AN13505

OTA Storage Structure on RT1010

Rev. 0 — 20 January 2022 Application Note

Contents

1 Introduction

NXP has released a set of OTA projects based on MCU on Github. These projects support the i.MXRT series and related security functions, which have attracted widespread attention from users. However, these projects are based on EVK development board of NXP. The capacity of the onboard Flash determines the storage structure of the entire OTA project. Therefore, the OTA storage structures vary from each other due to the differences in the capacity of Flash. This application note introduces the default storage structure and summarizes some latest experiences while supporting customers. To replace the Flash, complete the configuration of the OTA projects faster.

2 OTA storage structure overview

The entire OTA consists of SBL and SFW. You can download them from:

- SBL: https://github.com/NXPmicro/sbl
- SFW: https://github.com/NXPmicro/sfw

The onboard flash is an 8 MB QSPI Flash, so 8 MB of space is allocated. Figure 1 shows the storage structure of the entire OTA. OTA includes SBL (Secure Bootloader), OTA Flag Data, Slot 1, Slot 2, and Customer Data areas.

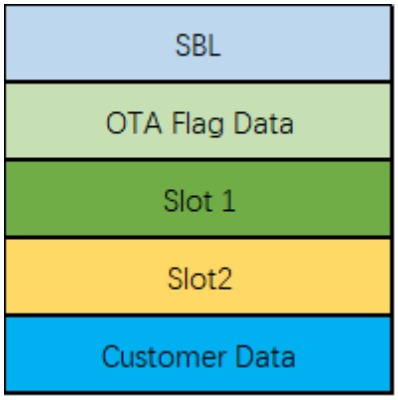

Figure 1. Storage structure

2.1 Secure BootLoader (SBL)

After the chip POR is started, SBL determines the execution of programs stored in Slot 1 or Slot 2 according to the information of OTA Flag Data. SBL supports program verification, revert, and other functions.

NOTE

The remap function needs 4 kB alignment.

Table 1 illustrates the common address space allocation of SBL.

Table 1. SBL memory map

	Function	Add_Start	Add_End
SBL (1 MB - 4 kB)	OTFAD	0x60000000	0x600003FF
	m_boot_hdr_conf_start	0x60000400	0x60000FFF
	m_boot_hdr_ivt_start	0x60001000	0x6000101F
	m_boot_hdr_conf_start	0x60001020	0x6000102F
	m_boot_hdr_dcd_start	0x60001030	0x60001FFF
	SBL_Code_Vector_Start	0x60002000	0x600FEFFF

2.2 OTA Flag Data

OTA Flag Data area is used to store some flag information during the process of OTA update. According to the flag information of SBL, the image can be updated, reverted, and jumped to the corresponding slot running program. Table 2 shows the address space allocation of this area.

Table 2. OTA Flag Data memory map

	Function	Add_Start	Add_End
Reserved (4 kB - 32 B)	Reserved	0x600FF000	0x600FFFDF
OTA Flag Data (32 B)	OTA Flag	0x600FFFE0	0x600FFFFF

2.3 Slot 1 and Slot 2

Slot 1 and Slot 2 are used to store the application program. The remap function can switch between Slot 1 and Slot 2, which requires 4 kB alignment. Slot 1 and Slot 1 each has 1 MB space. Table 3 shows the common address space allocation of Slot 1 and Slot 2.

Table 3. Slot 1 and Slot 2 memory map

	Function	Add_Start	Add_End
Slot 1 (1 MB)	Image Header	0x60100000	0x601003FF
	Application Image	0x60100400	0x601FFFFF
Slot 2 (1 MB)	Image Header	0x60200000	0x602003FF
	Application Image	0x60200400	0x602FFFFF

2.4 Customer Data

Customer Data is used to store the information of customers (optional, not compulsory). Table 4 shows the address space allocation of customer data area.

Table 4. Customer Data memory map

	Function	Add_Start	Add_End
Customer Data (5 MB)	Customer Data	0x60300000	0x607FFFFF

3 Example: How to customize a storage structure

3.1 Overview

This chapter introduces storage space allocation and some key points in the allocation process combined with real customer application cases. The customer demands to use a Flash with a capacity of 512 kB and does not need related Security functions. Therefore, after disabling the Security function, use IAR to compile a 25 kB `SBL.bin` file. Since the Remap function used in OTA requires 4 kB alignment and the minimum erasing capacity of Flash is 4 kB, the allocated capacity of SBL is 28 kB. Table 5 describes the current address space allocation of SBL.

Table 5. SBL space allocation

Function	Add_Start	Add_End
SBL	0x60000000	0x60006FFF

Although OTA Flag Data has only 32 Bytes data, these data must be read, written, and erased during the update process. However, the erasing process of Flash must be carried out according to the size of the Sector. Therefore, it requires at least 4 kB space. Table 6 describes the current address and space allocation of OTA Flag Data.

Table 6. OTA flag data space allocation

Function	Add_Start	Add_End
OTA Flag Data	0x60007000	0x60007FFF

The remap function can switch between Slot 1 and Slot 2. The address of Remap needs 4 kB alignment. Table 7 describes the address space to store the application layer program.

Table 7. Slot 1 and Slot 2 space allocation

Function	Add_Start	Add_End
Slot 1	0x60008000	0x60043FFF
Slot 2	0x60044000	0x6007FFFF

So far, the entire 512 K Flash space is used up. As customer does not need the customer data area while aiming to take full advantage of the space on the application programs.

3.2 Address space allocation of specific demands

First, the SBL area contains relevant information used for Flash boot, such as, IVT, Flash Config Block, etc. This part of code can be used directly without modification. In terms of address space allocation, the SBL area can be regarded as a Flash XIP boot **Hello World** project. Table 8 shows the SBL address space allocation.

Table 8. SBL address space allocation

	Function	Add_Start	Add_End
SBL (At Least 28 kB)	OTFAD	0x60000000	0x600003FF
	m_boot_hdr_conf_start	0x60000400	0x60000FFF
	m_boot_hdr_ivt_start	0x60001000	0x6000101F
	m_boot_hdr_conf_start	0x60001020	0x6000102F
	m_boot_hdr_dcd_start	0x60001030	0x60001FFF
	SBL_Code_Vector_Start	0x60002000	0x60006FFF

Second, OTA Flag Data has total 32 bytes that are used for indicating three statuses: update, revert, and none (no update and no revert).

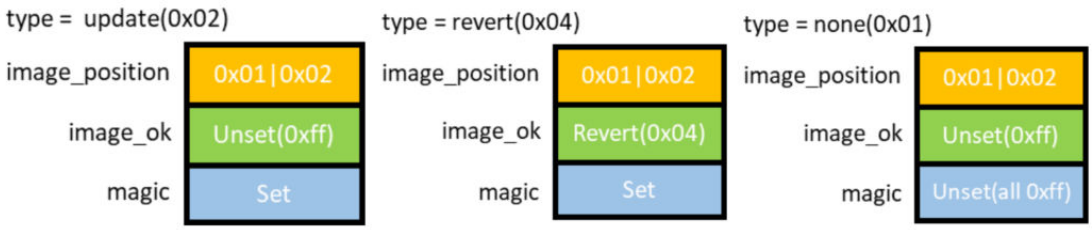

Figure 2. Three statuses of OTA flag data

These 32 bytes are normally stored before the head address of Slot 1.

Table 9. OTA Flag Data address space allocation

	Function	Add_Start	Add_End
OTA Flag Data (4 kB - 32 B)	User Data	0x60007000	0x60007FDF
32 B	OTA Flag	0x60007FE0	0x60007FFF

Finally, when it comes to the address space allocation of Slot 1 and Slot 2, the head address of the application program, namely the head address of the interrupt vector table, does not start from the head address of Slot 1.

There are two reasons:

- At the start address of the image, 32 bytes of Image Header information are used for OTA.
- The start address of the interrupt vector table in the application program must be calculated.

The basic calculation rule is: the result of 4 times the number of interrupt vectors and then aligns the result up to an integer multiple of a power of 2. Although the total number of interrupt vectors of RT1010 is 256, the actual number of available interrupt vectors is 96. So the size of the interrupt vector table is 96 * 4 = 384. The result of 384 align up to an integer multiple of a power of 2 is 512, namely `0x200`. As a result, the actual start address of the application programs is `0x200`. The space allocation of Slot 2 must satisfy this requirement too.

Table 10. Slot 1 and Slot 2 address space allocation

	Function	Add_Start	Add_End
Slot 1 (240 kB)	Image Header	0x60008000	0x600081FF
	Application Image	0x60008200	0x60043FFF
Slot 2 (240 kB)	Image Header	0x60044000	0x600441FF
	Application Image	0x60044200	0x6007FFFF

Table 11 shows the complete address space allocation.

Table 11. Complete address space allocation

	Function	Add_Start	Add_End
SBL (At Least 28 kB)	OTFAD	0x60000000	0x600003FF
	m_boot_hdr_conf_start	0x60000400	0x60000FFF
	m_boot_hdr_ivt_start	0x60001000	0x6000101F

Table continues on the next page...

Table 11. Complete address space allocation (continued)

	Function	Add_Start	Add_End
	m_boot_hdr_conf_start	0x60001020	0x6000102F
	m_boot_hdr_dcd_start	0x60001030	0x60001FFF
	SBL_Code_Vector_Start	0x60002000	0x60006FFF
OTA Flag Data (4 kB - 32 B)	User Data	0x60007000	0x60007FDF
32 B	OTA Flag	0x60007FE0	0x60007FFF
Slot 1 (240 kB)	Image Header	0x60008000	0x600081FF
	Application Image	0x60008200	0x60043FFF
Slot 2 (240 kB)	Image Header	0x60044000	0x600441FF
	Application Image	0x60044200	0x6007FFFF

Use key words shown in Table 12 when customizing a storage structure. Normally there are two important addresses: `BOOT_FLASH_ACT_APP` and `BOOT_FLASH_CAND_APP`. Other address information depends on these two important address information. The internal program can do calculations based on these two address information.

Table 12. Key words

OTA Flag Data (4 kB - 32 B)	User Data	BOOT_FLASH_ACT_APP - 4 KB (Do not need to modify, SW will handle it)
32 B	OTA Flag	BOOT_FLASH_ACT_APP - 32 B (Do not need to modify, SW will handle it)
Slot 1 (240 kB)	Image Header	BOOT_FLASH_ACT_APP
	Application Image	BOOT_FLASH_ACT_APP + 0x200
Slot 2 (240 kB)	Image Header	BOOT_FLASH_CAND_APP
	Application Image	BOOT_FLASH_ACT_APP + 0x200

NOTE

OTA method involved in this document is based on the Remap function. Therefore, it only supports RT1010, RT1060, RT1064, RT1170, and RT1160.

4 Revision history

Rev. number	Date	Substantive changes
0	20 January 2022	Initial release.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 20 January 2022

Document identifier: AN13505

arm