Customer Training Workshop

Traveo™ II Boot

Q2 2021

Infineon

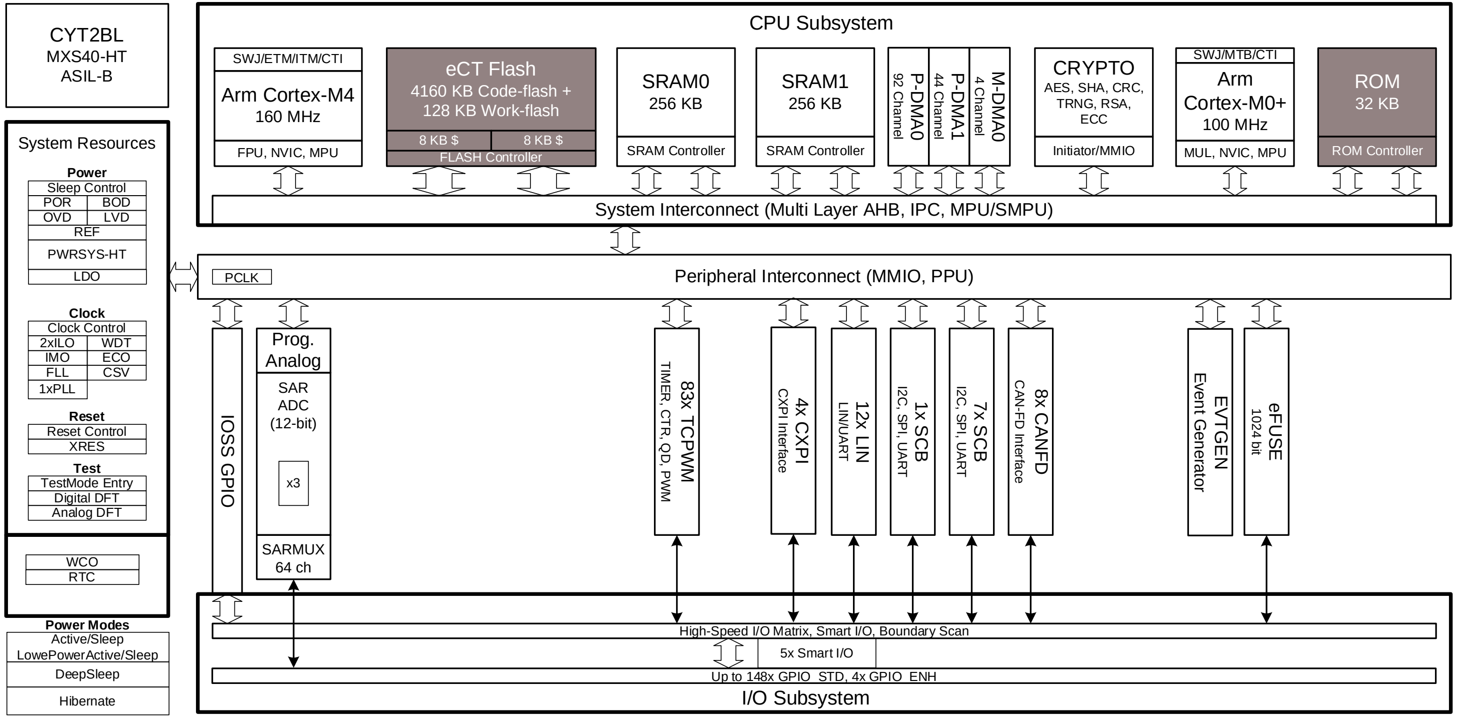
Target Products

- 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



CYT2BL
MXS40-HT
ASIL-B

System Resources

Power
Sleep Control
POR BOD
OVD LVD
REF
PWRSYS-HT
LDO

Clock
Clock Control
2xILO WDT
IMO ECO
FLL CSV
1xPLL

Reset
Reset Control
XRES

Test
TestMode Entry
Digital DFT
Analog DFT

WCO
RTC

Power Modes
Active/Sleep
LowPowerActive/Sleep
DeepSleep
Hibernate

CPU Subsystem

SWJ/ETM/ITM/CTI
Arm Cortex-M4
160 MHz
FPU, NVIC, MPU

eCT Flash
4160 KB Code-flash +
128 KB Work-flash
8 KB $ 8 KB $
FLASH Controller

SRAM0
256 KB
SRAM Controller

SRAM1
256 KB
SRAM Controller

P-DMA0
92 Channel

P-DMA1
44 Channel

M-DMA0
4 Channel

CRYPTO
AES, SHA, CRC, TRNG, RSA, ECC
Initiator/MMIO

SWJ/MTB/CTI
Arm
Cortex-M0+
100 MHz
MUL, NVIC, MPU

ROM
32 KB
ROM Controller

System Interconnect (Multi Layer AHB, IPC, MPU/SMPU)

PCLK
Peripheral Interconnect (MMIO, PPU)

IOSS GPIO

Prog.
Analog
SAR
ADC
(12-bit)
x3
SARMUX
64 ch

83x TCPWM
TIMER, CTR, QD, PWM

4x CXPI
CXPI Interface

12x LIN
LIN/UART

1x SCB
I2C, SPI, UART

7x SCB
I2C, SPI, UART

8x CANFD
CAN-FD Interface

EVTGEN
Event Generator

eFUSE
1024 bit

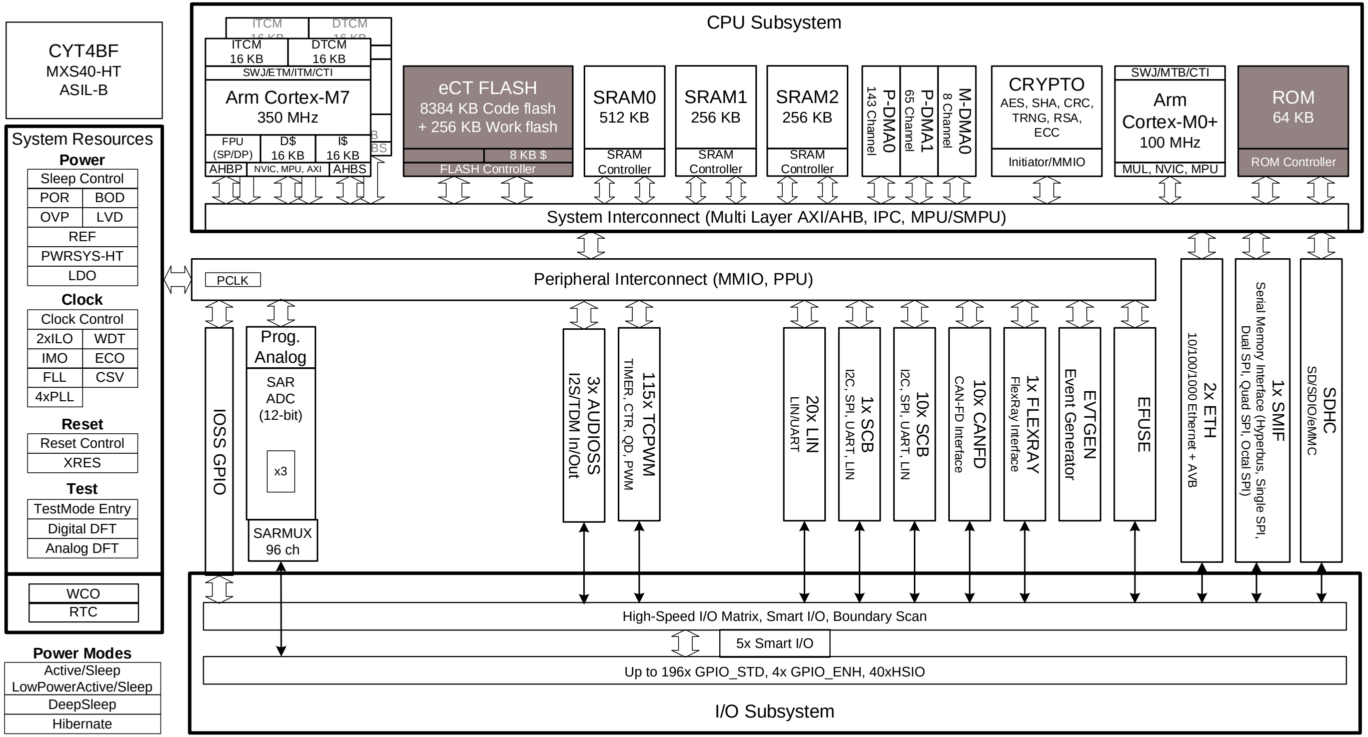
High-Speed I/O Matrix, Smart I/O, Boundary Scan
5x Smart I/O
Up to 148x GPIO_STD, 4x GPIO_ENH
I/O Subsystem

Hint Bar

Review TRM chapters 11 and 34 for additional details

Introduction to Traveo II Body Controller High

- Boot is part of the CPU subsystem



CYT4BF
MXS40-HT
ASIL-B

System Resources

Power: Sleep Control, POR, BOD, OVP, LVD, REF, PWRSYS-HT, LDO

Clock: Clock Control, 2xILO, WDT, IMO, ECO, FLL, CSV, 4xPLL

Reset: Reset Control, XRES

Test: TestMode Entry, Digital DFT, Analog DFT

WCO, RTC

Power Modes: Active/Sleep, LowPowerActive/Sleep, DeepSleep, Hibernate

CPU Subsystem

ITCM 16 KB, DTCM 16 KB, SWJ/ETM/ITM/CTI, Arm Cortex-M7 350 MHz, FPU (SP/DP), D$ 16 KB, I$ 16 KB, AHBP, NVIC, MPU, AXI, AHBS

eCT FLASH: 8384 KB Code flash + 256 KB Work flash, 8 KB $, FLASH Controller

SRAM0 512 KB, SRAM Controller

SRAM1 256 KB, SRAM Controller

SRAM2 256 KB, SRAM Controller

P-DMA0 143 Channel

P-DMA1 65 Channel

M-DMA0 8 Channel

CRYPTO: AES, SHA, CRC, TRNG, RSA, ECC, Initiator/MMIO

SWJ/MTB/CTI, Arm Cortex-M0+ 100 MHz, MUL, NVIC, MPU

ROM 64 KB, ROM Controller

System Interconnect (Multi Layer AXI/AHB, IPC, MPU/SMPU)

Peripheral Interconnect (MMIO, PPU)

PCLK

IOSS GPIO

Prog. Analog: SAR ADC (12-bit), x3

SARMUX 96 ch

3x AUDIOSS: I2S/TDM In/Out

115x TCPWM: TIMER, CTR, QD, PWM

20x LIN: LIN/UART

1x SCB: I2C, SPI, UART, LIN

10x SCB: I2C, SPI, UART, LIN

10x CANFD: CAN-FD Interface

1x FLEXRAY: FlexRay Interface

EVTGEN: Event Generator

EFUSE

2x ETH: 10/100/1000 Ethernet + AVB

1x SMIF: Serial Memory Interface (Hyperbus, Single SPI, Dual SPI, Quad SPI, Octal SPI)

SDHC: SD/SDIO/eMMC

High-Speed I/O Matrix, Smart I/O, Boundary Scan

5x Smart I/O

Up to 196x GPIO_STD, 4x GPIO_ENH, 40xHSIO

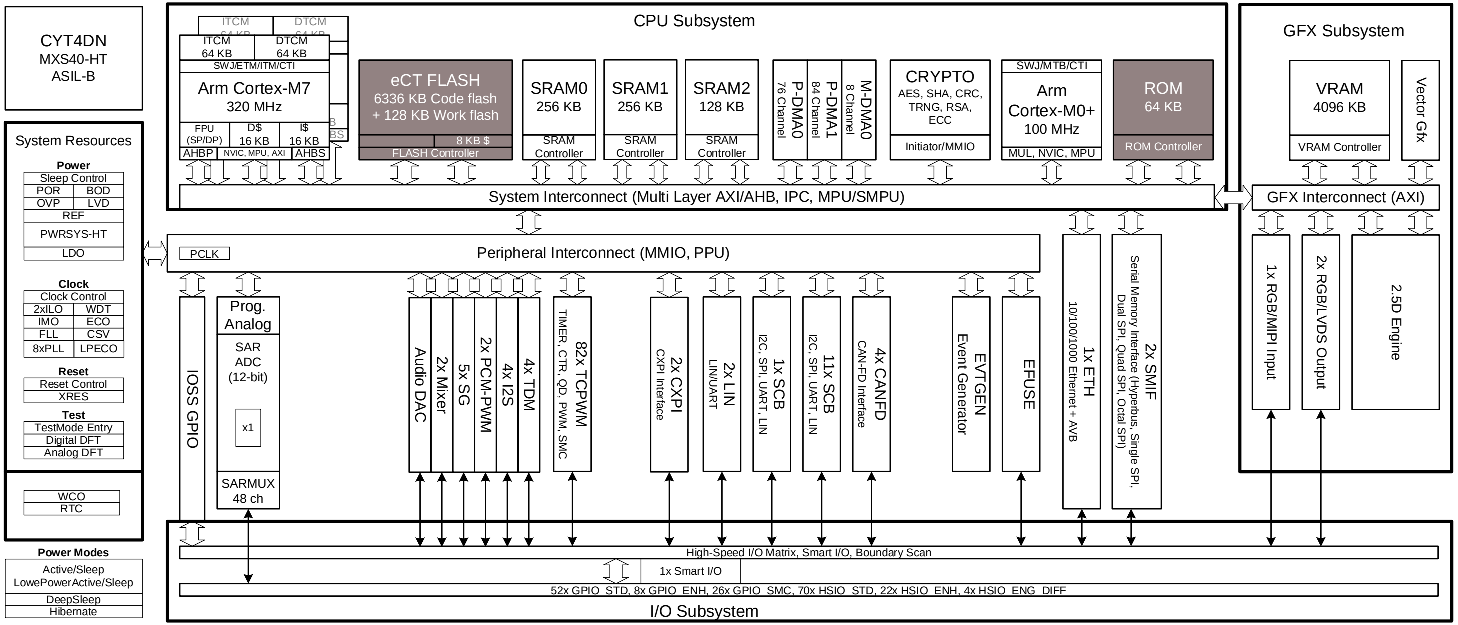
I/O Subsystem

Hint Bar

Review TRM chapters 11 and 38 for additional details

Introduction to Traveo II Cluster

› Boot is part of the CPU subsystem

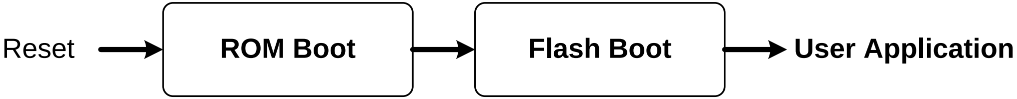


Hint Bar

Review TRM chapters 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 DAP access
  - Enables system calls



Hint Bar

Review the BootROM and Flash Boot TRM chapters for additional details

Review the datasheet for ROM and Flash boot time

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)

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	Detects corruption/error

(D) Configures the protection unit (MPU, SMPU, 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 the BootROM TRM chapter 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

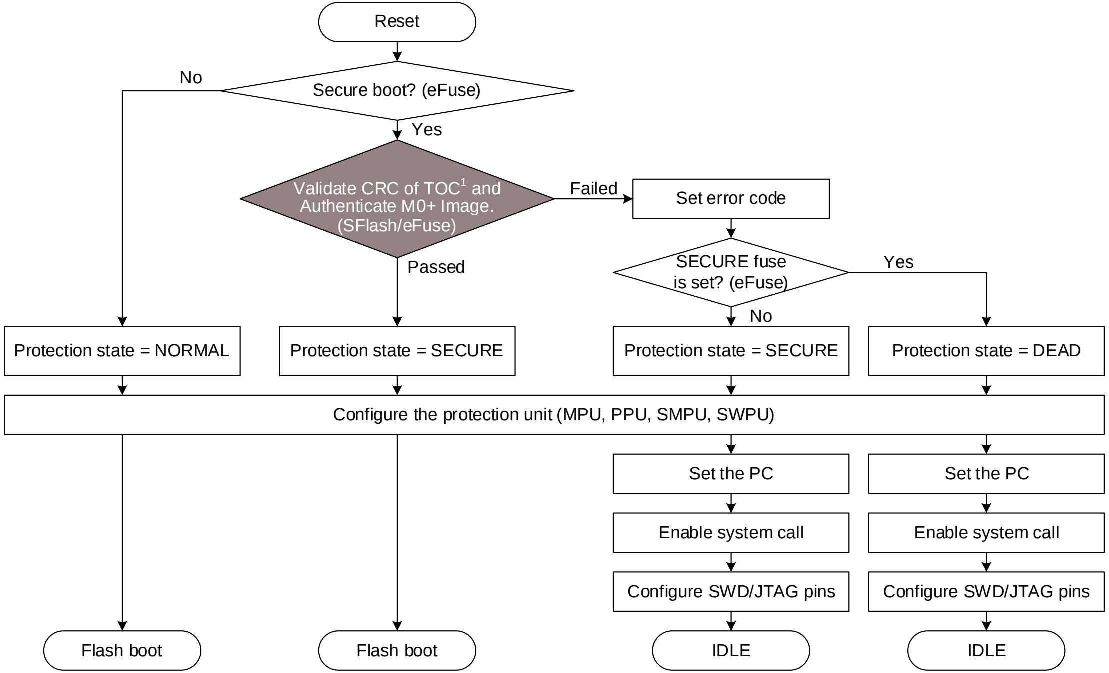
(A) Verifies the secure boot with eFuse info

(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 the next process



Reset
Secure boot? (eFuse)
No
Yes
Validate CRC of TOC[1] and Authenticate M0+ Image. (SFlash/eFuse)
Failed
Set error code
Passed
SECURE fuse is set? (eFuse)
Yes
No
Protection state = NORMAL
Protection state = SECURE
Protection state = SECURE
Protection state = DEAD
Configure the protection unit (MPU, PPU, SMPU, SWPU)
Set the PC
Set the PC
Enable system call
Enable system call
Configure SWD/JTAG pins
Configure SWD/JTAG pins
Flash boot
Flash boot
IDLE
IDLE

Hint Bar

Review the BootROM TRM chapter 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 (TOC1) set at the factory and a Part2 object (TOC2) set by the user
- TOC1: This is an object for device protection setting
- TOC2: This is an object for the digital signature scheme

Flash Boot

- Operation
  - (A) Validates the TOC2 and secures image application[1] 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

(A) Validates the TOC2 and secure image application in Flash

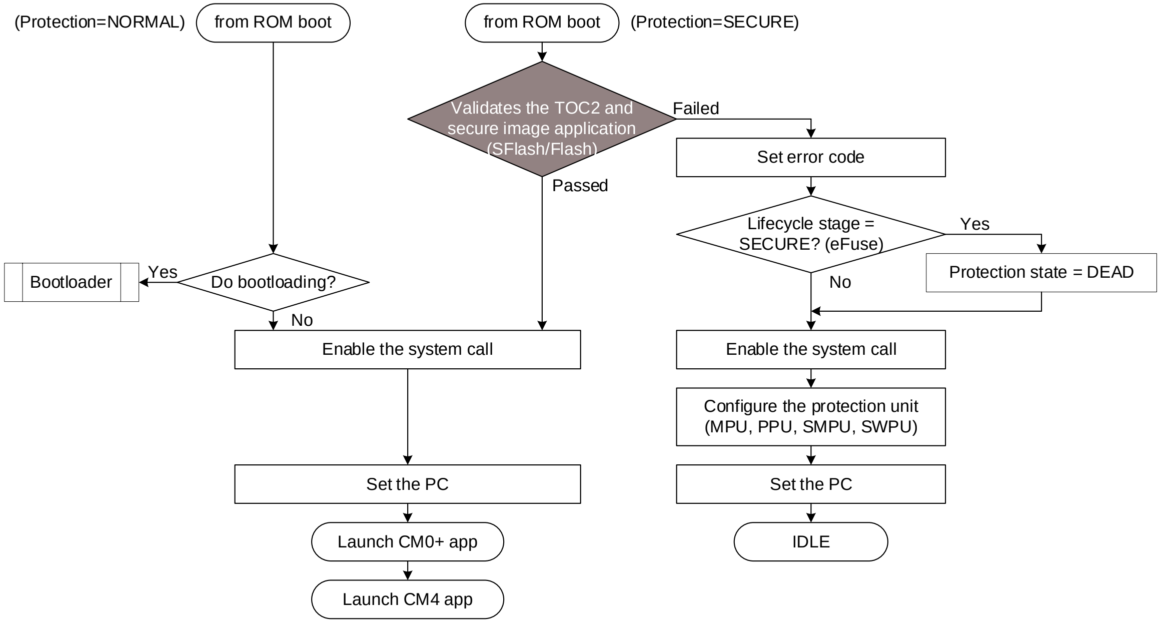
(B) Enables the system calls

(C) Enables the system calls

(D) Configures the protection unit

(E) Sets the PC for all masters

(F) Launches the CM0+ and CM4 applications



(Protection=NORMAL)

from ROM boot

Do bootloading?

Yes

Bootloader

No

Enable the system call

Set the PC

Launch CM0+ app

Launch CM4 app

from ROM boot

(Protection=SECURE)

Validates the TOC2 and secure image application (SFlash/Flash)

Failed

Passed

Set error code

Lifecycle stage = SECURE? (eFuse)

Yes

Protection state = DEAD

No

Enable the system call

Configure the protection unit (MPU, PPU, SMPU, SWPU)

Set the PC

IDLE

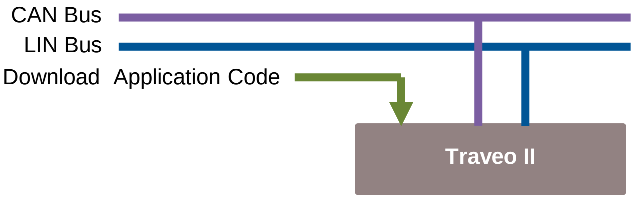
Hint Bar

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



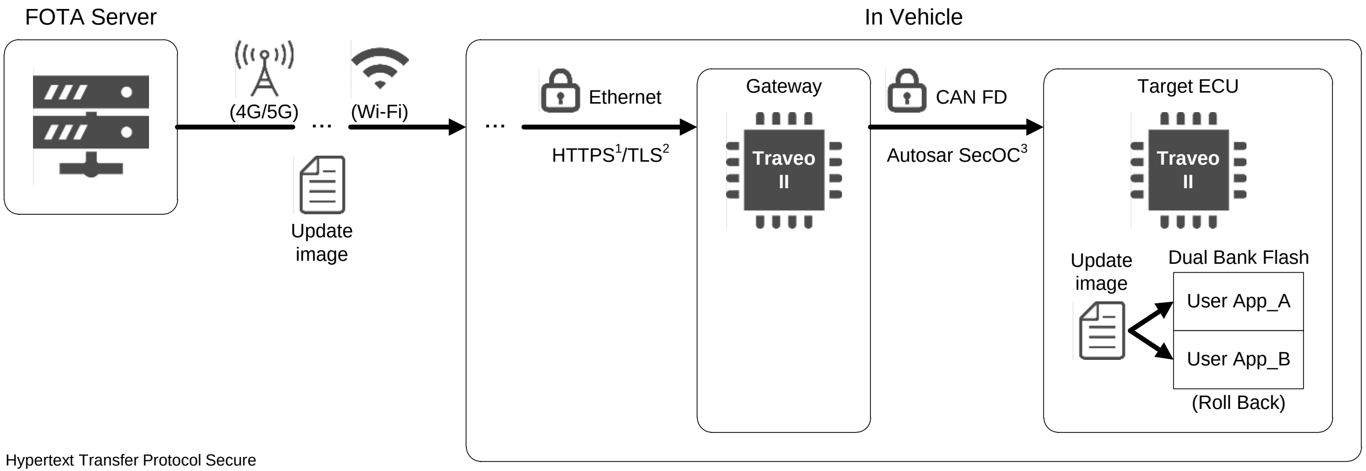
- Bootloader Activation Conditions
  - The internal bootloader will activate if all these conditions are met:
    - Two words at the start of flash must be 0xFFFF_FFFF
    - TOC2 is valid and TOC2_FLAGS bit FB_BOOTLOADER_DISABLE should be 2'b01 (default). Otherwise, TOC2 is erased
    - Protection mode 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



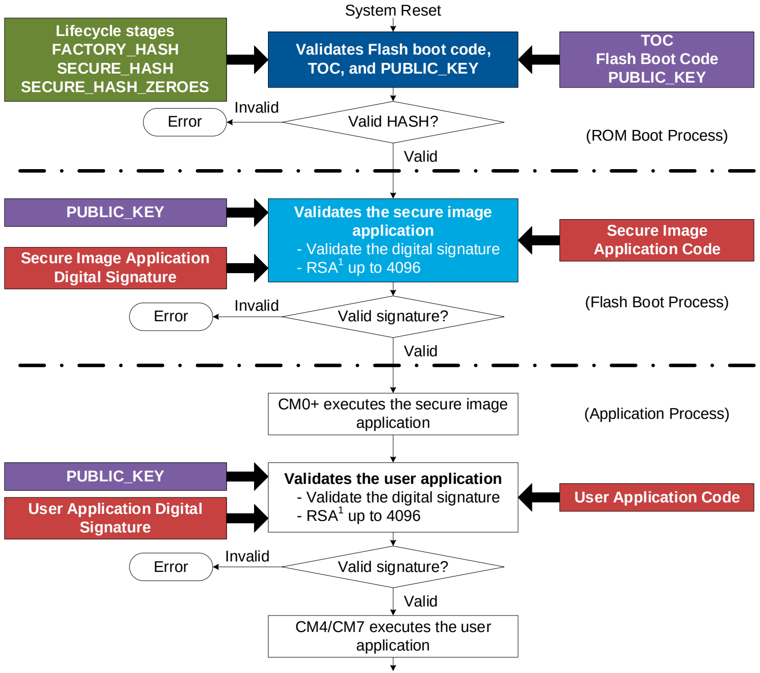
Hint Bar

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



System Reset

Lifecycle stages
FACTORY_HASH
SECURE_HASH
SECURE_HASH_ZEROES

Validates Flash boot code, TOC, and PUBLIC_KEY

TOC
Flash Boot Code
PUBLIC_KEY

Error

Invalid

Valid HASH?

Valid

(ROM Boot Process)

PUBLIC_KEY

Secure Image Application Digital Signature

Validates the secure image application
- Validate the digital signature
- RSA[1] up to 4096

Secure Image Application Code

Error

Invalid

Valid signature?

Valid

(Flash Boot Process)

CM0+ executes the secure image application

(Application Process)

PUBLIC_KEY

User Application Digital Signature

Validates the user application
- Validate the digital signature
- RSA[1] up to 4096

User Application Code

Error

Invalid

Valid signature?

Valid

CM4/CM7 executes the user application

The following colors indicate where objects are stored.

🟪 SFlash objects

🟥 Flash objects

🟩 eFuse objects

The following colors indicate where objects are executed.

🟦 ROM boot

🟦 Flash boot

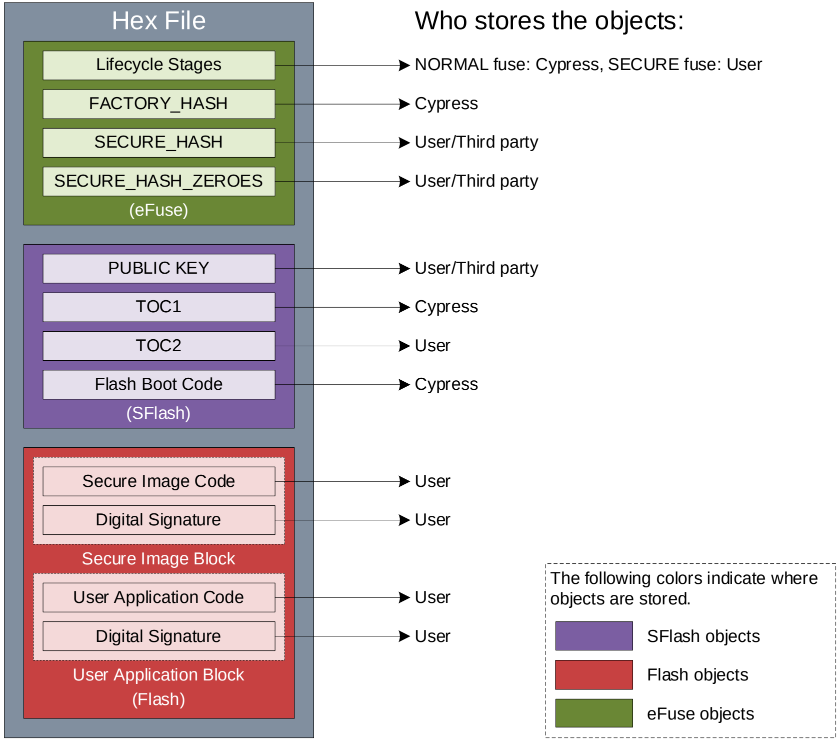
Hint Bar

Review the BootROM and Flash Boot TRM chapters for additional details

[1] For RSA 2K/3K/4K support, see the device-specific datasheet (under the section Part Number/Ordering Code Nomenclature, Hardware option).

Secure System Hex File

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



Hex File

Who stores the objects:

Lifecycle Stages → NORMAL fuse: Cypress, SECURE fuse: User

FACTORY_HASH → Cypress

SECURE_HASH → User/Third party

SECURE_HASH_ZEROES → User/Third party

(eFuse)

PUBLIC KEY → User/Third party

TOC1 → Cypress

TOC2 → User

Flash Boot Code → Cypress

(SFlash)

Secure Image Code → User

Digital Signature → User

Secure Image Block

User Application Code → User

Digital Signature → User

User Application Block

(Flash)

The following colors indicate where objects are stored.

🟪 SFlash objects

🟥 Flash objects

🟩 eFuse objects

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	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	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	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	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	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)

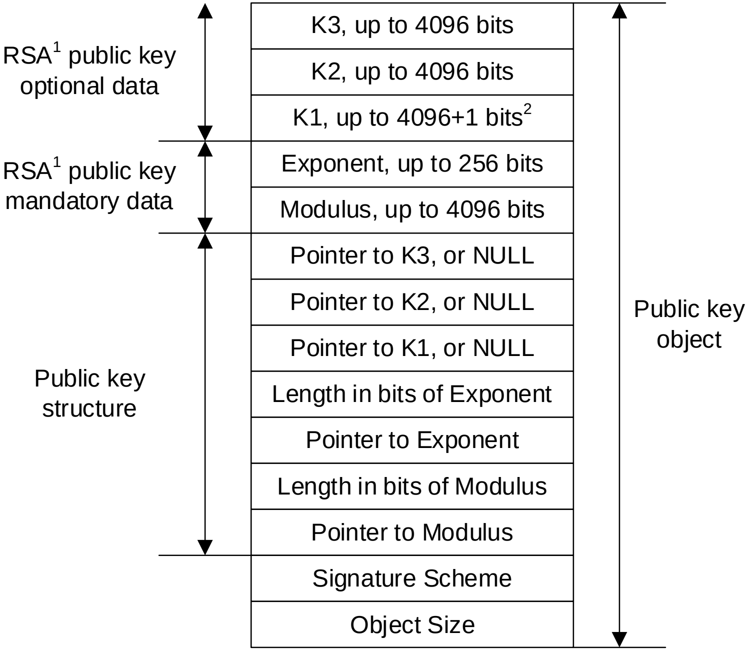
Names		Bits	Description
SECURE Access Restrictions	SMIF_XIP_ENABLE	20	Indicates what portion of SMIF_XIP is accessible through the system access port
DEAD Access Restrictions	<Same as SECURE Access Restrictions>		The structure is identical to the one above but used when entering DEAD mode. It assumes that this structure is more restrictive than SECURE
Critical Object Hash	FACTORY_HASH		SHA-256 (upper 128 bits) that covers objects in TOC1. 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 TOC1 and TOC2. 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 diagram shows the key object structure used for signature verification



RSA[1] public key optional data

RSA[1] public key mandatory data

Public key structure

K3, up to 4096 bits

K2, up to 4096 bits

K1, up to 4096+1 bits[2]

Exponent, up to 256 bits

Modulus, up to 4096 bits

Pointer to K3, or NULL

Pointer to K2, or NULL

Pointer to K1, or NULL

Length in bits of Exponent

Pointer to Exponent

Length in bits of Modulus

Pointer to Modulus

Signature Scheme

Object Size

Public key object

Hint Bar

Review the Flash Boot TRM chapter for additional details

[1] For RSA 2K/3K/4K support, see the device-specific datasheet (under the section Part Number/Ordering Code Nomenclature, Hardware option).

[2] 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 public key data protection
Signature Scheme	A signature scheme 0 - RSASSA-PKCS1-v1.5 with RSA up to 4096 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 TOC2 in SFlash

- TOC2 is stored in SFlash and is used to configure Flash boot and ROM boot firmware

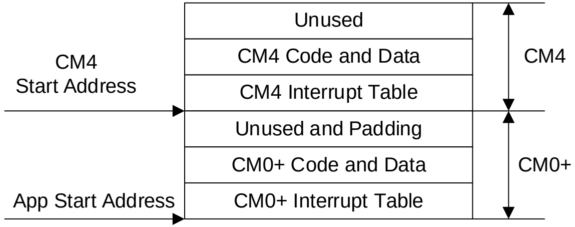
Offset	Names	Purpose
0x00	TOC2_OBJECT_SIZE	Object size in bytes starting from offset 0x00 until the last entry in TOC2
0x04	TOC2_MAGIC_NUMBER	Magic number (0x01211220)
0x08	TOC2_SMIF_CFG_STRUCT_ADDR	Null terminated table of pointers representing the SMIF configuration structure
0x0C	TOC2_FIRST_USER_APP_ADDR	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

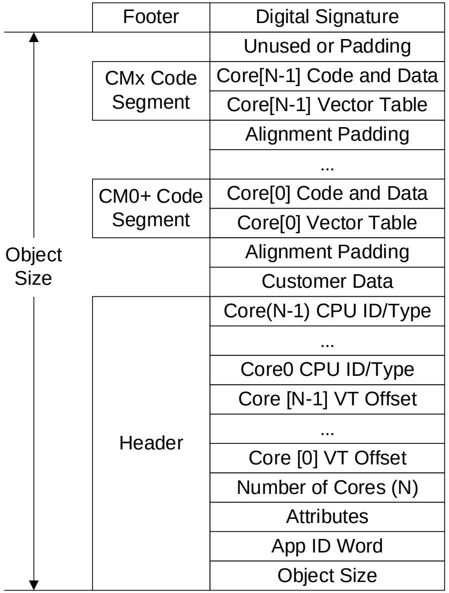
Review the Flash Boot TRM chapter for additional details

Definition of Application Block in Code Flash

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



Address	Layout	Region
	Unused	CM4
	CM4 Code and Data	
CM4 Start Address	CM4 Interrupt Table	
	Unused and Padding	CM0+
	CM0+ Code and Data	
App Start Address	CM0+ Interrupt Table	



	Segment	Layout
	Footer	Digital Signature
Object Size		Unused or Padding
	CMx Code Segment	Core[N-1] Code and Data
		Core[N-1] Vector Table
		Alignment Padding
		...
	CM0+ Code Segment	Core[0] Code and Data
		Core[0] Vector Table
		Alignment Padding
		Customer Data
	Header	Core(N-1) CPU ID/Type
		...
		Core0 CPU ID/Type
		Core [N-1] VT Offset
		...
		Core [0] VT Offset
		Number of Cores (N)
		Attributes
		App ID Word
		Object Size

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.
*F	7098118	02/24/2021	Updated the RSA support size (pages 13, 18, 19)