

Customer Training Workshop

Traveo™ II Boot

Q4 2020



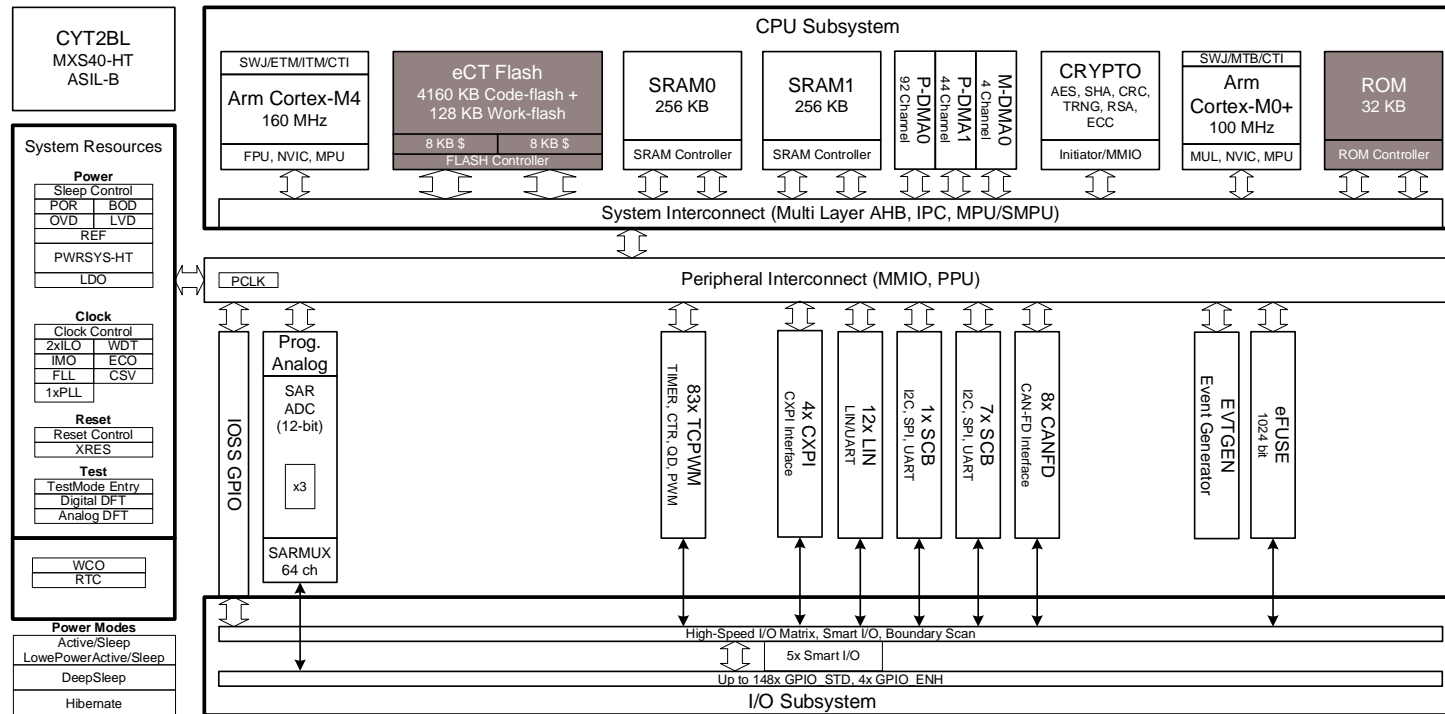
Target Products

› Target product list for this training material

Family Category	Series	Code Flash Memory Size
Traveo™ II Automotive Body Controller Entry	CYT2B6	Up to 576 KB
Traveo II Automotive Body Controller Entry	CYT2B7	Up to 1088 KB
Traveo II Automotive Body Controller Entry	CYT2B9	Up to 2112 KB
Traveo II Automotive Body Controller Entry	CYT2BL	Up to 4160 KB
Traveo II Automotive Body Controller High	CYT3BB/CYT4BB	Up to 4160 KB
Traveo II Automotive Body Controller High	CYT4BF	Up to 8384 KB
Traveo II Automotive Cluster	CYT3DL	Up to 4160 KB
Traveo II Automotive Cluster	CYT4DN	Up to 6336 KB

Introduction to Traveo II Body Controller Entry

› Boot is part of the CPU subsystem

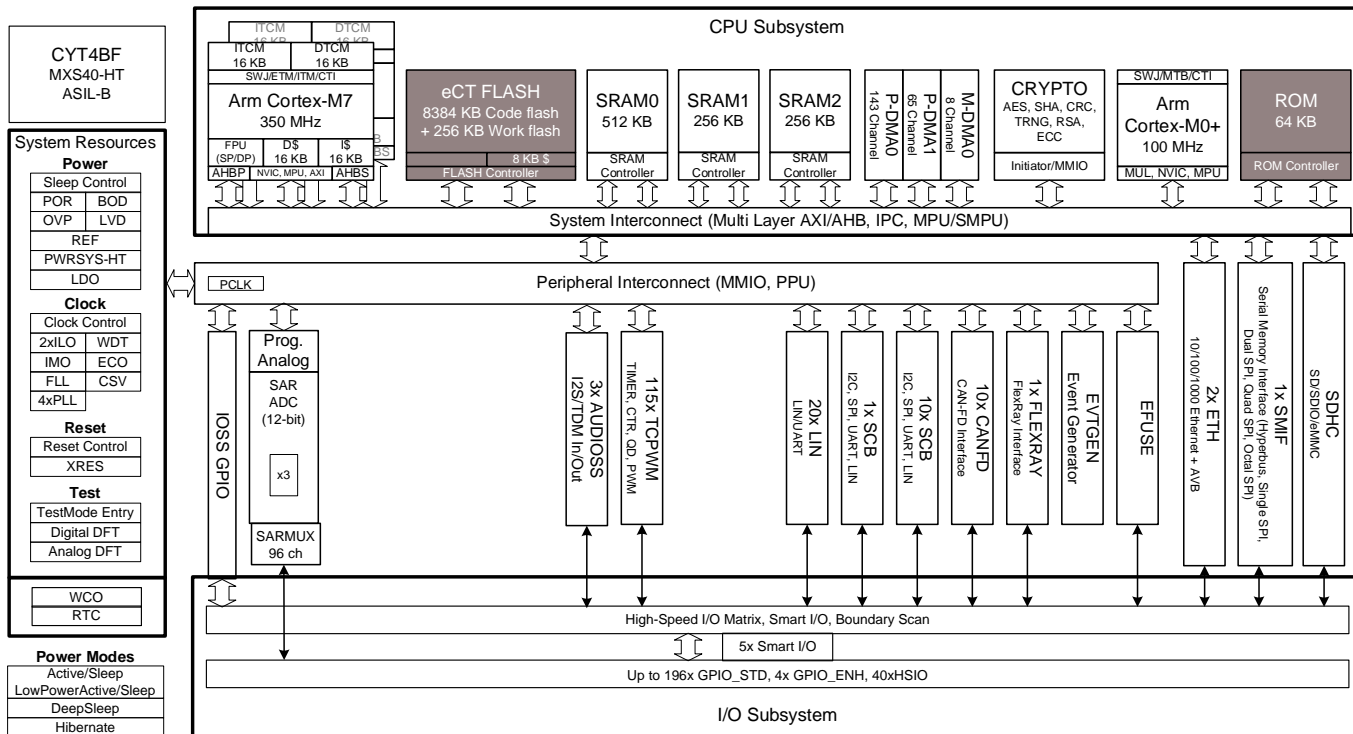


Hint Bar

Review TRM chapter 11 and 34 for additional details

Introduction to Traveo II Body Controller High

› Boot is part of the CPU subsystem

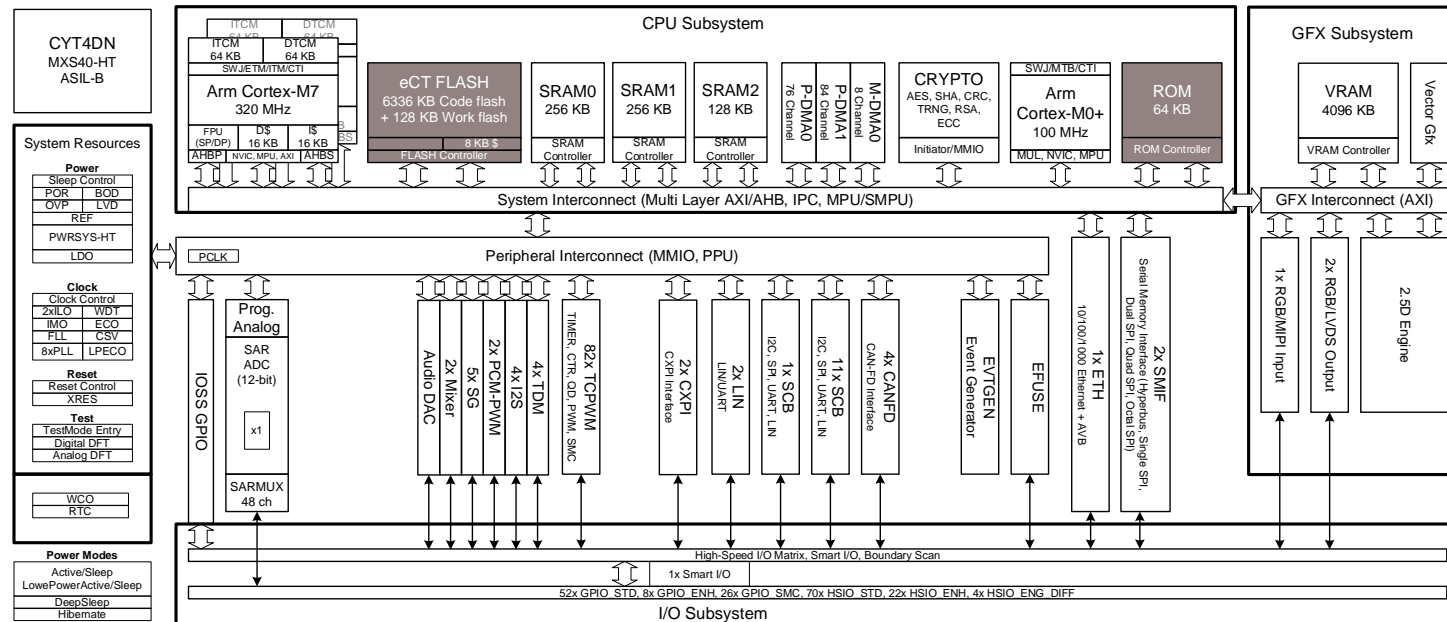


Hint Bar

Review TRM chapter 11 and 38 for additional details

Introduction to Traveo II Cluster

› Boot is part of the CPU subsystem



Hint Bar

Review TRM chapter 11 and 40 for additional details

Overview

> Features

- Traveo II has ROM and Flash boot
- After reset, CM0+ starts executing from the ROM boot
- CM4/CM7 are deactivated until the boot process is completed
- The user application runs on CM0+ after boot
- CM4/CM7 are activated by the CM0+ user application
- Supports secure boot
- Enables protection setting (configuration of MPU, SMPU, PPU, and SWPU)
- Enables the DAP access
- Enables system calls



Hint Bar

Review BootROM and Flash Boot TRM chapters for additional details

Arm® Cortex®-M0+ (CM0+)

Arm Cortex-M4 (CM4)

Arm Cortex-M7 (CM7)

Memory Protection Unit (MPU)

Shared memory protection unit (SMPU)

Peripheral protection unit (PPU)

Software protection unit (SWPU)

Debug access port (DAP)

Review datasheet for ROM and Flash Boot time.

ROM Boot

> Operation

- (A) Verifies the secure boot with eFuse information
- (B) Validates the Flash boot code and SFlash objects with eFuse information
- (C) Applies the appropriate protection state

State	Description
NORMAL	Lifecycle stage is NORMAL_PROVISIONED
SECURE	Lifecycle stage is SECURE or SECURE_WITH_DEBUG.
DEAD	Detected Corruption/error

- (D) Configures the protection unit (MPU, S MPU, PPU, and SWPU)
- (E) Goes to the next process:
 - If the validation passes, MCU will jump to Flash boot
 - If the validation fails, the following will be set, and MCU will be in idle state
 - Enables system calls
 - Sets PC for all masters
 - Enables DAP access

Hint Bar

Review BootROM TRM chapters for additional details

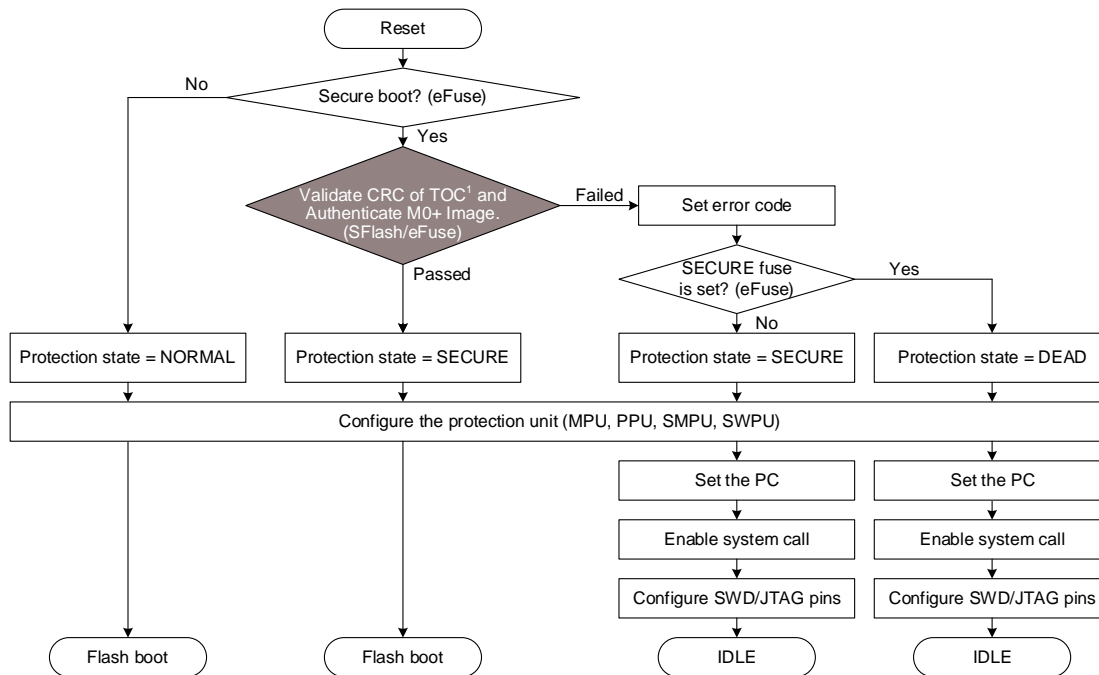
Supervisory Flash (SFlash)

For details of (A) to (E), see the next slide

ROM Boot Flow

> ROM boot proceeds based on the boot objects in SFlash and eFuse

- (A) Verifies the secure boot with eFuse info
- (B) Validates the Flash boot code and SFlash objects with eFuse info
- (C) Applies the protection state
- (D) Configures the protection unit
- (E) Goes to next process



Hint Bar

Review BootROM TRM chapters for additional details

For PC value details, see section 3 of the Flash Boot TRM chapter

¹ Table of contents (TOC): The TOC is divided into a Part1 object (TOC 1) set at the factory and a Part2 object (TOC 2) set by the user
 - TOC 1: This is an object for device protection setting
 - TOC 2: This is an object for the digital signature scheme

Flash Boot

> Operation

(A) Validates the TOC2 and secures image application¹ in Flash

- This happens only when the protection state is SECURE
 - If the validation fails, MCU will be in idle state
 - If the validation passes, MCU will proceed to the next step

(B) If necessary, MCU will jump to the bootloader

- This happens only when the protection state is NORMAL

(C) Enables the system calls

(D) Configures the protection units (MPU, SMPU, PPU, and SWPU)

(E) Sets the PC for all masters

(F) Launches the CM0+ and CM4/CM7 applications

- This happens only when the protection state is Normal or SECURE

Hint Bar

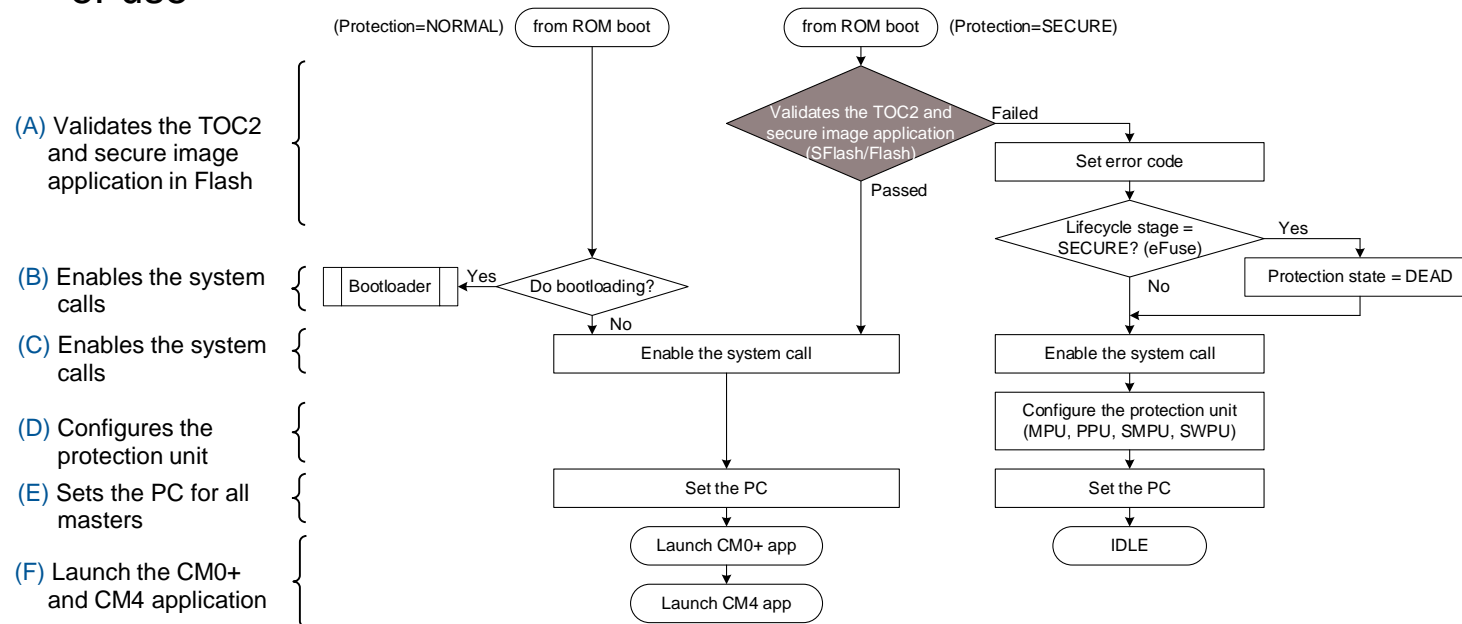
Review the Flash Boot TRM chapter for additional details

For details of (A) to (E), see the next slide.

¹ The secure image application is the CM0+ user application including the software for validation of the CM4 user application.

Flash Boot Flow

› Flash boot proceeds based on the boot objects in SFlash, Flash, and eFuse



Hint Bar

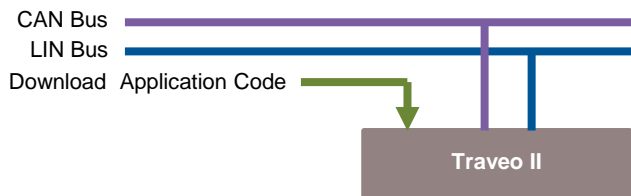
Review the Flash Boot TRM chapter for additional details

For PC value details, see section 3 of the Flash Boot TRM chapter

Bootloader

> Overview

- The bootloader is part of the Flash boot code
- The bootloader downloads the user application (Flash Loader) through the CAN or LIN interface and stores it in the RAM



> Bootloader Activation Conditions

- The internal bootloader will activate if all these conditions are met:
 - Two words at the start of flash must be 0xFFFF_FFFF.
 - TOC2 is valid and TOC2_FLAGS bit FB_BOOTLOADER_DISABLE should be 2'b01 (default). Otherwise, TOC2 is erased.
 - Protection mode is not SECURE or SECURE_DEAD.
 - No debugger connection occurs during a 1-second wait window.

Hint Bar

Review the Flash Boot TRM chapter for additional details

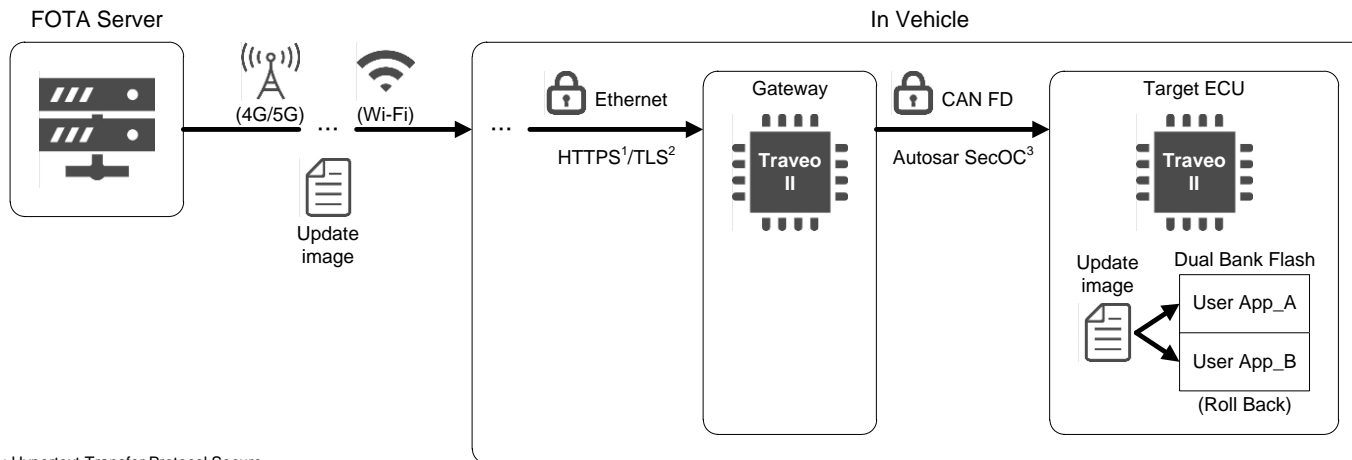
Secure Firmware Over The Air (FOTA)

> Use Case

- Traveo II can update the user application while preventing unauthorized access to the network and data tampering using the Hardware Security Module (HSM)
- Rollback by Dual Bank Flash: If a new application encounters a critical error, you can roll back to the previous running application

Hint Bar

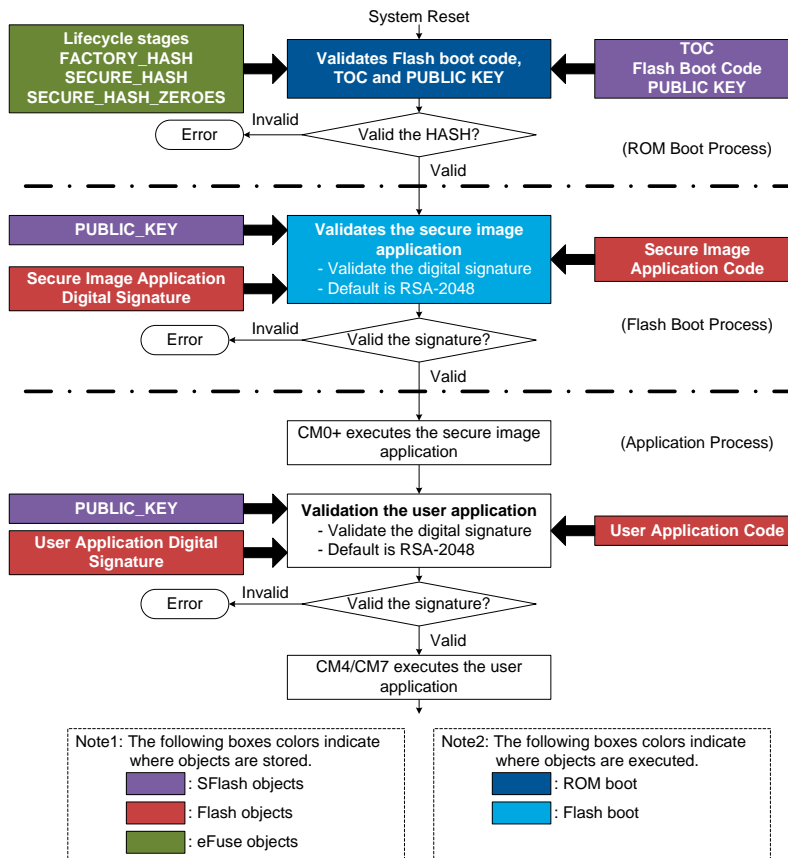
Review the Flash Boot TRM chapter for additional details



¹ HTTPS : Hypertext Transfer Protocol Secure
² TLS : Transport Layer Security
³ SecOC : Secure Onboard Communication

Secure Boot

- > Guarantees that only the intended firmware runs on the system
- > Operation
 - The ROM boot validates Flash boot code, TOC, and PUBLIC KEY
 - The Flash boot validates the secure image application
 - The secure image application validates user application for CM4/CM7

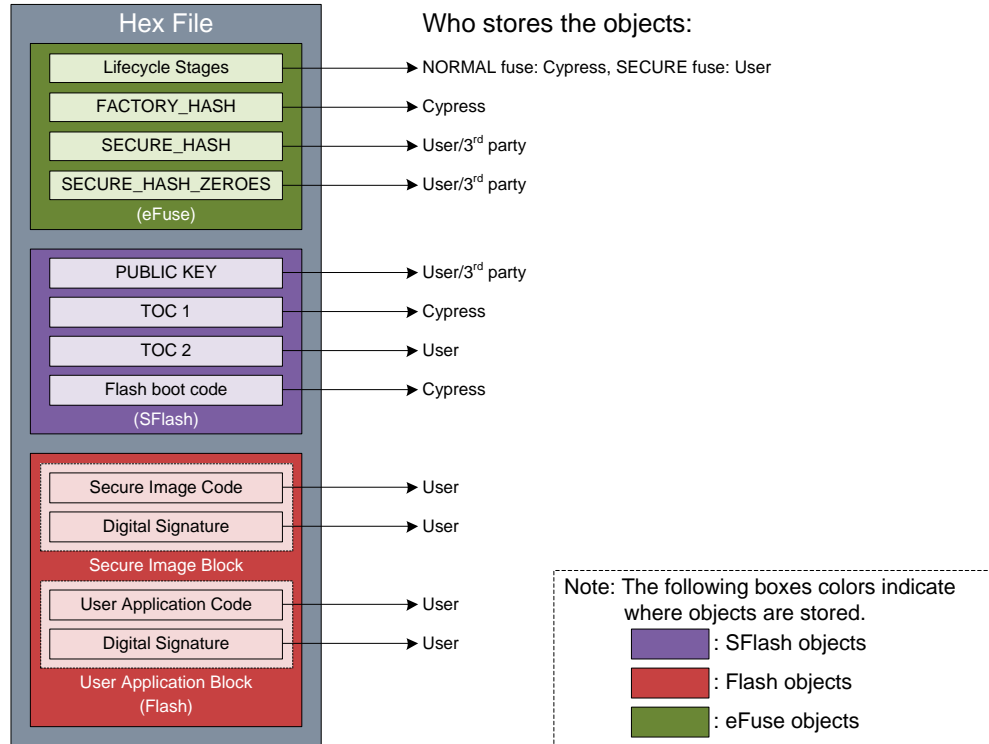


Hint Bar

Review the BootROM and Flash Boot TRM chapters for additional details

Secure System Hex File

- > The secure system hex file contains several objects and not just the user's code

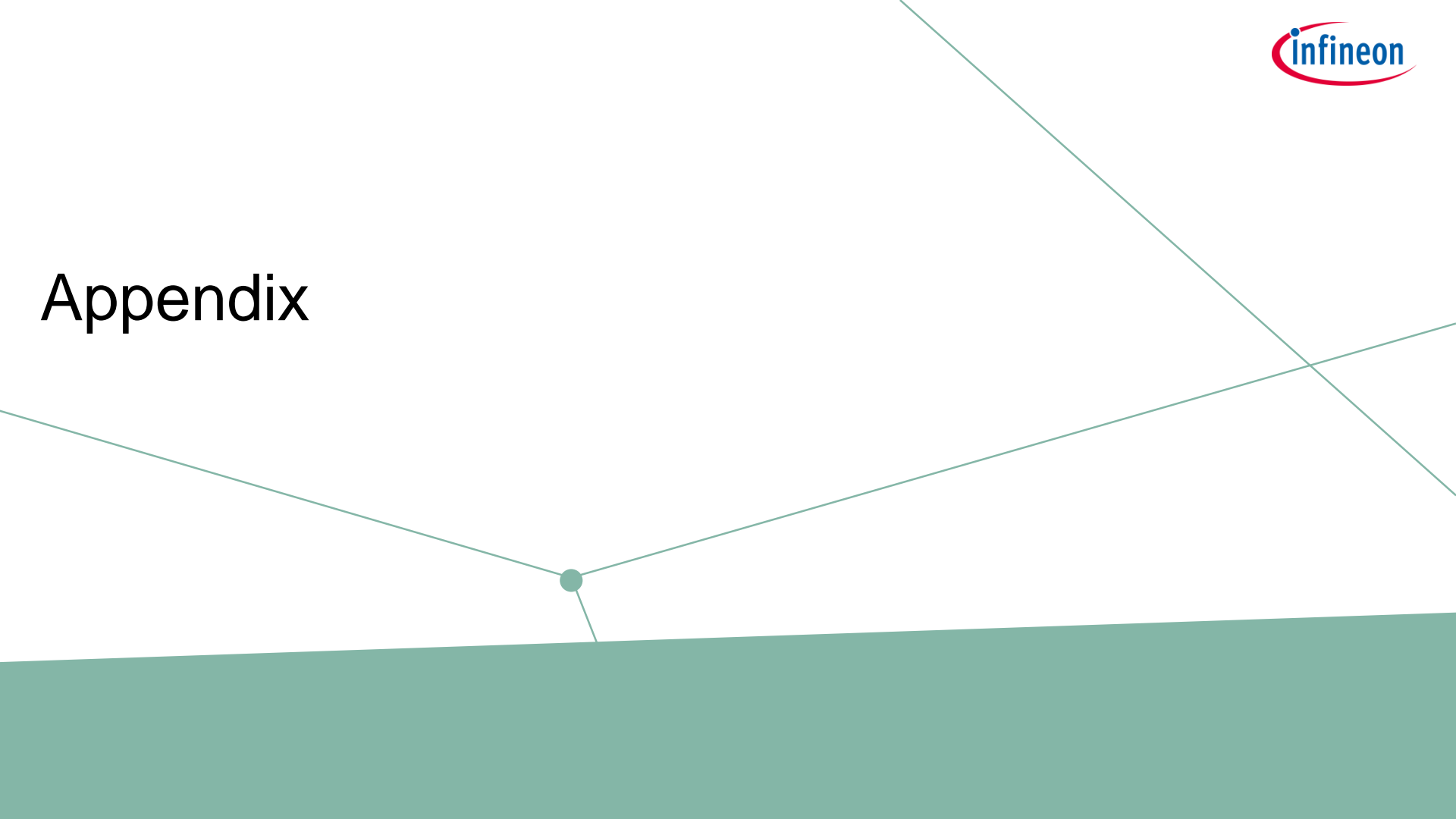


Hint Bar

Review the Flash Boot TRM chapter for additional details

For details, see the Appendix

Appendix



Definition of eFuse Bits (1/2)

Names		Bits	Description
SECURE Access Restrictions	AP_CTL_CM0_DISABLE	1:0	Indicates that this device does not allow access to the CM0+ access port.
	AP_CTL_CM4_DISABLE ¹	3:2	Indicates that this device does not allow access to the CM4 access port.
	AP_CTL_SYS_DISABLE	5:4	Indicates that this device does not allow access to the system access port.
	SYS_AP_MPU_ENABLE	6	Indicates that the MPU on the system access port must be programmed and locked according to the settings in the next six fields.
	DIRECT_EXECUTE_DISABLE	7	Disables DirectExecute system call functionality (implemented in software).
	FLASH_ALLOWED	10:8	This field indicates what portion of main flash is accessible through the system access port. Only a portion of flash starting at the bottom of the area is exposed.
	SRAM_ALLOWED	13:11	This field indicates what portion of SRAM is accessible through the system access port. Only a portion of SRAM starting at the bottom of the area is exposed. Encoding is the same as FLASH_ALLOWED.
	WORK_FLASH_ALLOWED	15:14	This field indicates what portion of work flash is accessible through the system access port. Only a portion of work flash starting at the bottom of the area is exposed.
	SFLASH_ALLOWED	17:16	This field indicates what portion of supervisory flash is accessible through the system access port. Only a portion of supervisory flash starting at the bottom of the area is exposed.
MMIO_ALLOWED	19:18	This field indicates what portion of the MMIO region is accessible through the system access port.	

Hint Bar

Review the BootROM and Flash Boot TRM chapters for additional details

¹ It applies to products on which CM4 is implemented.

Definition of eFuse Bits (2/2)

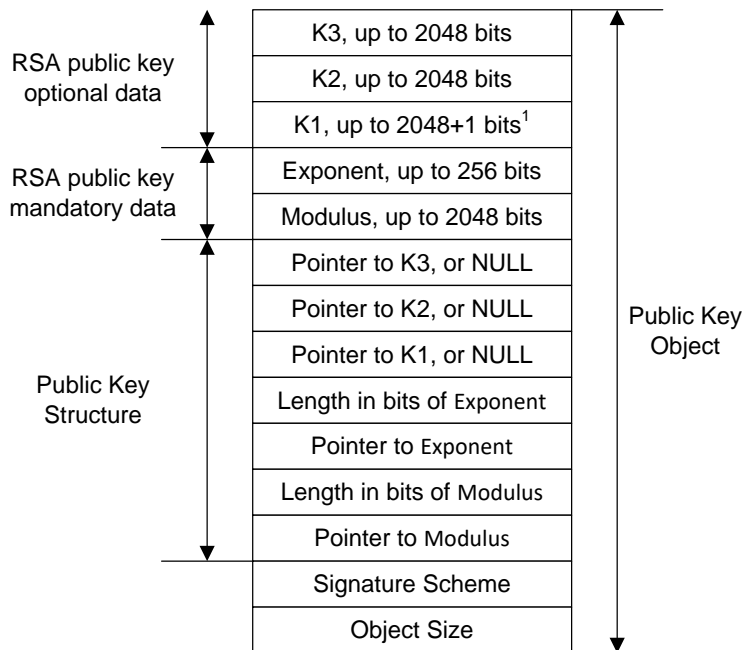
Names		Bits	Description
SECURE Access Restrictions	SMIF_XIP_ENABLE	20	This field indicates what portion of SMIF_XIP is accessible through the system access port.
DEAD Access Restrictions	<Same as SECURE Access Restrictions>		The structure is identical to the one above but used when entering DEAD mode. It assumes that this structure is more restrictive than SECURE.
Critical Object Hash	FACTORY_HASH		SHA-256 (upper 128 bits) that covers objects in TOC Part1. It is checked before transitioning to SECURE_WITH_DEBUG or SECURE.
	SECURE_HASH		SHA-256 that covers the flash boot image and other objects in TOC Part1 and Part2. Flash boot code is not started unless this value is correct.
	SECURE_HASH_ZEROES		The number of bits that are '0' (fuses that are not blown) in the SHA-256. This guarantees that when a HASH is programmed, it cannot be changed into another valid HASH value.

Hint Bar

Review the BootROM and Flash Boot TRM chapters for additional details

Definition of PUBLIC KEY in SFlash

- > The structure of the key object used for signature verification is shown in the diagram



Hint Bar

Review the Flash Boot TRM chapter for additional details

¹ Modulus, Exponent, K1, K2, and K3 must be 32-bit aligned, the data is little endian.

Definition of PUBLIC KEY in SFlash

Public Key Object Member Name	Description
Object Size	A size in bytes used in SECURE_HASH calculation for a Public Key data protection.
Signature Scheme	A signature scheme. 0 - RSASSA-PKCS1-v1.5 with RSA-2048 and SHA-256 (other values are reserved).
Pointer to Modulus	A pointer to an RSA public key modulus data.
Length in bits of Modulus	A length in bits of an RSA public key modulus.
Pointer to Exponent	A pointer to an RSA public key exponent data.
Length in bits of Exponent	A length in bits of an RSA public key exponent data.
Pointer to K1	A pointer to an optional RSA public key coefficient, named Barrett coefficient.
Pointer to K2	A pointer to an optional RSA public key coefficient, named inverse modulus.
Pointer to K3	A pointer to an optional RSA public key coefficient, named rBarr coefficient.

Hint Bar

Review the Flash Boot TRM chapter for additional details

Definition of Boot Protection Setting in SFlash

...
FUSE_WRITE_PU (16B)
N_FUSE_WRITE_PU (4B)
...
FUSE_READ_PU (16B)
N_FUSE_READ_PU (4B)
...
FLASH_WRITE_PU (16B)
N_FLASH_WRITE_PU (4B)
...
PPU Config. (1B)
PPU_ID (2B)
N_PPU (4B)
...
SMPU15 (16B)
N_SMPU (4B)
Object Size (4B)

Names	Description
FUSE_WRITE_PU	Data structure of FUSE_WRITE_PU
N_FUSE_WRITE_PU	Number of FUSE_WRITE_PUs stored in this object. It is followed by the contents of FUSE_WRITE_PUs
FUSE_READ_PU	Data structure of FUSE_READ_PU
N_FUSE_READ_PU	Number of FUSE_READ_PUs stored in this object. It is followed by the contents of FUSE_READ_PUs
FLASH_WRITE_PU	Data structure of FLASH_WRITE_PU
N_FLASH_WRITE_PU	Number of FLASH_WRITE_PUs stored in this object. It is followed by the contents of FLASH_WRITE_PUs
PPU_ID, PPU Config defines a PPU	PPU_ID is the PPU number (2 bytes) and PPU Config is described using 1 byte (4 bits for Write class and 4 bits for Read class)
N_PPU	Number of PPU structures stored in this object.
SMPU15	Contains SMPU region address and SMPU region attributes
N_SMPU	Number of SMPU structures (starting form SMPU15) stored in this object
Object Size	Size of boot protection object in bytes

Hint Bar

Review the BootROM and Flash Boot TRM chapters for additional details

Definition of TOC 2 in SFlash

› TOC2 is stored in SFlash and is used to configure flash boot and ROM boot firmware

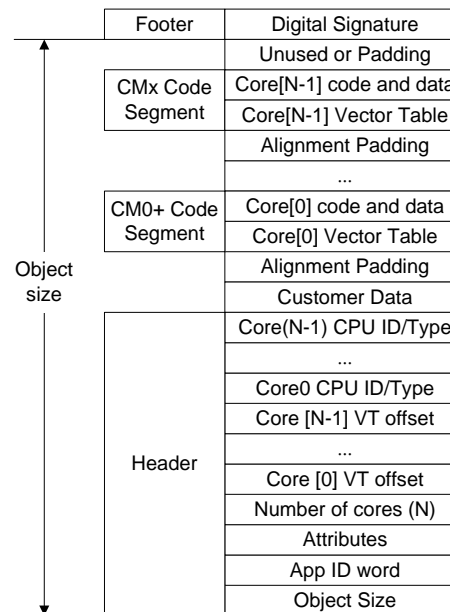
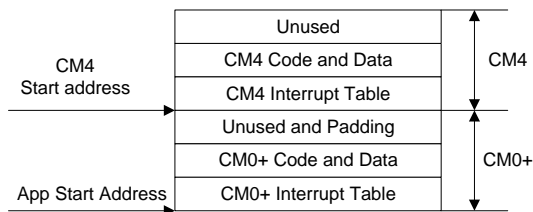
Offset	Names	Purpose
0x00	TOC2_OBJECT_SIZE	Object size in bytes starting from offset 0x00 until the last entry in TOC2.
0x04	TOC2_MAGIC_NUMBER	Magic number (0x01211220)
0x08	TOC2_SMIF_CFG_STRUCT_ADDR	Null terminated table of pointers representing the SMIF configuration structure.
0x0C	TOC2_FIRST_USER_APP_ADDR	Address of CM0+ First User Application Object (such as HSM in Traveo II)
0x10	TOC2_FIRST_USER_APP_FORMAT	First Application Object Format.
0x14	TOC2_SECOND_USER_APP_ADDR	Address of CM0+ Second User Application Object (0's if none)
0x18	TOC2_SECOND_USER_APP_FORMAT	Second Application Object Format
0x1C	TOC2_FIRST_CM4_0_USER_APP_ADDR	Address of CM4 core0 First User Application Object
0x20	TOC2_SECOND_CM4_0_USER_APP_ADDR	Address of CM4 core0 Second User Application Object
0x24	TOC2_FIRST_CM4_1_USER_APP_ADDR	Address of CM4 core1 First User Application Object
0x28	TOC2_SECOND_CM4_1_USER_APP_ADDR	Address of CM4 core1 Second User Application Object
0x100	TOC2_SHASH_OBJECTS	Number of additional objects (not including objects for FACTORY_HASH) starting from offset 0x104 to be verified for SECURE_HASH
0x104	TOC2_SIGNATURE_VERIF_KEY	Address of signature verification key (0 if none). The object is signature scheme specific. It is the public key in case of RSA.
0x108	TOC2_APP_PROTECTION_ADDR	Address of User SWPU object stored in SFlash.
0x1F8	TOC2_FLAGS	TOC2 configuration. If TOC2 is erased, Flash boot assumes TOC2_FLAGS = 0x0000_0242.

Hint Bar

Review the Flash Boot TRM chapter for additional details

Definition of Application Block in Code Flash

- > All core applications are encapsulated based on the following standard Cypress application formats
 - Basic Application Format (CyBAF)
 - CyBAF can be used only in VIRGIN and NORMAL protection modes
 - Cypress Secure Application Format (CySAF)
 - CySAF can be used in SECURE protection mode



Hint Bar

Review the Flash Boot TRM chapter for additional details



Part of your life. Part of tomorrow.

Revision History

Revision	ECN	Submission Data	Description of Change
**	6123648	04/06/2018	Initial release
*A	6323653	08/01/2018	Added page 2 Updated pages 3, 4, and 5
*B	6702818	10/16/2019	Updated page 2, 3, 4, 6, 7, 8, 9, 10, 14, 22 Delete CYT2B5 series Added page 5
*C	6659762	01/06/2020	Minor Change: Corrected revision in page 23, it reflects *A instead of *B in previous revision.
*D	6815787	02/10/2020	Page 7: Updated the protection table, 11 (merged with 12) Added page 12 Updated pages 3 to 21 (Minor changes)
*E	7048389	12/19/2020	Updated pages 2, 3, 4, 5.