S32K1XX – AN OTA SOLUTION FOR AUTOMOTIVE EDGE NODES

OSVALDO ROMERO
JUAN ROMERO

SYSTEMS AND APPLICATIONS ENGINEERING

AMF-AUT-T2815 | AUGUST 2017

SECURE CONNECTIONS FOR A SMARTER WORLD

AGENDA

01. FOTA Overview
02. Secure firmware updates
03. S32K1xx capabilities
04. S32K144 use case
05. Demo
06. Conclusions

01.

FOTA Overview

Today: 90% of Auto Innovation via electronics

NXP is #1

#1 INFOTAINMENT
TUNERS
SOFTWARE-DEFINED DIGITAL RADIO
MULTIMEDIA PROCESSORS
SOUND SYSTEM DSPs & AMPLIFIERS
NFC BT PAIRING
WIRELESS POWER CHARGING
POWER MANAGEMENT

ADAS & SECURITY

POWERTRAIN & CHASSIS
MICROCONTOLLERS
PRESSURE/ MOTION SENSORS
BATTERY MANAGEMENT
DRIVERS

STANDARD PRODUCTS
LOGIC
POWER
DISCRETES

#1 SECURE CAR ACCESS
IMMOBILIZER/ SECURITY
REMOTE KEYLESS ENTRY
PASSIVE KEYLESS ENTRY/ GO
BI-DIRECTIONAL KEYS
NFC
ULTRA WIDE BAND

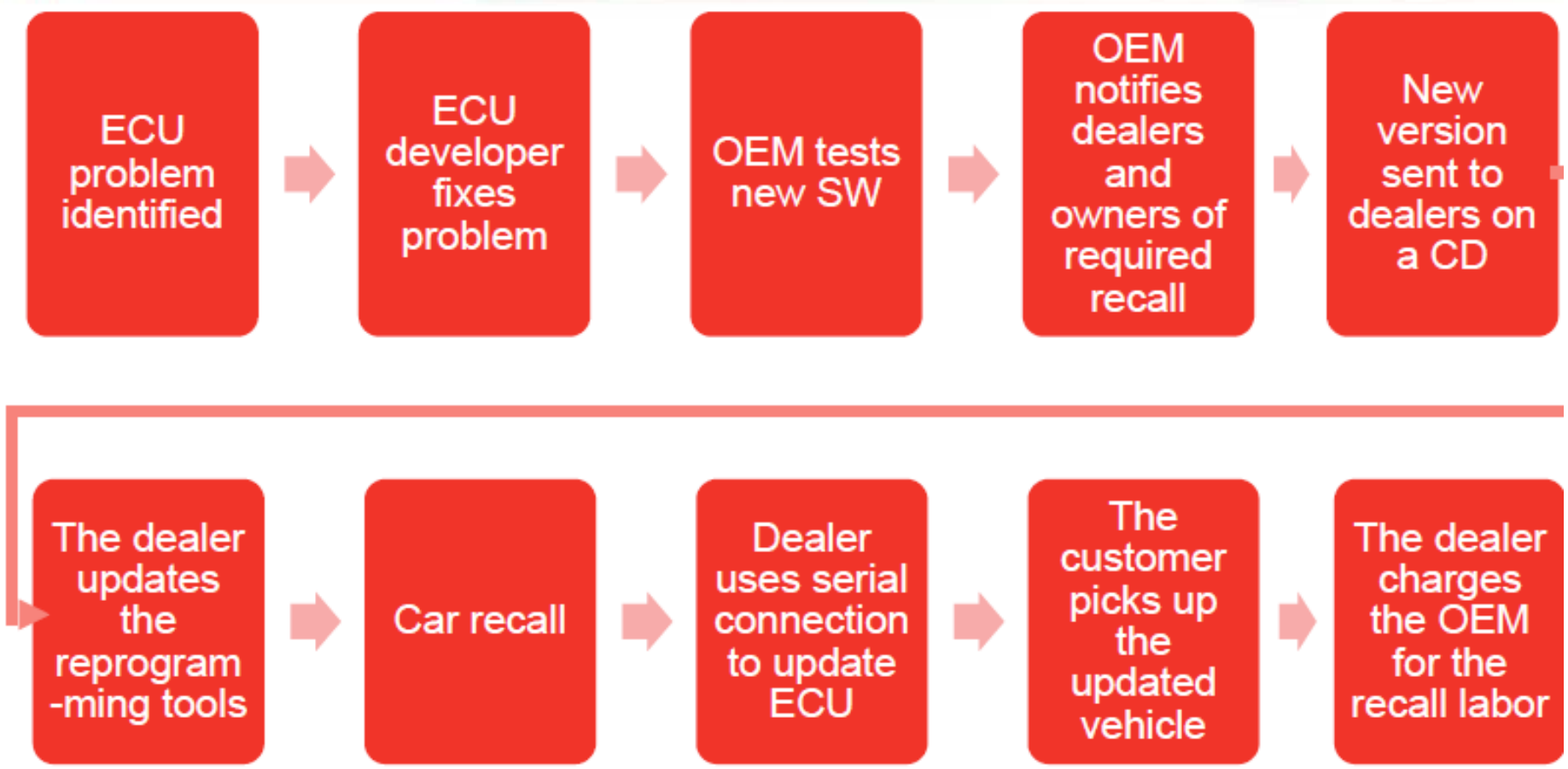
#1 VEHICLE NETWORKING
CAN/LIN/ FLEXRAY
ETHERNET
CENTRAL GATEWAY CONTROLLER
SECURITY
RF

#1 BODY
MICROCONTROLLERS
POSITION/ ANGLE SENSORS
SYSTEM BASIS CHIPS

#1 SAFETY
MICROCONTROLLERS AIRBAG
ANALOG AIRBAG
MICROCONTROLLERS BRAKING
ANALOG BRAKING
SENSORS BRAKING
TIRE PRESSURE MONITORING

#1 Auto Analog/ RF | **#1 Auto MCU (ex JPN)** | **#1 Auto Merchant MEMS Sensors**

FOTA Overview: **Common Recall Process**

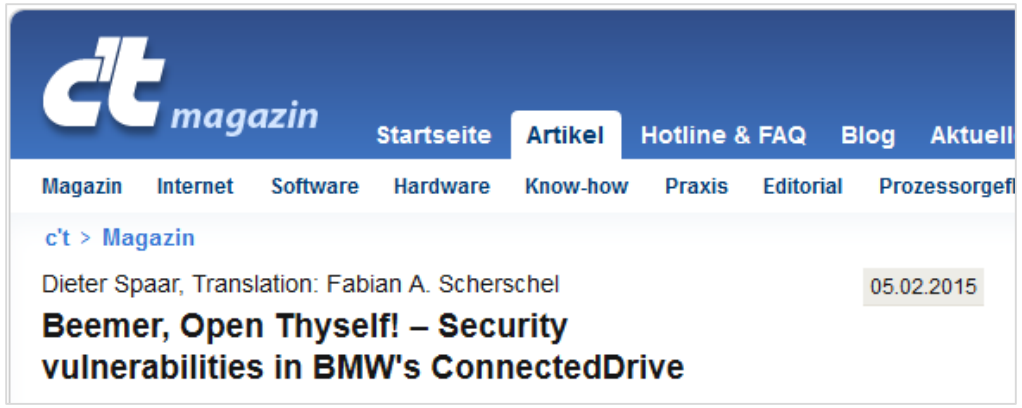


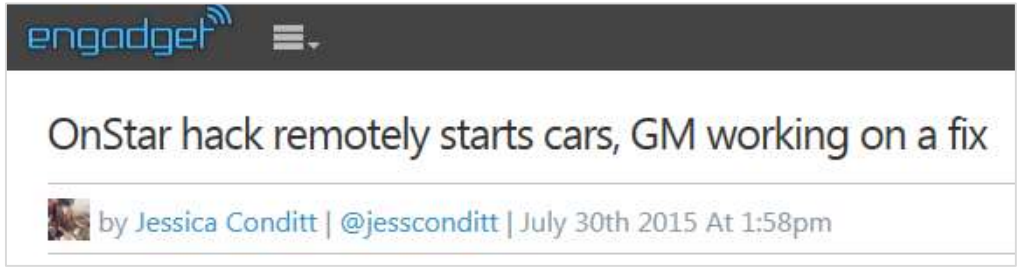
Vector and Redbend (2014).*Update ECUs using Delta- and Over-the-Air-Technology* [PDF Slides]. Retrieved April 22, 2016 from https://vector.com/portal/medien/cmc/events/Webinars/2014/Vector_RedBend_Webinar_Flashing_over_the_air_and_delta_technology_20140121_EN.pdf

FOTA Overview: **Motivations**

- Increasing number of recalls
- Dealer update $
- As firmware complexity increases, the probability of required firmware updates also increases
- User convenience vs going to dealer
- Safety can be improved with quicker updates

Car Hacking is ‘Hot’





… is an Attractive Target for Hackers!

Valuable Data

- Collection of data/info
- Storage of data
- Diagnostic functions

Protect Privacy

High Vulnerability

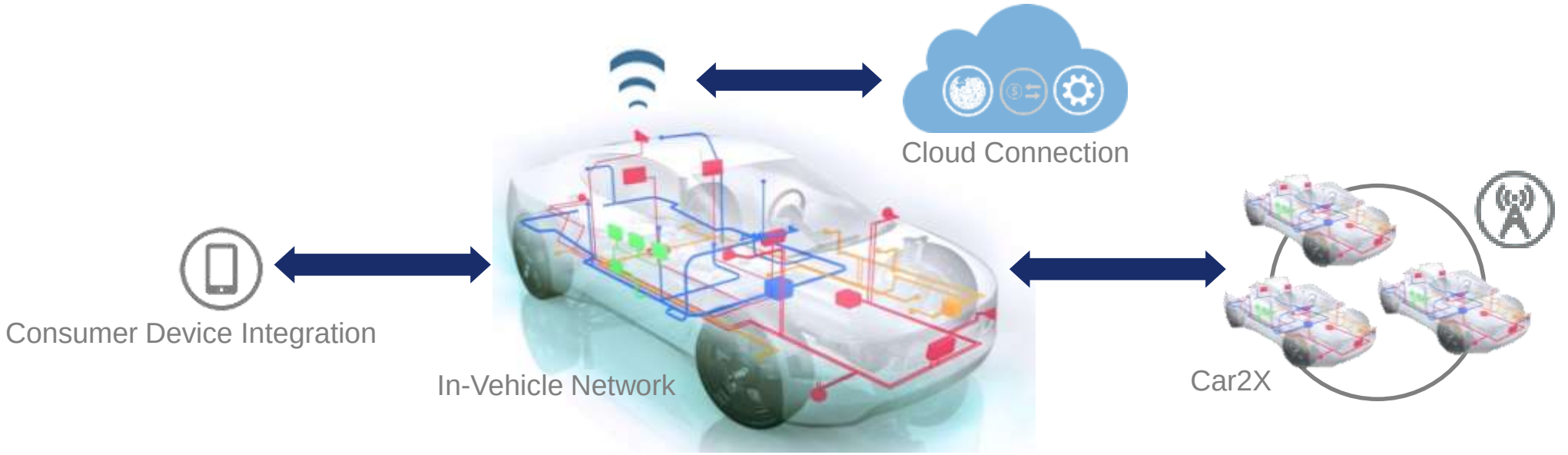
- Increasing number of nodes
- More advanced features
- X-by-Wire

Increase Safety

Easy (Remote) Access

- Fully Connected Car
- External & internal interfaces
- Wired & wireless interfaces

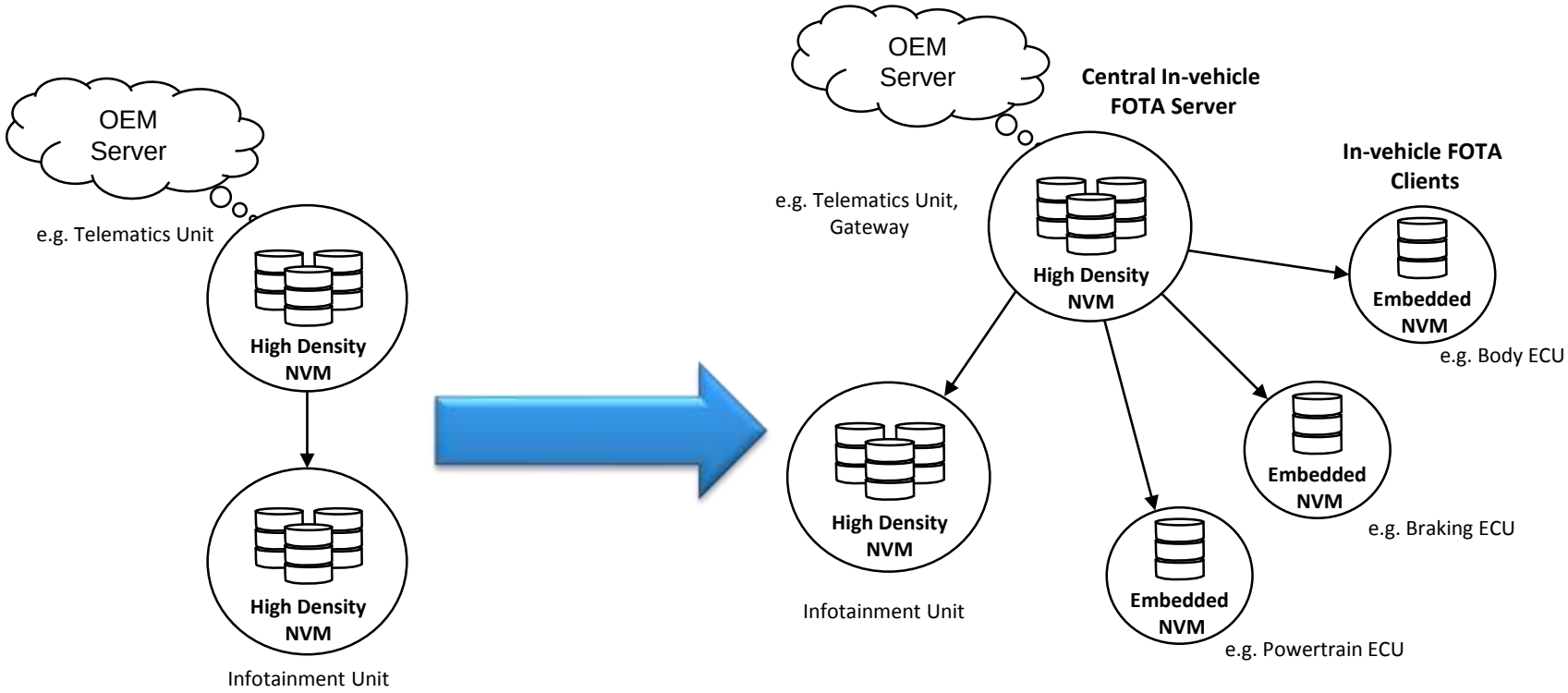
Prevent Unauthorized Access



02.

Secure FW Updates

FOTA Overview: MOVING DEEPER INTO THE VEHICLE



FOTA Update of Infotainment & Telematics Systems

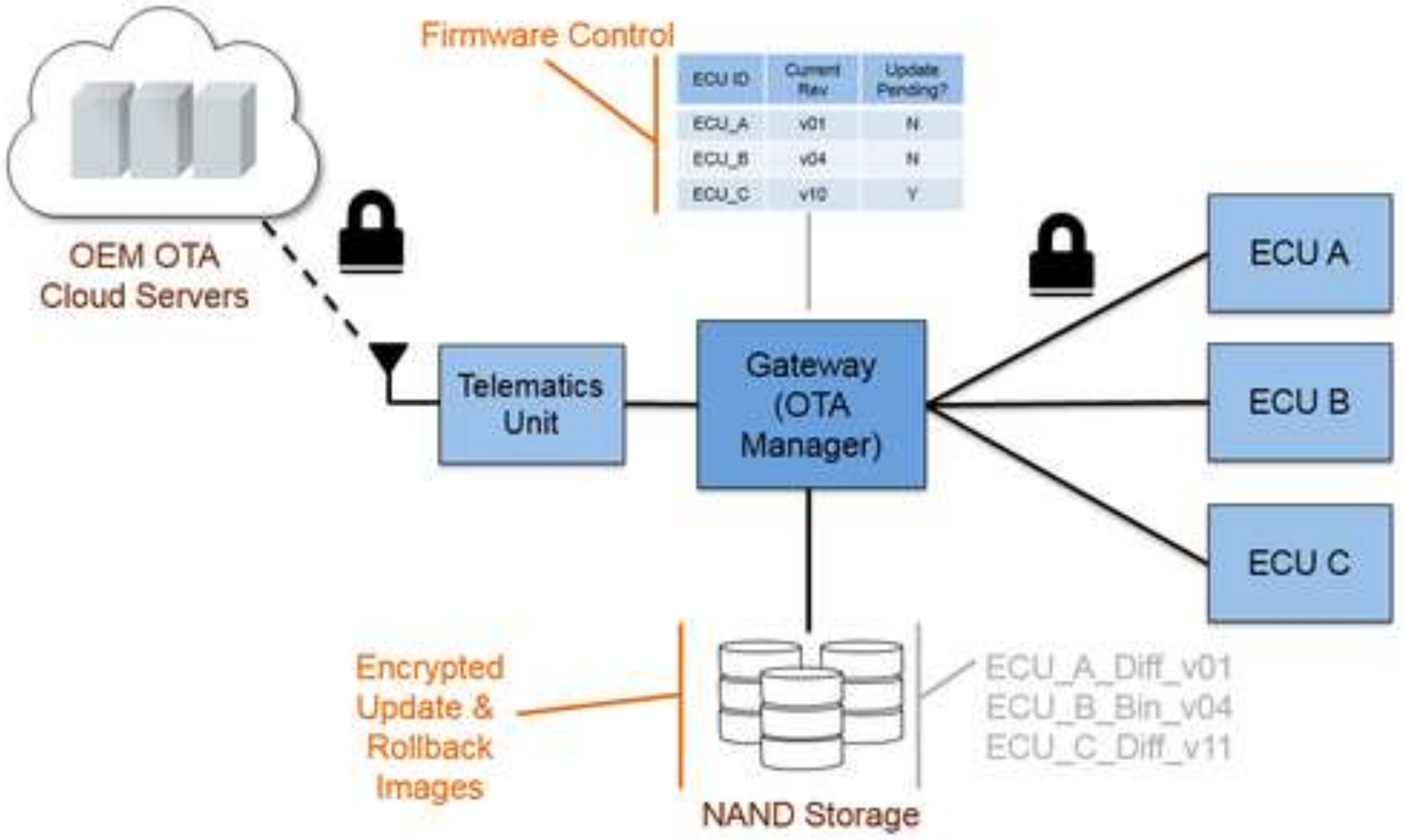
Focused on updates to software on the infotainment & telematics, but not propagating further into the vehicle architecture

FOTA Update of Major ECUs within the vehicle

New Challenges with this architecture:

- Security throughout
- Cost sensitivity of embedded ECUs
- Embedded NVM vs High Density NVM
- Strategy of when & what to update

FOTA Overview: **Moving Deeper in the Vehicle**



FOTA Overview: **In Place Use Case**

Reset vectors to bootloader, which is never erased

Advantages:

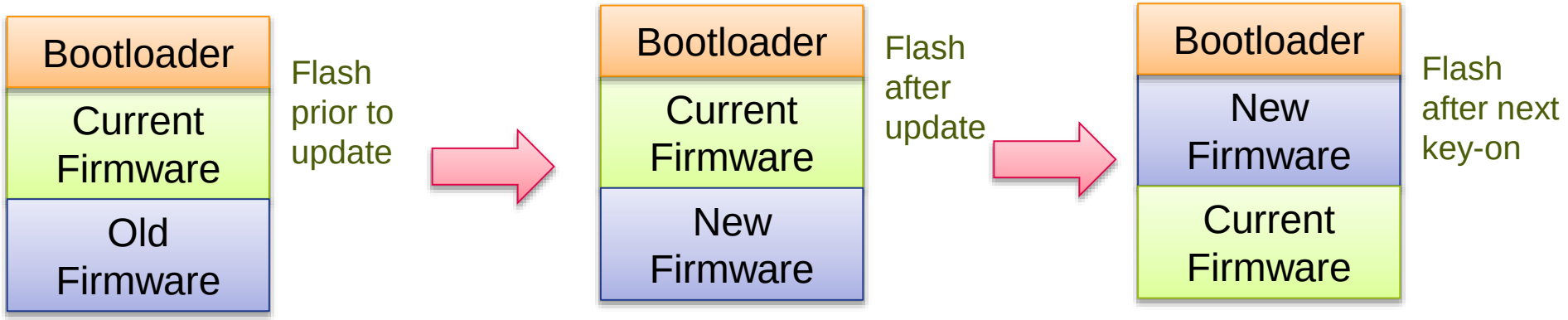
- No need for additional flash

Disadvantage:

- Requires vehicle downtime during update process
- Not possible to instantly “roll-back” if an issue occurs

FOTA Overview: **A/B Swap Use Case**

Reset vectors to bootloader, which is never erased



Advantages:

- Update can be carried out while current application is actively running from flash
- Always have original firmware to roll back to in case of issue
- Vehicle is always available – guaranteed no vehicle downtime regardless of update errors

Disadvantage:

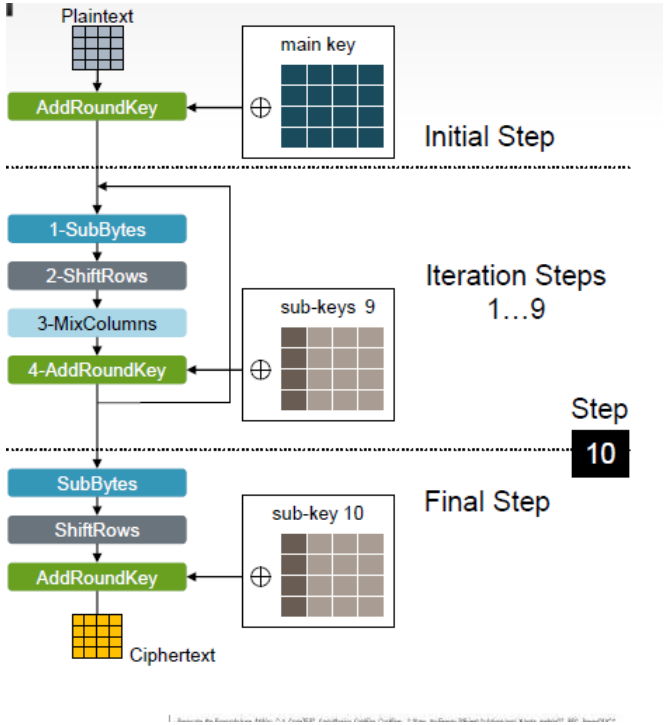
- Requires ~2x flash application storage
- Requires RWW capabilities
- SW remapping is required

FOTA Overview: Assumptions

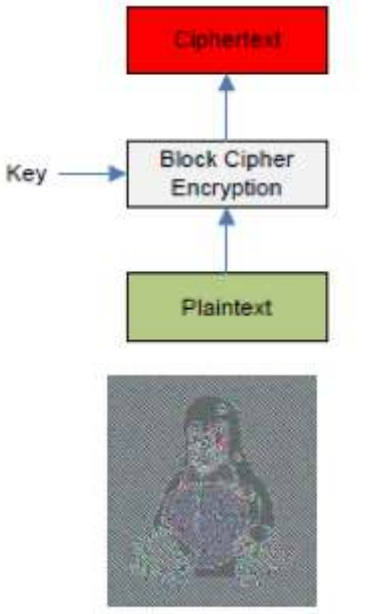
- End node:
  - gets partial or full image for flashing
  - will have at least enough spare erased flash for a full image
  - receives updated software over serial link
  - has boot block which never changes with OTA updates
- Best case: update is performed while running existing software
- Before new firmware becomes active, application/boot firmware can perform:
  - Security validation
  - Functional validation
- New firmware starts on reset following the update completion

Security Overview: **AES 128**

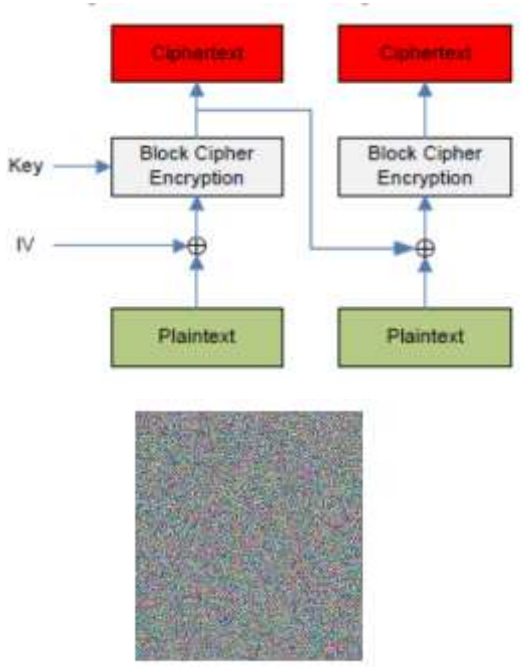
- Crypto and decryption algorithm: AES-128
- AES Encryption/Decryption in ECB or CBC mode



Plaintext



ECB:Ciphertext



CBC:Ciphertext

Security Overview: **CMAC Generator**

- Cipher based Message Authentication Code (CMAC)
- A MAC algorithm inputs:
  - Secret key
  - Message of arbitrary length
- A MAC algorithm output:
  - MAC value
- The MAC value protects both a message's data integrity as well as its authenticity.

03.

S32K1xx Capabilities

S32K14x: **Block Diagram**

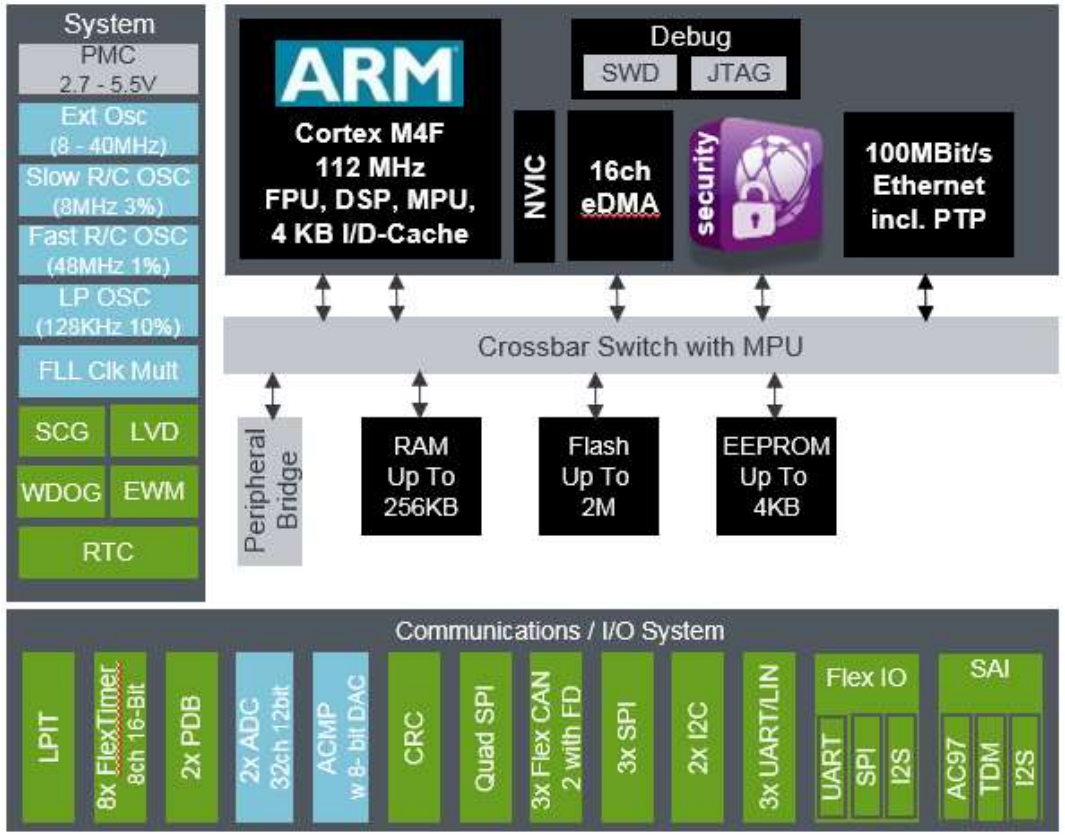
High performance
- ARM Cortex M4F up to 112MHz w FPU
- eDMA from 57xxx family

Software Friendly Architecture
- High RAM to Flash ratio
- Independent CPU and peripheral clocking
- 48MHz 1% IRC – no PLL init required in LP
- Registers maintained in all modes
- Programmable triggers for ADC → no SW delay counters or extra interrupts

Functional safety
- ISO26262 support for ASIL B or higher
- Memory Protection Unit
- ECC on 512K Flash / 64K Dataflash and RAM
- Independent internal OSC for Watchdog
- Diversity between ADC and ACMP
- Diversity between SPI/SCI and FlexIO
- Core self test libraries
- Scalable LVD protection
- CRC

Low power
- Low leakage technology
- Multiple VLP modes and IRC combos
- Wake-up on analog thresholds

Security
- CSEc (SHE-spec)

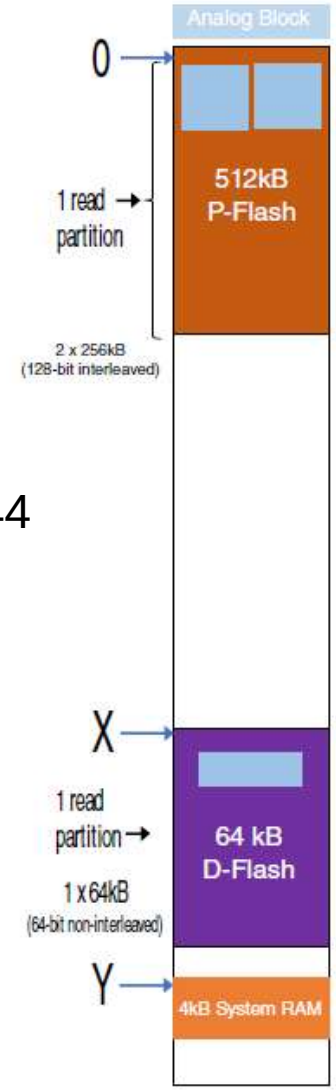


Packages & IO
- Open-drain for 3.3 V and hi-drive pins
- Powered ESD protection
- Packages: 100 BGA, 64 LQFP, 100 LQFP

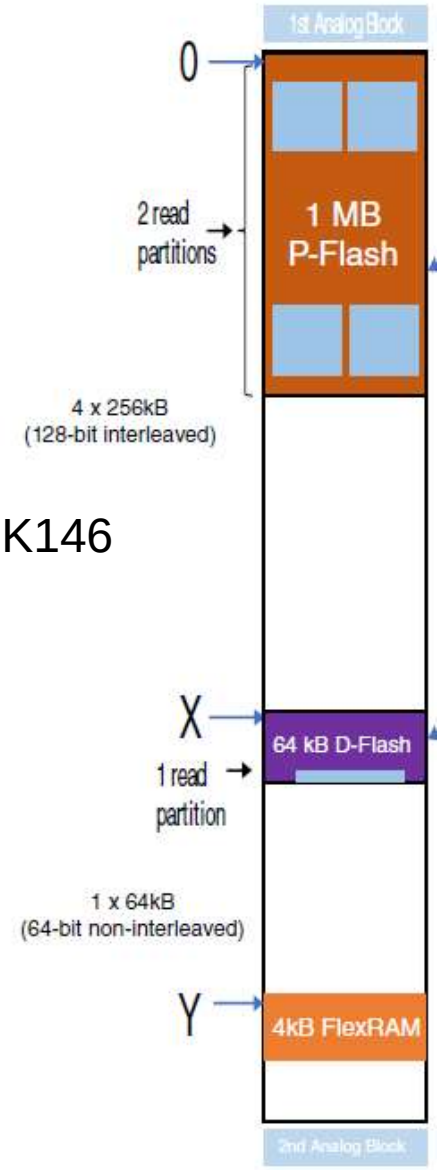
Operating Characteristics
- Voltage range: 2.7V to 5.5V
- Temperature (ambient): -40°C to +125°C

S32K14x: Flash Architecture

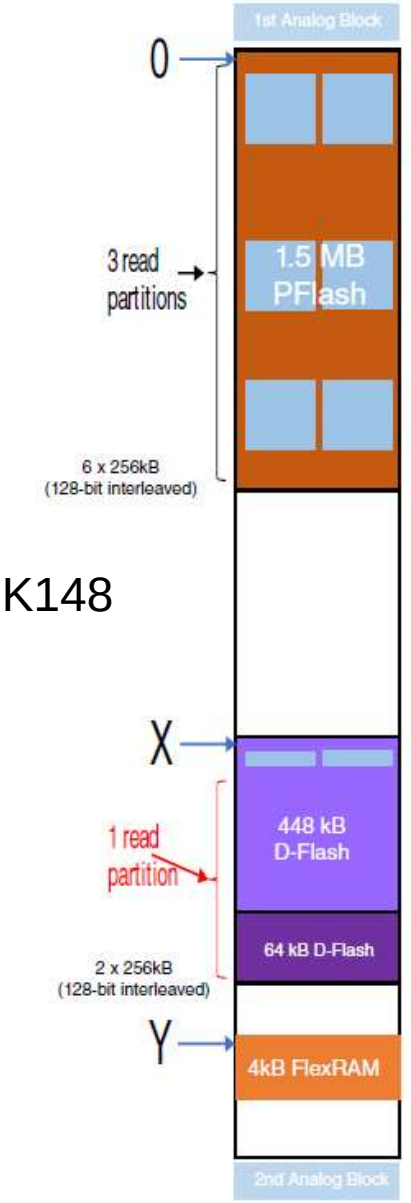
S32K144



S32K146



S32K148



S32K14x: Flash Architecture

FOTA relevant features:

- Sector size (= minimum erase size)
  - **4K Bytes** in Program Flash (bank 0)
  - **2K Bytes** in Data Flash (bank 1)
- Read-While-Write (RWW) features between Bank0 (Program Flash) and Bank1 (Data Flash)

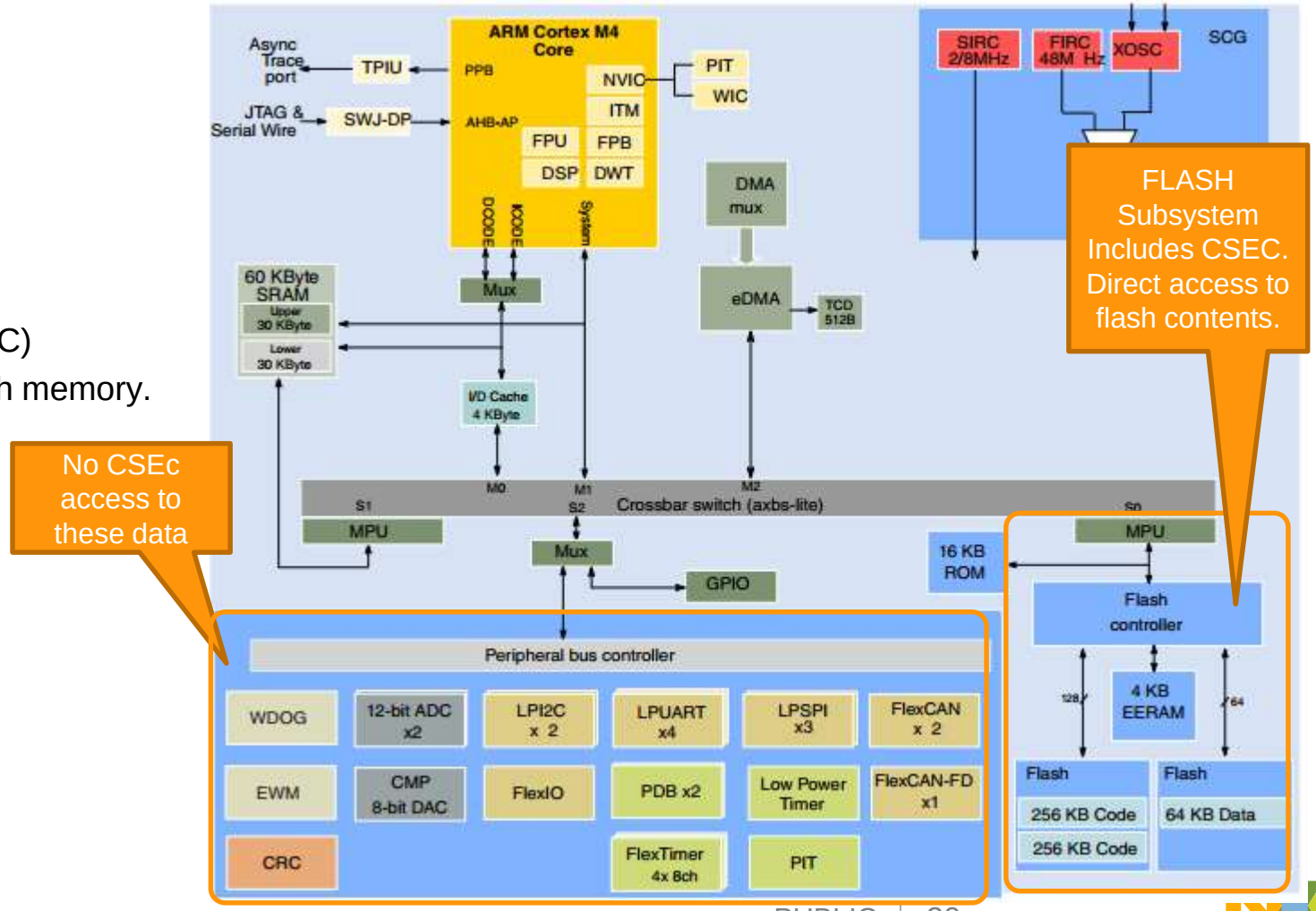
Key additional flash features:

- **C90TFS** (Thin-Film-Storage) technology
- ECC support: **Single Bit Error Correction and Double Bit Error Detection**
  - 32bit ECC word in data flash
  - 64bit ECC word in program flash
- Access time: **Flash clock is about #1/4 of the core clock**

S32K Security Module (CSEc) – Overview

- SHE functionality moves from dedicated master module into the flash system
- **Full SHE Specification compliant** and support of all Global-B security requirements
- **Secure key storage** only accessible by CSEc
- **True Random Number** System
- **Sequential boot / parallel** boot supported
- CSEc supports **AES-128** with ECB, CBC and CMAC mode
- **Crypto Keys**
  - Several General-Purpose keys
  - Special Purpose keys (e.g. Secret, Master and Secure-Boot Key & CMAC)
  - Support of additional encrypted keys in public flash memory.

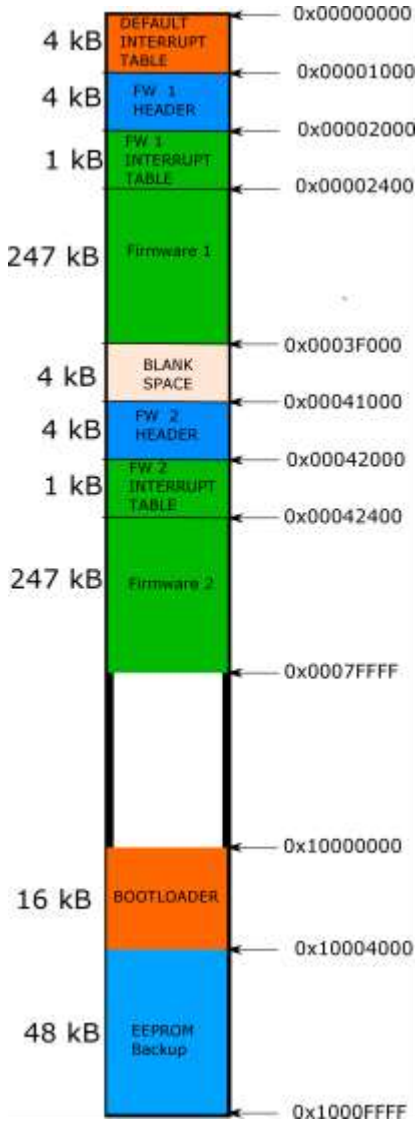
- KEY-Properies
  - Write-protection
  - Secure-Boot-Failure
  - Debug-Connect
  - Wildcard-UID
  - Key-Usage (key or CMAC)
  - Verify-Only
  - 28bit-Update-Counter



04.

S32K144 Use Case

S32K144 use case: **Memory Map for A/B Swap**



- Default Interrupt table and bootloader not erased.
- 0x000000004 -> stores bootloader Reset Handler
- Reset Handler located at Bootloader space
- FW HEADER:
  - Fw version .
  - Developers information.
  - Signature.
  - Erased/Updated after each firmware update
  - Size: 4kB (sector size)
- FW size 248kB (62 sectors)
- RWW between bootloader and firmware application.
- EEPROM: Store secure keys, application usage.

S32K144 use case: **Boot process**

1. After Reset: fetch PC value @ 0x00000004

2. Bootloader init peripherals

3. Bootloader search for oldest and newest image.
   - Check FW Header information
   - Value 0x55AA55AA, at end of fw header
   - Assign FW to be updates (Oldest)

4. Jump to newest application
   1. Relocate VTOR table
   2. PC fetch value from new firmware interrupt table

S32K144 use case: **Update Process**

1. After Reset: fetch PC value @ 0x00000004

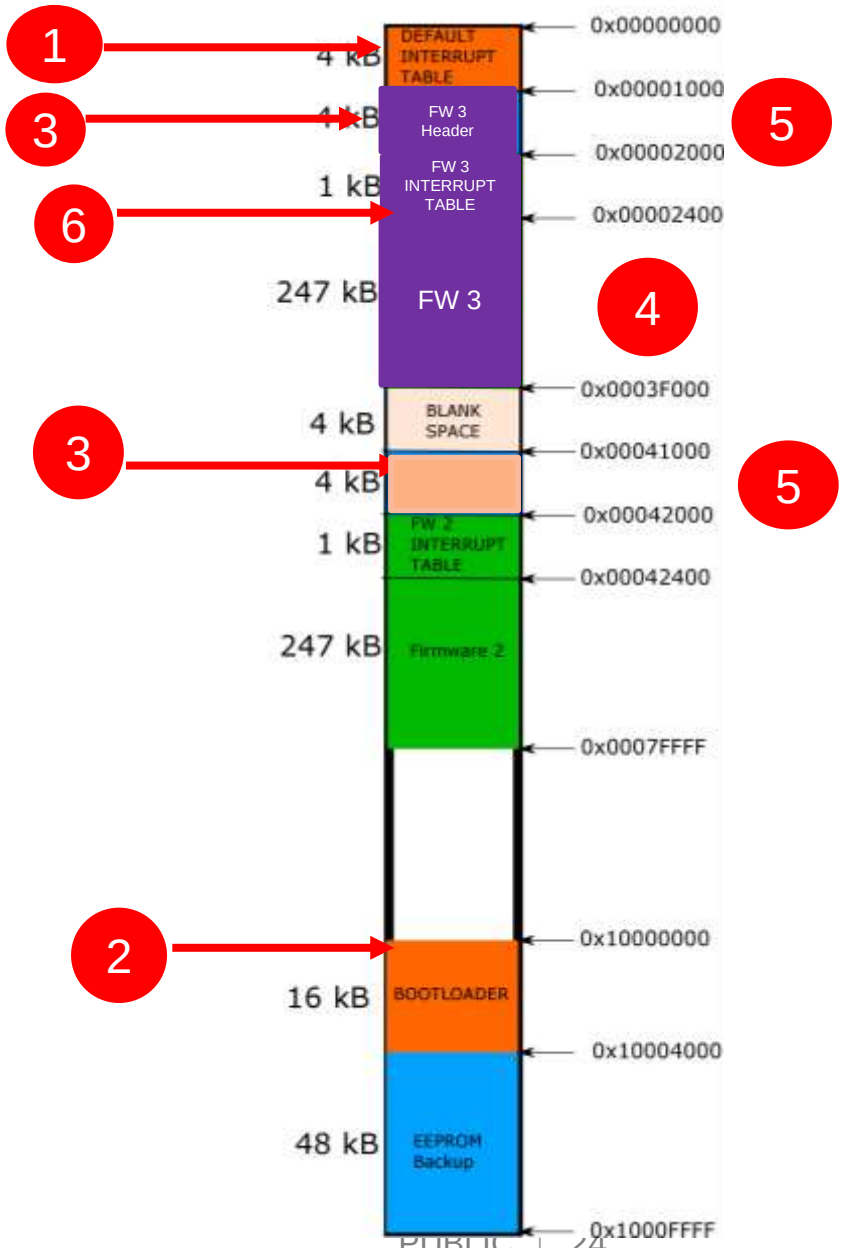
2. 2. Bootloader init peripherals

3. Bootloader search for oldest and newest image.
   - Check FW Header information
     - Value 0x55AA55AA, at end of fw header
     - Assign FW to be updates (Oldest)

4. Update trigger received.
   - Receive header first
     - Validate is a new version
   - Start updating new firmware in oldest location

5. Update Completed
   - Deinit bootloader peripherals
   - Update new firmware header
   - Erase/Update older firmware header

6. Jump to new application
   - Relocate VTOR table
   - PC fetch value from new firmware interrupt table



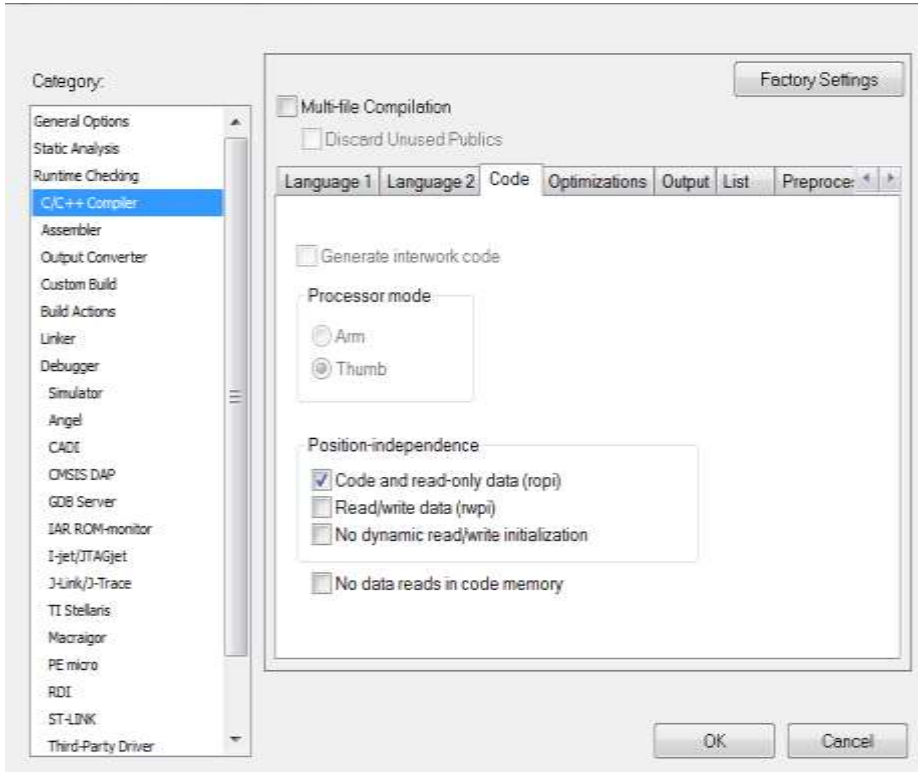
S32K144 use case: **A/B Swap Options without Flash Remapping**

- Problem:
  - 2 images in different physical address.
  - No flash swap, flash remapping feature

- Solutions:
  - Separate object file for each firmware.
    - Requires more overhead in file management!

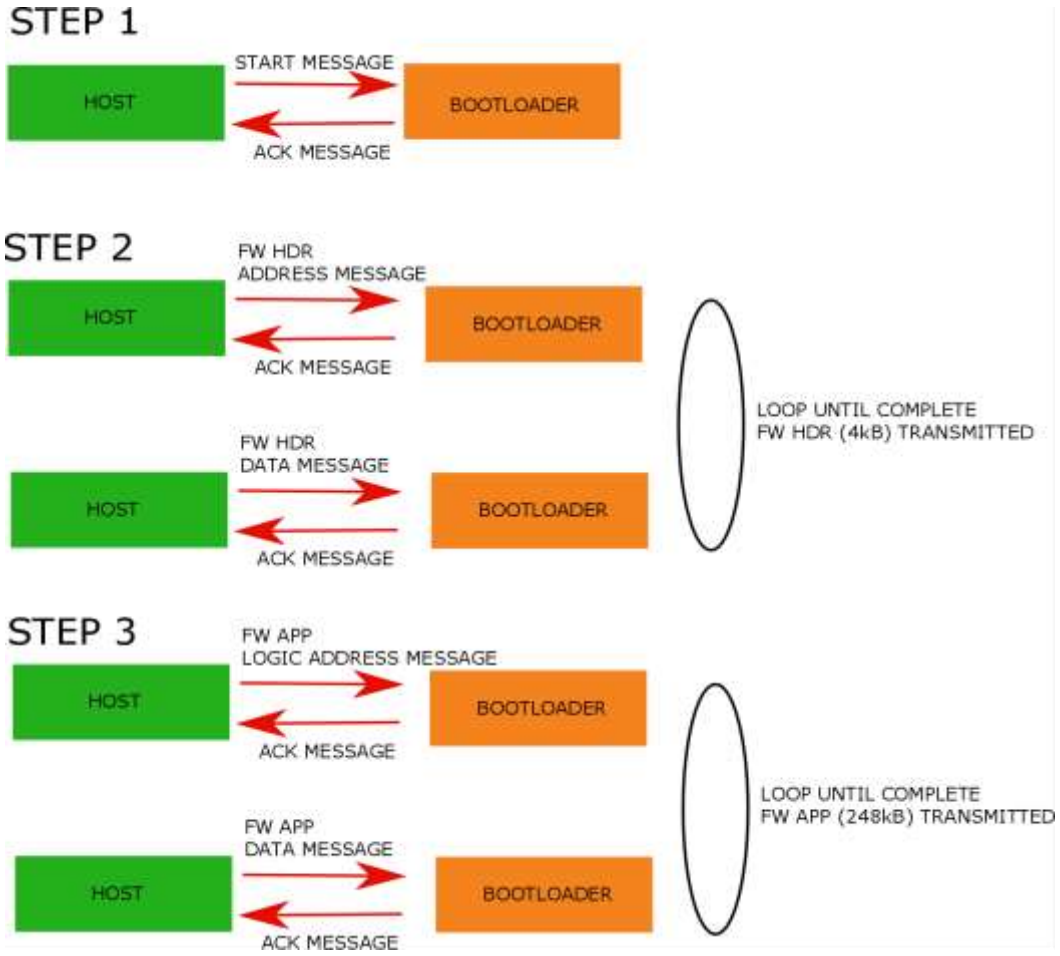
- Position independent code
  - Same linker file for all firmware updates
  - No file management
  - No absolute branches
  - Offset to each interrupt table entry needs to be added. Done automatically by bootloader!
  - Addresses of the interrupt table, should be modified.

IAR ropi feature

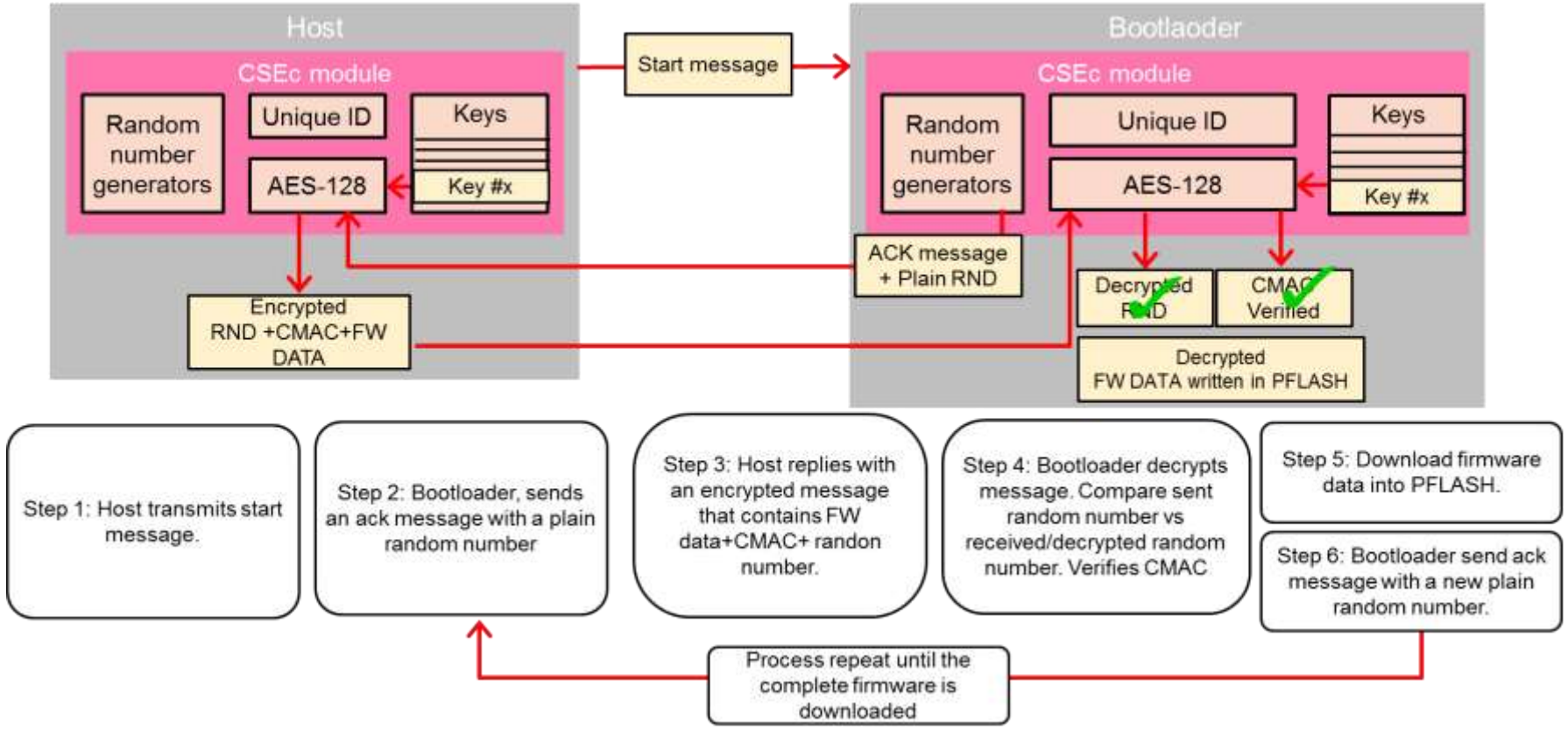


S32K144 use case: **Communication Process**

- Step 1: Trigger update
  - Communication Message from Host to edge node (bootloader fw)
  - Response of ack form host to edge node.
- Step 2: Transmit Header
  - Host sends address
  - Edge node responds with Ack
  - Host sends header data
  - Edge node validate data
  - Edge node responds with Ack
- Step 3: Transmit Application
  - Host sends app logic address
  - Edge node responds with Ack
  - Host sends app data
  - Edge node receives and write data into flash
  - Edge node responds with Ack



S32K144 use case: **Secure Communication Process**



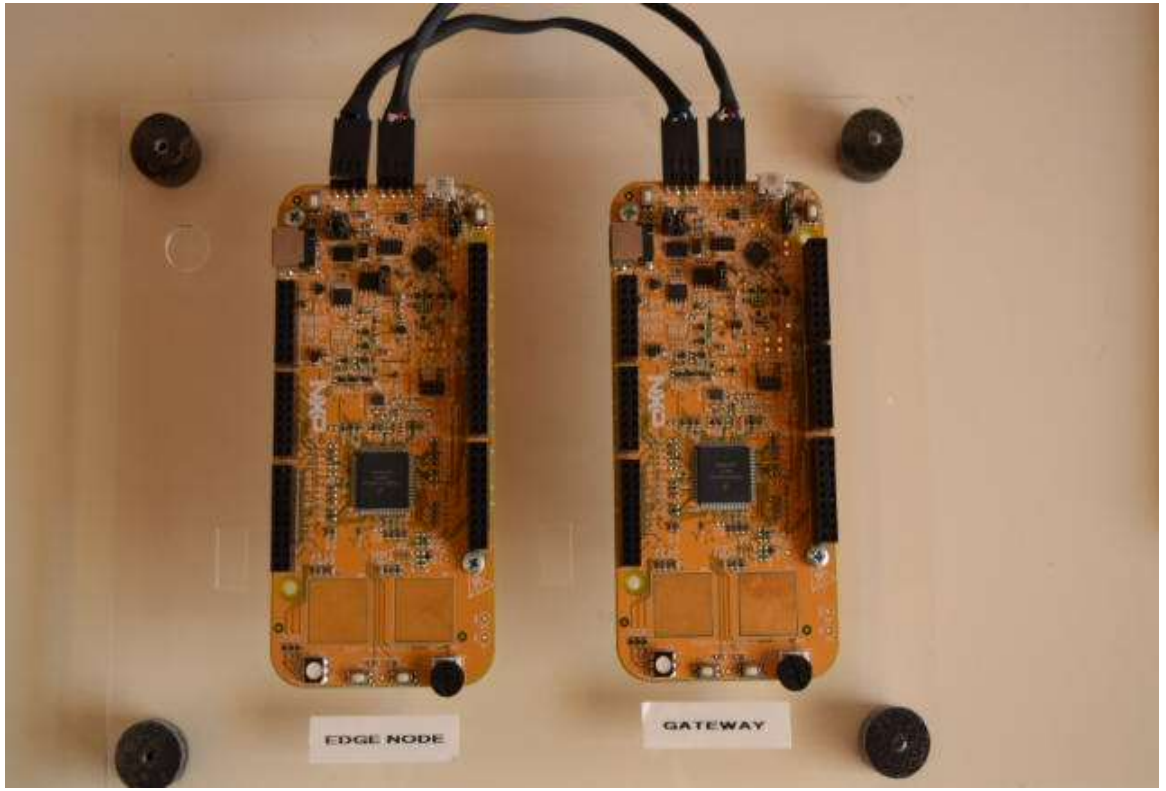
- Random number: protects against replay attacks → Authenticity and freshness of message.
- Encryption: protects against eavesdropping → Confidentiality
- CMAC → Data integrity

05.

Demo

S32K144 Demo: **Setup**

- 2x S32K144 EVBs
  - 1 EVB Gateway
  - 1 EVB Edge Node
- CANFD communication
- Gateway stores 2x application images in its internal memory.
  - FW application: blinking led and serial print.



S32K144 Demo: **Communication messages**

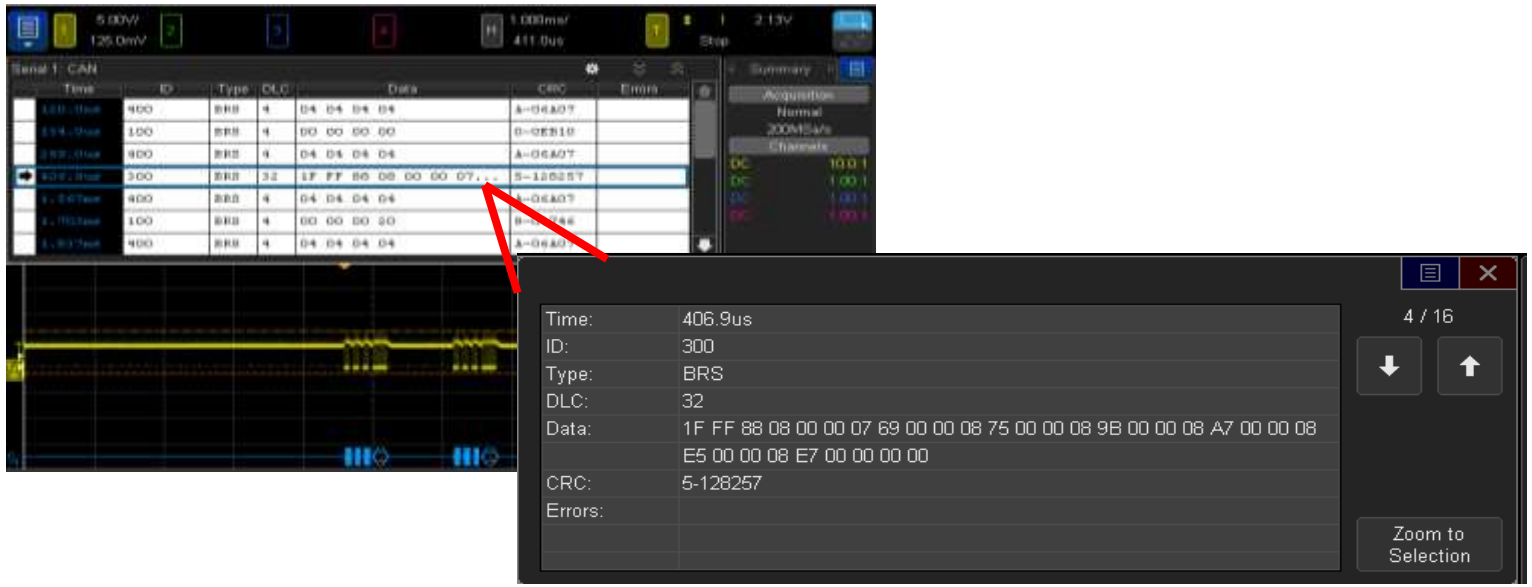
MESSAGE	ID	CAN PAYLOAD BYTES	CANFD PAYLOAD BYTES	PAYLOAD	DIRECTION	DESCRIPTION
START	0x200	4	4	0x15151515	HOST -> BOOTLOADER	Triggers update process
ADDRESS	0x100	4	4	Address of firmware header or firmware logic address End of data: 0x53535353	HOST -> BOOTLOADER	Contains the address for the fw hdr information or the logic address of the fw application. Contains end of data payload.
DATA	0x300	8	32	Firmware header or firmware application data.	HOST -> BOOTLOADER	Contains firmware header or firmware application data., that is downloaded to pflash.
ACK	0x400	4	4	Acknowledge payload: 0x04040404 Error payload: 0x55555555	BOOTLOADER -> HOST	Contains acknowledge payload. Contains error payload.

S32K144 Demo: **CANFD Communication**

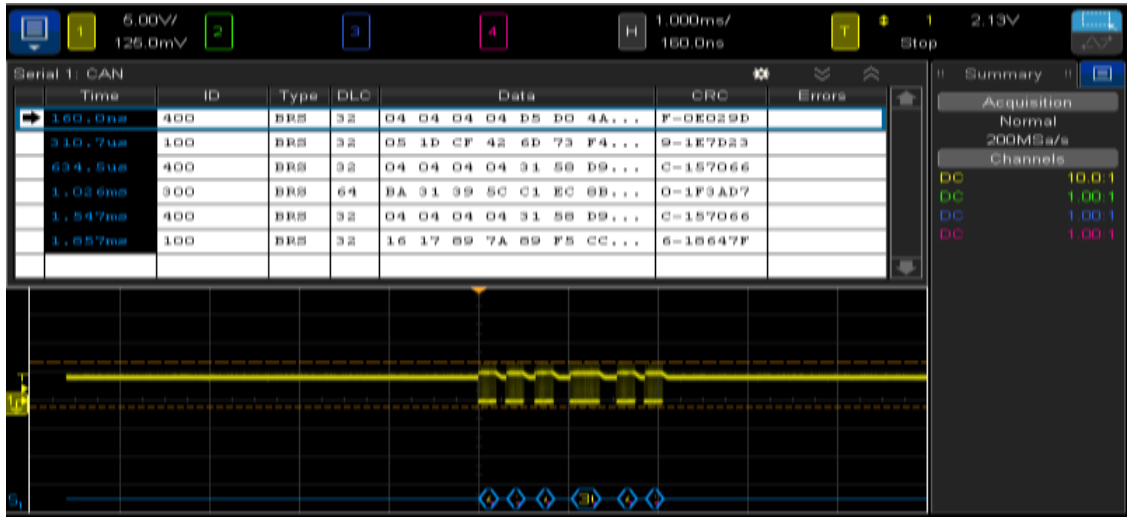
- CANFD @ 2Mbps
- 32B payload
- MSB send first
- FW application plain data is send via CANFD

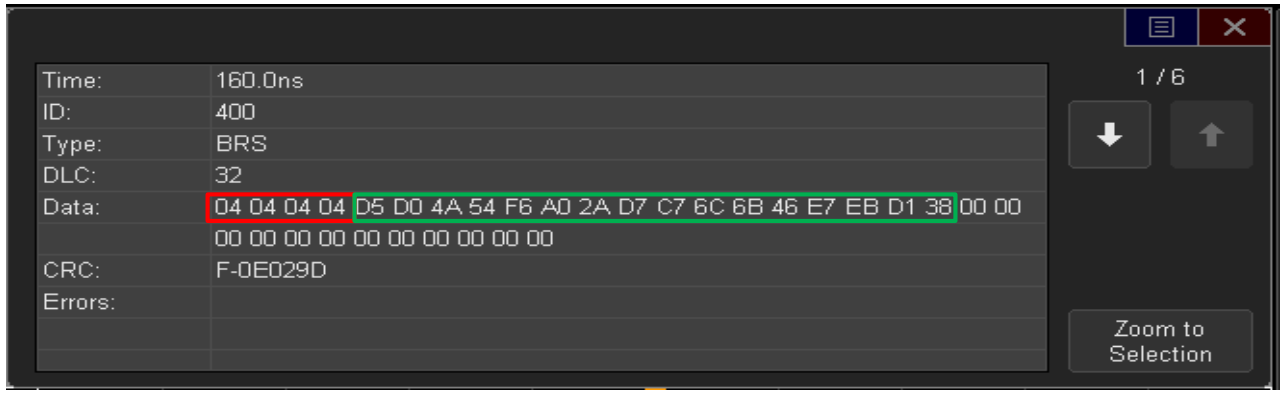
0x2000 : 0x2000 <Hex> New Renderings...

Address	0 - 3	4 - 7	8 - B	C - F
00002000	0888FF1F	69070000	75080000	9B080000
00002010	A7080000	E5080000	E7080000	00000000
00002020	00000000	00000000	00000000	E9080000
00002030	EB080000	00000000	ED080000	EF080000
00002040	E1080000	E1080000	E1080000	E1080000
00002050	E1080000	E1080000	E1080000	E1080000



S32K144 Demo: **Secure CANFD Communication**





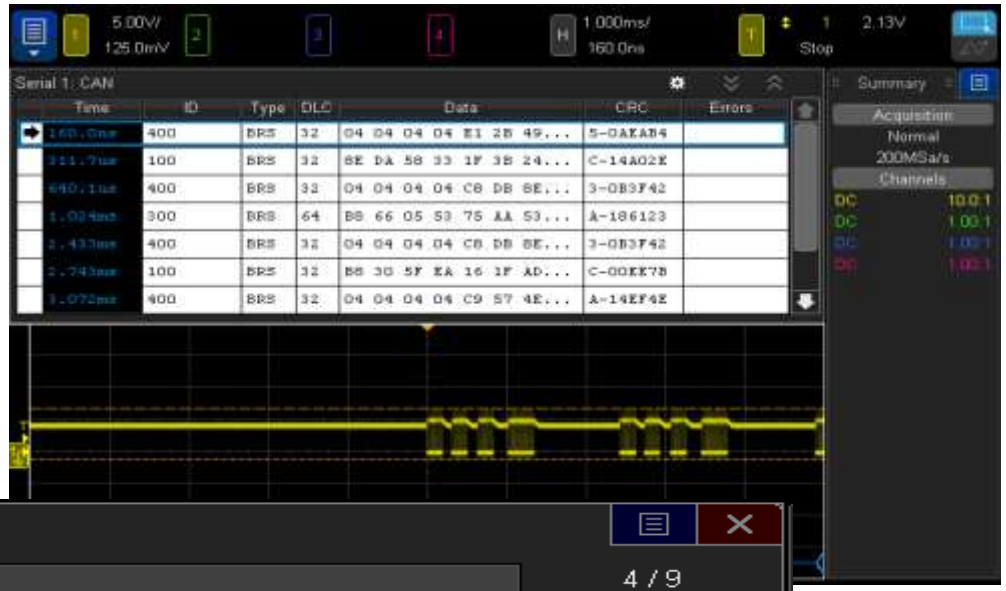
ACK PAYLOAD

PLAIN RANDOM NUMBER

S32K144 Demo: **Secure CANFD Communication**

0x2000 : 0x2000 <Hex> | New Renderings...

Address	0 - 3	4 - 7	8 - B	C - F
00002000	0888FF1F	69070000	75080000	9B080000
00002010	A7080000	E5080000	E7080000	00000000
00002020	00000000	00000000	00000000	E9080000
00002030	EB080000	00000000	ED080000	EF080000
00002040	E1080000	E1080000	E1080000	E1080000
00002050	E1080000	E1080000	E1080000	E1080000



Time: 1.024ms
ID: 300
Type: BRS
DLC: 64
Data: B8 66 05 53 75 AA 53 F3 69 7E 9E 46 6D AA AD 86 01 9E 7A A0 7A 30
EA EC BC 71 3F D8 F5 B5 8F 10 4B 34 57 E6 53 C8 9F EB F9 C6 C6
DE D5 49 A9 22 71 BF 5F 7C 45 FF 2E 08 7E B0 58 EA 0F 26 27 24
CRC: A-186123
Errors:

4 / 9

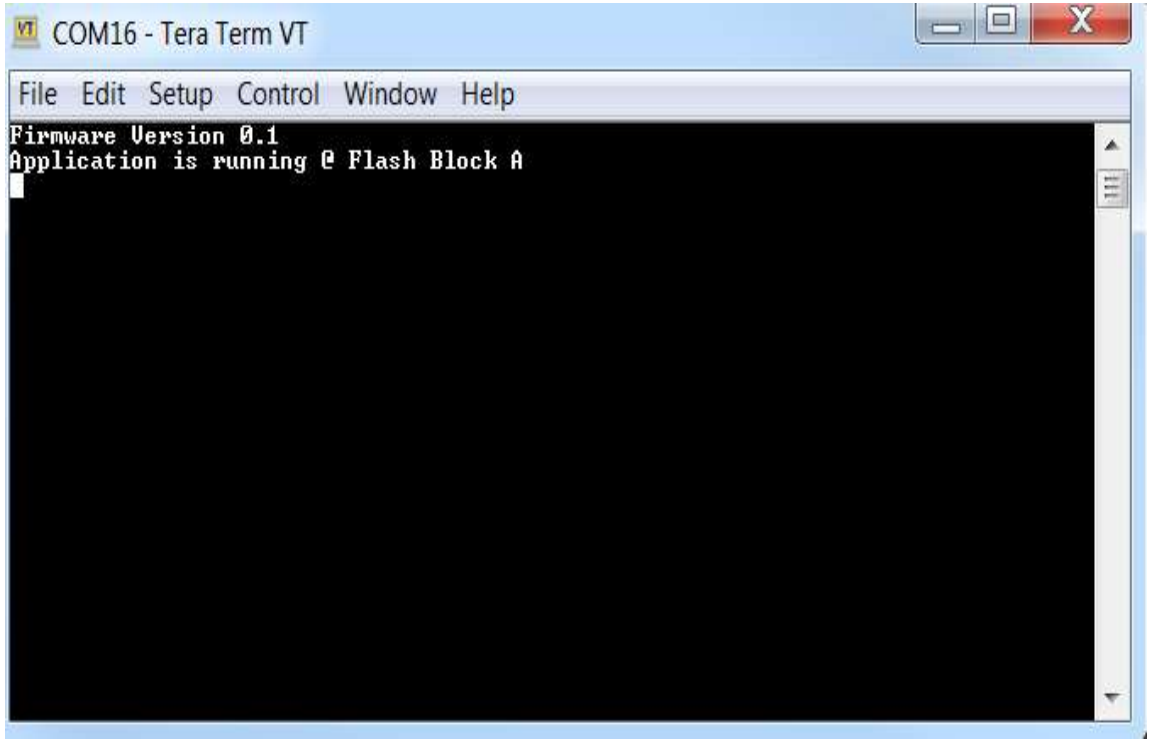
Zoom to Selection

ENCRYPTED
FW APPLICATION DATA

ENCRYPTED
FW APPLICATION CMAC

ENCRYPTED
RANDOM NUMBER

S32K144 Demo: **FW application running on edge node**



S32K144 Demo: **Demo Results**

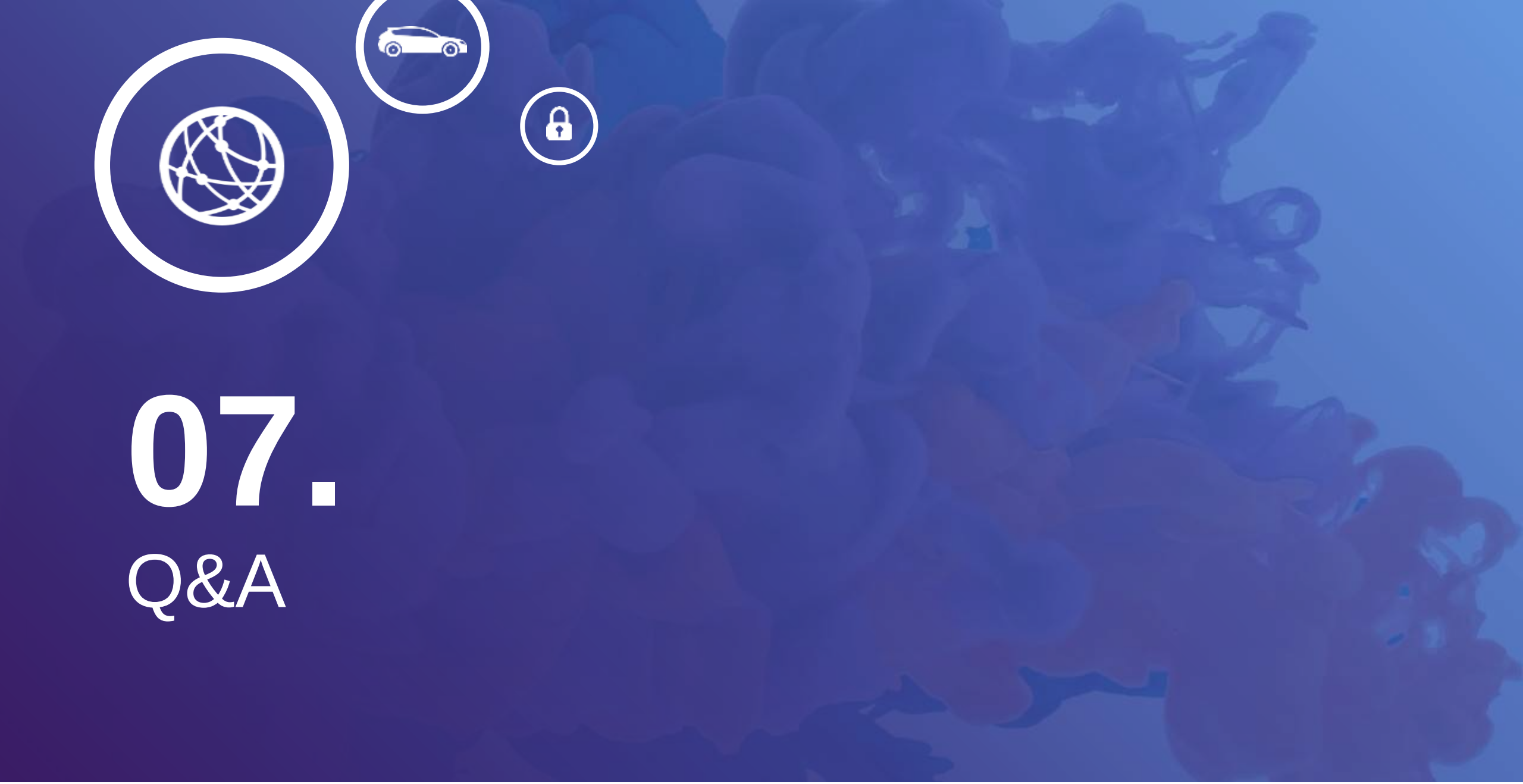
Parameter	Core Clk	Flash Clk	CANFD+ Security CMAC on every message (S)	CANFD (S)
Erase 244 kB Sector by sector 4kB	80 MHz	20MHz	0.2692s	0.26914s
Erase 4kB	80 MHz	20MHz	0.00421s	0.00421s
Program and check 4kb	80 MHz	20MHz	0.1194s	0.1219s
Program and check 32B Average of 1000	80 MHz	20MHz	760.38us	754.16
Decipher 64B Average of 1000	80 MHz	20MHz	43.28us	N/A
Decipher 32B Average of 1000:	80 MHz	20MHz	31.69us	N/A
CMAC verify 32B	80 MHz	20MHz	44.1us	N/A
From Start frame to Jump application	80 MHz	20MHz	17.79s	11.39s
Update per KB	80 MHz	20MHz	0.073s	0.047s

06.

Summary

S32K1xx – an OTA solution for automotive edge nodes.

- **Structured for Secure or traditional OTA updates**
- **Most scalable portfolio** – based on ARM Cortex
- **Future-proof** - Superior Performance and Features



07.

Q&A

SECURE CONNECTIONS
FOR A SMARTER WORLD