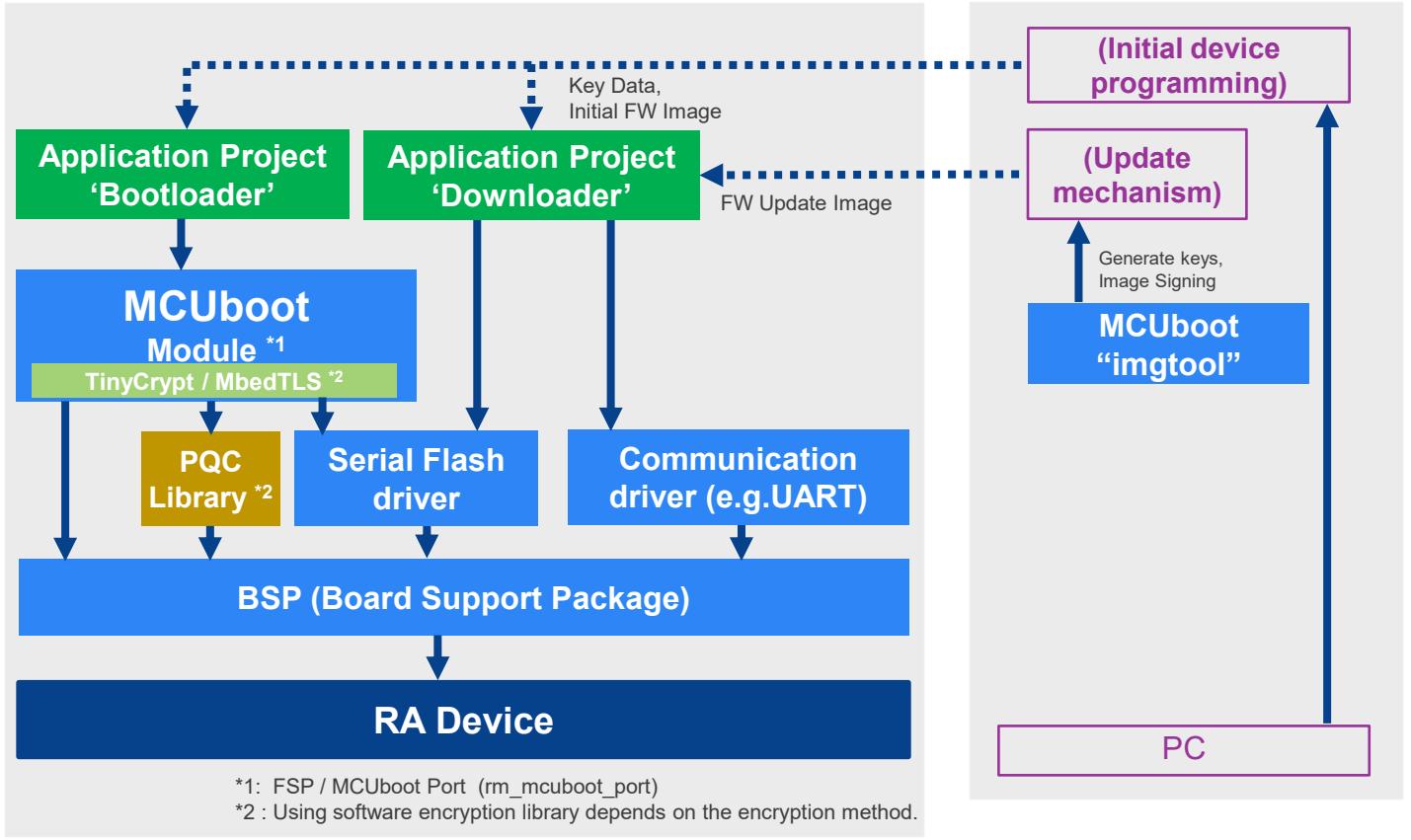
Secure firmware updates with signature verification

RA	RX
➢ **RA0, RA2 (TinyCrypt):** ECDSA NIST p256 + SHA256 ➢ **RA4, RA6, RA8 (MbedTLS):** ECDSA NIST p256 + SHA256, ECDSA NIST p384 + SHA384, RSA 2048 + SHA256, RSA 3072 + SHA256 ➢ **RA Family (PQC Library) :** ML-DSA-44/65/87	— For products without a hardware security engine, please use the Firmware Update Module.

Getting Started Guide and Sample Program

- MCUboot Module Guide (RA FSP)
- Sample Program (Published on the Renesas web page)

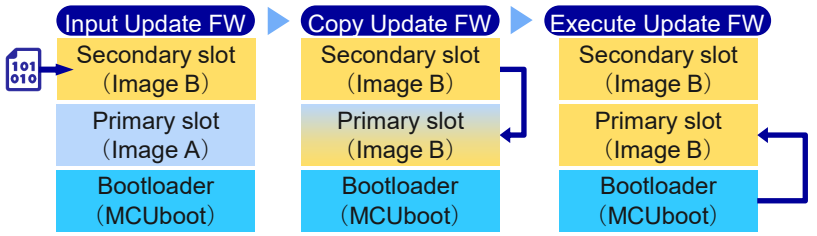
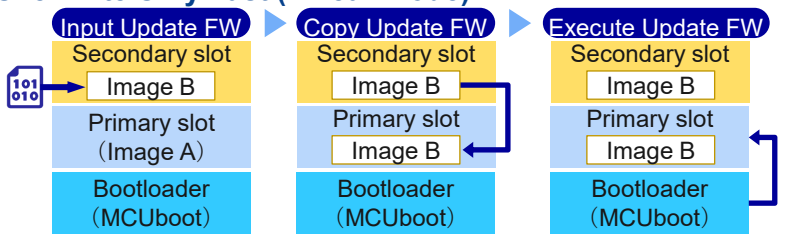
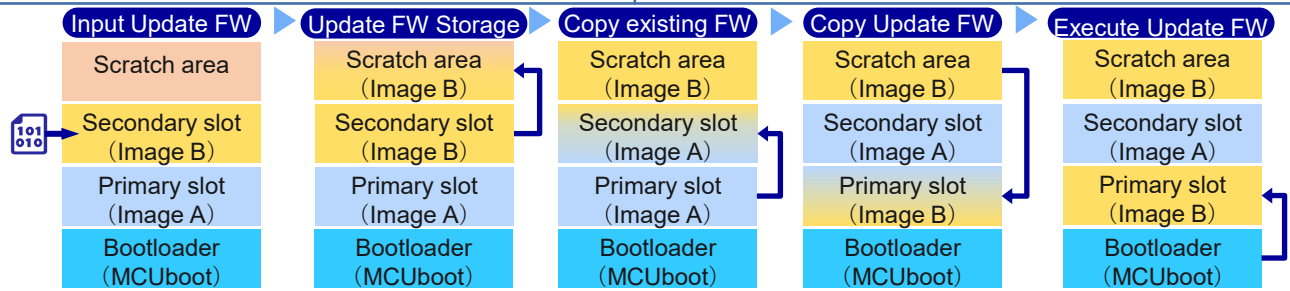
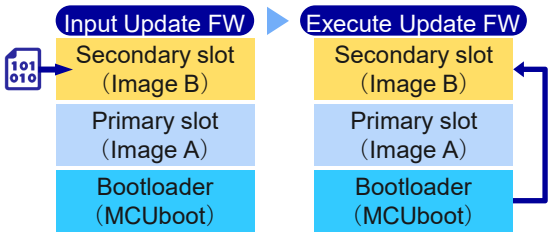
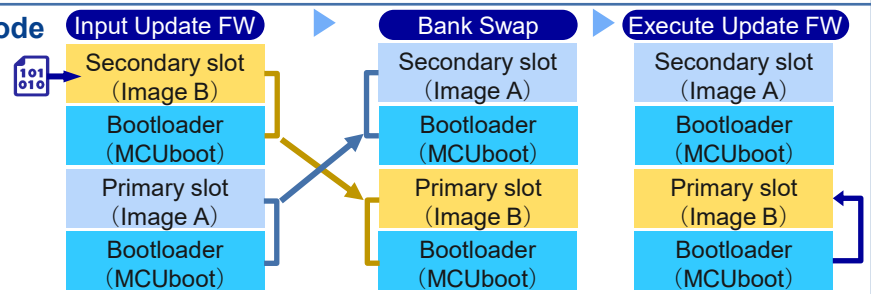
System configuration of the bootloader and demo application



RA & RX MCUBOOT UPDATE METHOD

The update firmware is acquired by the user application in the Primary slot (at Flash ROM).
Flash memory write operations in Linear mode are executed entirely at RAM area.

Selectable MCUboot Update Method according to the use case of your system!

Update Method	Update Procedure	Feature／Merit
Overwrite Only Method	**Overwrite Only (Linear mode)**  Copy entire Secondary slot to entire Primary slot **Overwrite Only Fast (Linear mode)**  Copy only image size for update of secondary slot to the primary slot	✓ Low memory footprint ✓ Overwrite the update image directly ✓ Update image encryption available
Swap Method	**Linear mode** 	✓ Update image encryption available
DirectXIP Method	**Linear mode**  **Dual mode** 	✓ High-speed updates are possible without the need for data movement such as copying.

MCUboot checks the version information defined at firmware creation, performs signature verification, and always boots the most recent update firmware image.

Renesas.com

Appendix :

Introduction to Firmware Update Solutions

FOR OPTIMAL FIRMWARE UPDATES ON IOT PRODUCTS BY RENESAS MCU FAMILY

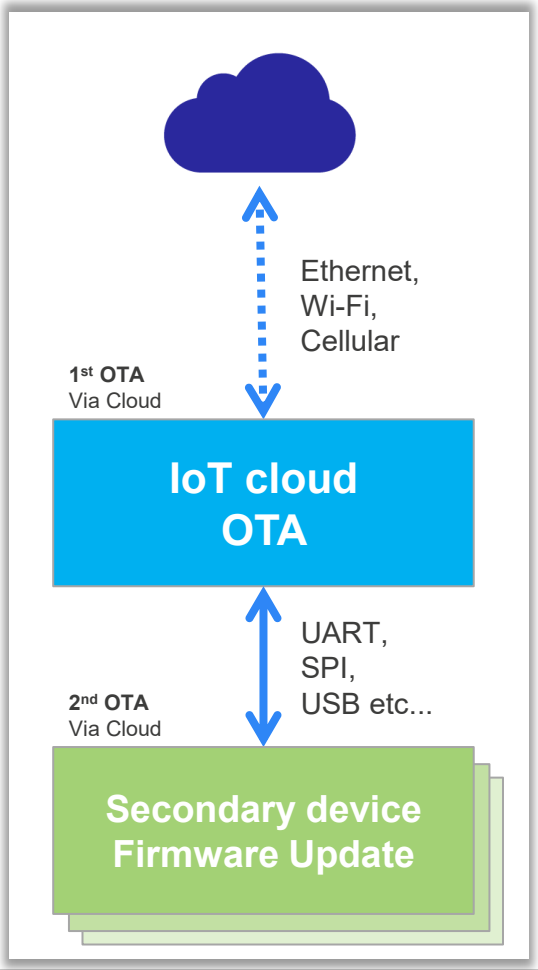
Support for firmware updates via **OTA(Over the Air)** through the cloud!

MCU Family	Secondary device Firmware Update	IoT cloud OTA
RENESAS RA	RA0, RA2, RA4 Series Non-RTOS	RA6, RA8 Series MQTT / OTA / TLS / TCP/IP*1, Fleet Provisioning (Coming soon) FreeRTOS
RENESAS RX	RX100/200 Series Non-RTOS	RX65N MQTT / OTA / TLS / TCP/IP*1 / Fleet Provisioning FreeRTOS
RENESAS RL78	RL78/G22, RL78/G23, RL78/G24, RL78/L23 Non-RTOS	RL78/G23 MQTT / OTA / FreeRTOS *2

Introduced in this material

*1 : Ethernet(MCU) Only

*2 : When using with these components with RL78/G23 (max ROM 768KB, RAM 48KB), RAM available for your app may be less than 10KB.



FIRMWARE UPDATE SOLUTIONS LIST

		RENESAS RX	RENESAS RL78
Secondary Device Firmware Update	FW UP Module Sample code	✓ RX Family Firmware Update Module Using Firmware Integration Technology Application Notes - Sample Code	✓ RL78/G22, RL78/G23, RL78/G24, RL78/L23 Firmware Update Module
	Firmware Updating Communications Module	✓ RX Family Firmware Updating Communications Module Using Firmware Integration Technology	✓ RL78/G23 Firmware Updating Communications Module
IoT Cloud OTA	Secondary device OTA Firmware Update Sample code	✓ RX65N Group Sample Code for OTA Update of a Secondary Device by Amazon Web Services with the Use of FreeRTOS	✓ RL78/G23 Sample Code for OTA Update of a Secondary Device by Amazon Web Services with the Use of FreeRTOS
	OTA firmware Update sample code	✓ RX Family How to Implement FreeRTOS OTA Using Amazon Web Services (202406-LTS Version)	✓ Getting Started Guide for Connecting Amazon Web Services in Wi-Fi Communication: RL78/G23-128p Fast Prototyping Board + FreeRTOS
	Development Support Tool	✓ QE for OTA: Development Assistance Tool for Firmware Update	

WIRELESS MODULE-BASED FIRMWARE UPDATE SOLUTIONS LIST

		RENESAS RX	RENESAS RL78
Wi-Fi	**Utilizing the MCU firmware update command of Wi-Fi DA16600 module**	✓ RX Family AWS Cloud Connectivity for MCU Firmware Update Over-the-Air on RX65N Cloud Kit Board with Wi-Fi DA16600 Application Note | Sample Code	✓ RL78/G23 AWS Cloud Connectivity for MCU Firmware Update Over-the-Air on RL78/G23-128p Fast Prototyping Board with Wi-Fi DA16600 | Sample Code
Bluetooth LE	**Utilizing Renesas SUOTA Service of the Bluetooth LE DA14535/DA14531 module**	✓ RX Family Renesas Firmware Update Over the Air (FOTA) Sample Program With Bluetooth Low Energy DA14535/DA14531 Application Note | Sample Code	✓ RL78 Family Renesas Firmware Update Over the Air (FOTA) Sample Program With Bluetooth Low Energy DA14535/DA14531 Application Note | Sample Code

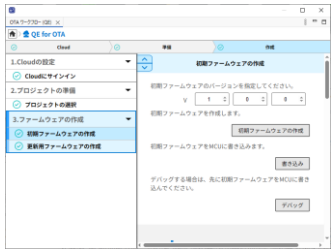
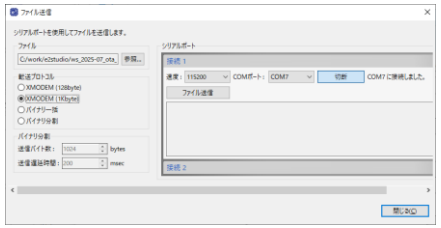
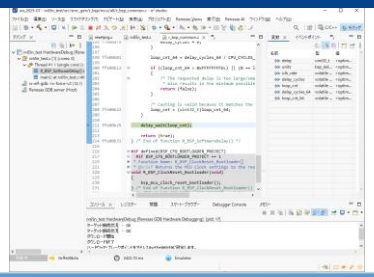
QE FOR OTA:
DEVELOPMENT ASSISTANCE TOOL FOR FIRMWARE UPDATE

From Over the Air (OTA) to Local application update
Firmware update with various system structure can be executed with simple GUI

With/Without Cloud(AWS)		Update Target		Update Method		
With Cloud	Without Cloud	Primary	Secondary	Dual-Bank	Partial update	Full Update
Wi-Fi / Ethernet	Serial Communication	FreeRTOS / non-OS	non-OS	Bank0: main / Bank1: buffer	Single Bank: main, buffer	Single Bank: main

Quick and Effective tool solution

Reduce time required for 1 IoT device OTA by 86%! Support smooth PoC development!

Create Firmware, Bootloader → Generate & implement Signature verification info → Execute Firmware Update → Debug

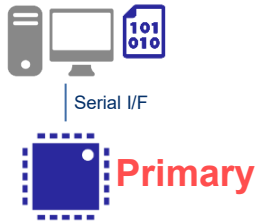
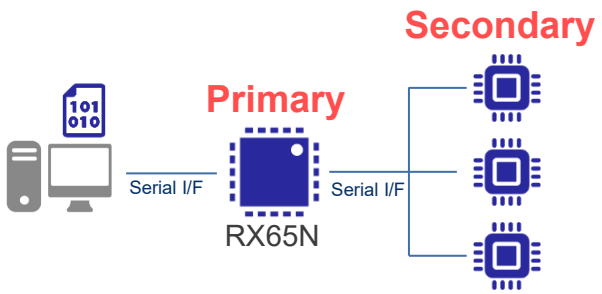
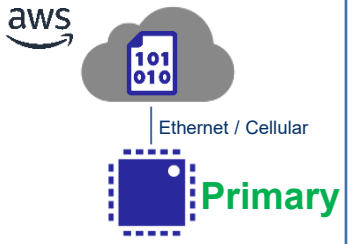
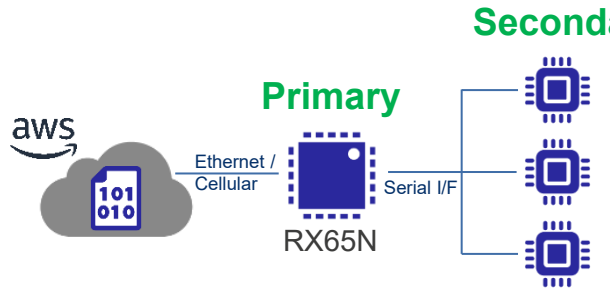
QE for OTA

OTA AND FIRMWARE UPDATE BY “QE FOR OTA”

QE for OTA

< More information >
- QE for OTA: Development Assistance for Cloud
- Firmware Update module

: Update FW Location

	Firmware Update WITHOUT using Cloud			Firmware Update via Cloud		
						
Update Target	**Primary**	**Primary**	**Secondary**	**Primary**	**Primary**	**Secondary**
Supported Devices	**<RA>** RA6M4, RA6M5 **<RX>** RX Family – All series **<RL78>** RL78/G22,G23,G24,L23	**<RX>** RX65N	**<RX>** RX23E-B, RX66T, RX660, RX261, RX140 **<RL78>** RL78/G23	**<RA>** RA6M5 **<RX>** RX65N **<RL78>** RL78/G23	**<RX>** RX65N	**<RX>** RX23E-B, RX66T, RX660, RX261, RX140 **<RL78>** RL78/G23
with / without RTOS	non-OS	FreeRTOS, non-OS	non-OS	FreeRTOS	FreeRTOS	non-OS
Flash Memory Update method	**< RA, RX >** Dual-Bank Method **< RL78 >** Partial Update Method	**< RX >** Partial Update Method	Dual-Bank Method*1, Partial Update Method, Full Update method	**< RA, RX >** Dual-Bank Method **< RL78 >** Partial Update Method	Dual-Bank Method	Dual-Bank Method*1, Partial Update Method, Full Update method
Communication method	Serial Communication*2			**< RA, RX >** Ethernet **< RL78 >** Cellular		Serial Communication*2

*1: Only supported by Dual-Bank mode-compatible devices. *2: Using " Serial Communication Firmware Updating Communications Module (RX, RL78) " for communication between MCUs.