FuSa in a nutshell - Memories in AURIX™ TC3xx

32-bit TriCore™ AURIX™ TC3xx microcontroller

About this document

Scope and purpose

The functional safety (FuSa) requirements of the standards in automotive, industrial, and other fields are a challenging subject. This document intends to provide basic knowledge and guidelines related to the NVM subsystem for the users of the AURIX™ TC3xx microcontroller in a functional safety scope.

This application note is part of a series of documents named 'FuSa in a nutshell' and listed in references [1].

Intended audience

This application note is intended for all those evaluating the AURIX™ TC3xx MCU, including functional safety and application engineers on the user side.

Keypoints

- The NVM subsystem of the AURIX™ TC3xx is presented at architectural and functional levels
- The safety mechanisms related to the NVM subsystem are presented
- Software guidelines are proposed on how to react to ECC errors and other assumptions of use

About this product family

Product family

This document focuses on the AURIX™ TC3x family, which are well-suited microcontrollers for safety- and security-critical applications.

Target applications

This document targets the Infineon automotive applications.

Structure of the document

This document will present AURIX™ TC3xx the NVM subsystem and introduce its related safety mechanisms.

Disclaimer

Application use cases, if any, are for training purposes only and are not to be taken as a blueprint for productive development.

This document does not replace the TC3xx safety manual but rather gives a simplified idea about the safety mechanisms of the TC3xx NVM subsystem.

Table of contents

About this document **1**
About this product family **1**
Table of contents **2**
1 Introduction to NVM subsystem **3**
1.1 PFLASH 3
1.1.1 PFlash architecture 3
1.2 PFRWB 4
1.3 PFI 4
1.4 DFLASH 5
1.4.1 Configuration blocks UCB and CFS 5
1.5 DFRWB 6
1.6 DMU 6
1.6.1 BootROM 6
1.7 FSI 6
2 Monitoring of NVM subsystem **7**
2.1 PFLASH functional safety mechanisms 7
2.1.1 Error correction code (ECC) 7
2.1.1.1 Managing PFlash ECC errors alarms 8
2.1.2 Monitoring the ECC mechanism 8
2.1.3 PFlash banks monitoring 8
2.2 PFI-related safety mechanisms 9
2.2.1 Read-path redundancy 9
2.2.2 PFlash incorrect operation 9
2.3 DMU 9
2.3.1 SRI transaction integrity 9
2.3.2 Wait cycles protection 9
2.3.3 Safety endinit mechanism 9
2.3.4 Master-specific access control 9
2.3.5 Read/Write Protection 9
2.3.5.1 Write protection 10
2.3.5.2 Read protection 10
2.4 DFLASH 11
2.4.1 DFLASH ECC 11
2.4.2 EEPROM emulation 11
2.5 Software hints for safety applications 11
2.5.1 CRC integrity checks on safety relevant registers 12
2.5.2 CRC integrity checks on the PFLASH 12
2.5.3 CRC update checks on the PFLASH 12
2.5.4 CPU memory protection system 12
2.5.5 PFlash erase counters 13
3 Conclusion **14**
4 Related resources **15**
References **16**
Revision history **17**
Disclaimer **18**

1 Introduction to NVM subsystem

The non-volatile memory is an important subsystem of the AURIX™ TC3xx platform, with the main purpose to retain code, data, and configurations in a reliable way.

Figure 1 shows the NVM subsystem and its interconnections.

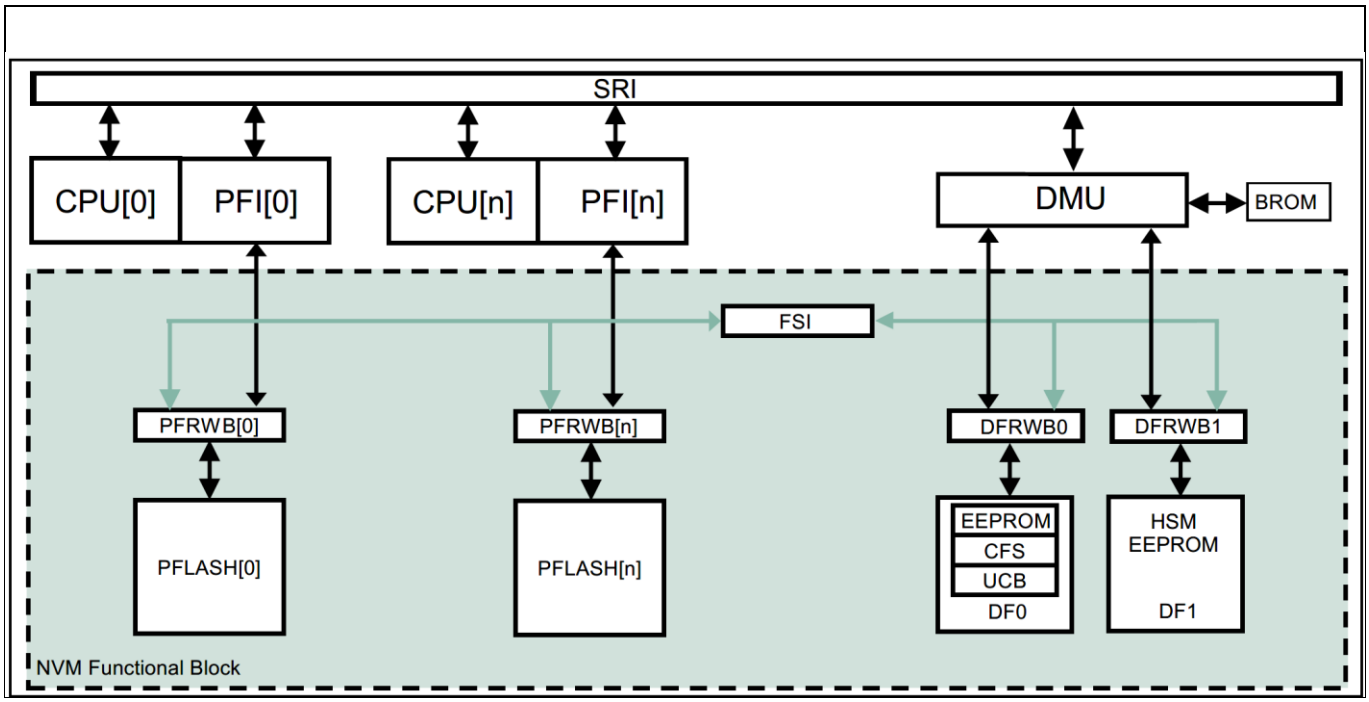


Figure 1 Non-volatile memory subsystem in TC3xx

The following chapter will focus on each main component, its role, and main connections.

1.1 PFLASH

The program flash (PFLASH) is used by the application to store instructions code and data constants.

The execution performance is optimized by using a point-to-point interface between a CPU and its corresponding PFLASH bank through their program flash interface (PFI).

PFlash access is available on both cacheable and non-cacheable segments of the system address range.

The total PFLASH size and distribution change from one device to another within the TC3xx family, refer to the product user manual [2] or the Infineon website for device-specific data.

1.1.1 PFlash architecture

The PFlash is structured into multiple banks ‘PFx (x=0-5)’ and depending on the AURIX TC3xx series and product variants, you can have up to six banks (PF0 to PF5). Each PFx bank contains up to three physical sectors that are composed of 64 logical sectors each. The following are the sizes and characteristics:

- Physical sectors: 1 Mbyte of memory
- Logical sectors: 16 Kbyte of memory, it is the smallest erasable unit
- Page: 32 bytes of memory, the smallest programmable unit

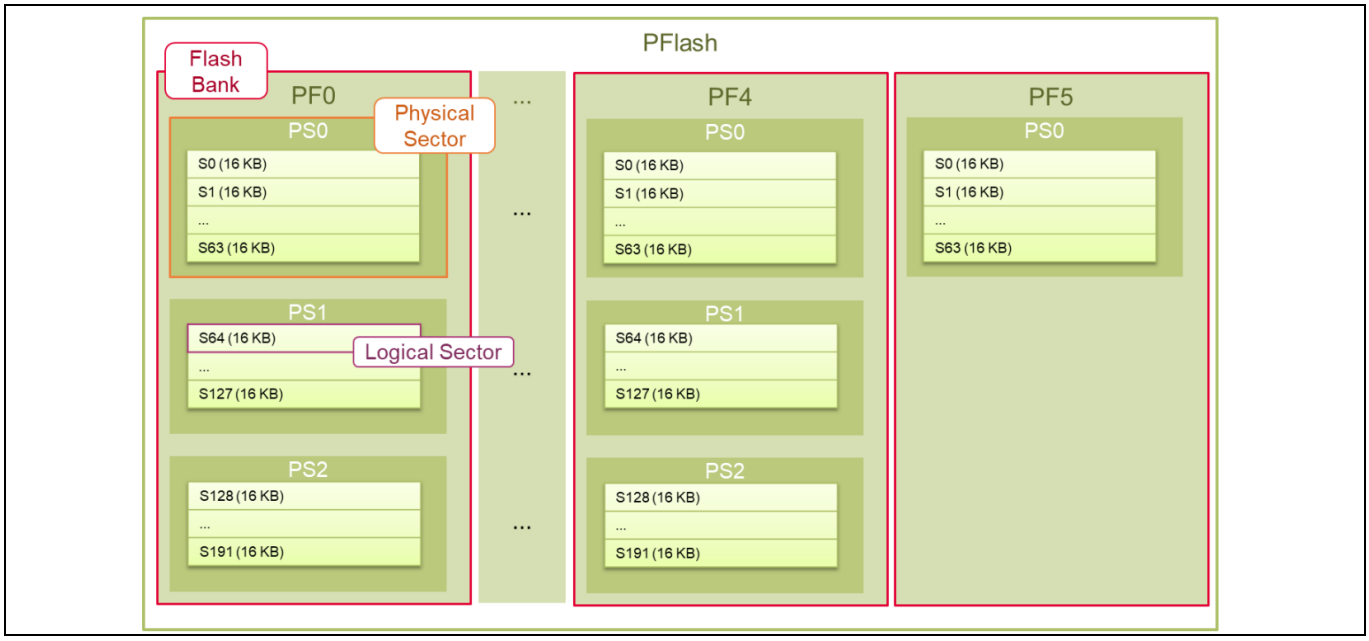


Figure 2 Structure of PFLASH in TC39x

1.2 PFRWB

The PFlash Read/Write Buffer (PFRWB) connects the physical memory to the interfaces (PFI and FSI) and performs the ECC correction and detection.

1.3 PFI

The program flash interface (PFI) provides a unique point-to-point fast connection from the PFx bank to its associated CPU.

Note: When Software over-the-air (SOTA) mode is enabled, the direct CPU to PF bank path is disabled (read operations pass through the shared resource interconnect (SRI) bus).

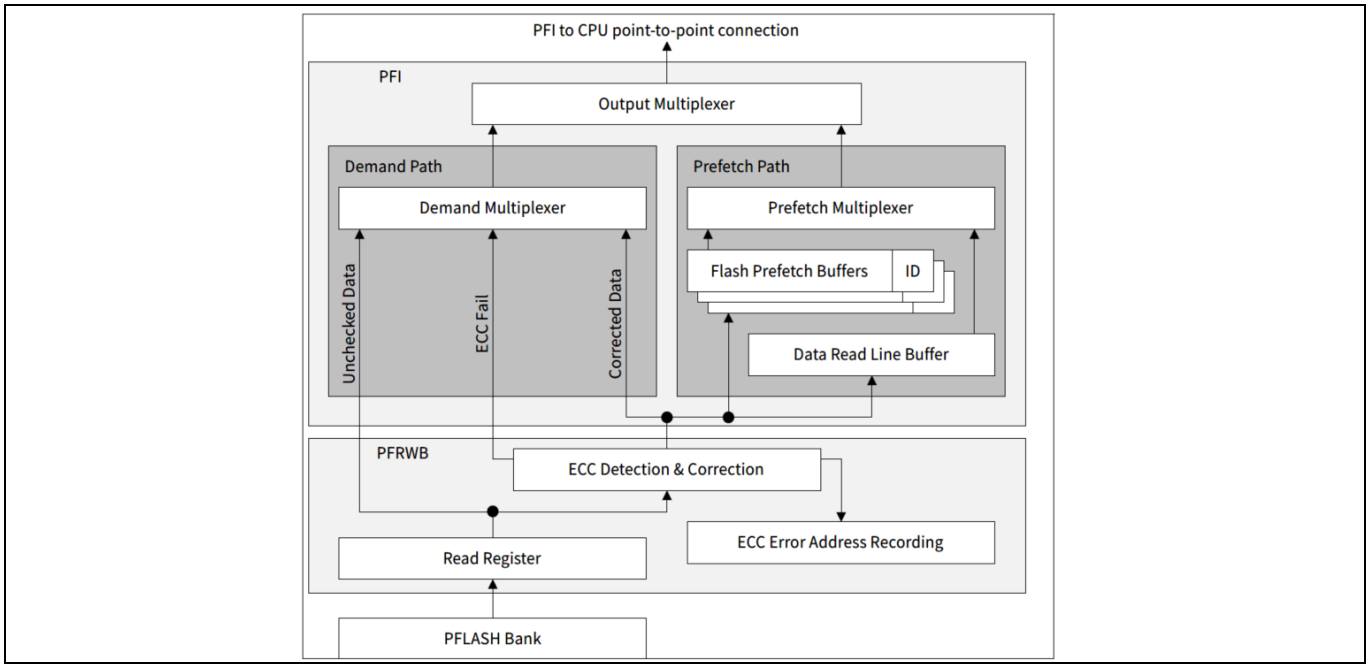


Figure 3 PFI connection to PFLASH bank block diagram

1.4 DFLASH

The Data Flash is used to store frequently updated data, it offers greater endurance per EEPROM sector (allowed number of erase and program cycles) than PFlash, but the read accesses are relatively slower because they are:

- Made across the SRI (no point-to-point connection with CPU)
- Only available on the non-cacheable address range

The DFLASH is divided into two banks DF0 and DF1.

The DFLASH banks can be configured in one of two operating modes, providing a trade-off between data storage capacity and memory endurance.

- Single-ended sensing: This is the default mode for DFLASH EEPROM emulation, in this configuration, each memory cell is read individually. This mode offers a larger storage capacity per sector (4 KB) but has lower endurance (125k) compared to complement sensing mode
- Complement sensing: Designed to achieve higher endurance (500k) by storing the data in differential pairs of cells. This technique reduces the size of sector to 2KB but allows higher endurance versus the single ended mode

Both DFLASH banks must be configured to use the same operating mode.

1.4.1 Configuration blocks UCB and CFS

Dflash bank 0 (DF0) contains regions to store configuration data:

- User configuration blocks (UCBs)
- Configuration sector (CFS), programmed at production and not accessible by the user

1.5 DFRWB

The DFlash Read/Write Buffer (DFRWB) connects the DFlash to the FSI.

1.6 DMU

The data memory unit (DMU) is a central component of the NVM subsystem and not exclusive to PFlash or Dflash, and serves to:

- Controls the command sequences that are executed on program and data flash memories
- Handle the reads from the Dflash

1.6.1 BootROM

The DMU contains the BootROM where the system firmware is programmed (non-user code).

The system firmware is executed immediately after PORST release.

The firmware is partially configurable by the user via the user configuration blocks (UCB) registers.

1.7 FSI

The flash standard interface (FSI) executes erase, program, and verify operations on both program and data flash.

2 Monitoring of NVM subsystem

The AURIX™ TC3xx is developed for functional safety use cases. The NVM block contains mechanisms that prevent a fault from leading directly to a safety violation or to remains latent.

The following chapters will focus on the PFlash part and its related components and then present additional mechanisms related to the Dflash.

2.1 PFLASH functional safety mechanisms

The content of PFlash banks is crucial because it contains the code executed by the CPU. Any fault leading to a transient or permanent corruption of the instructions code can lead to severe malfunctions. Therefore, PFlash offers dedicated safety mechanisms for monitoring several failures' modes.

2.1.1 Error correction code (ECC)

The reliability of the data stored in PFlash is increased by the ECC mechanism that functions by adding redundant ECC bits (calculated from the data and the address) during each programming operation.

With each read operation, the redundancy allows the hardware to detect and correct, if possible, a limited number of errors.

The PFLASH of the TC3xx family have a **Double error correction Triple error detection (DECTED)** ECC logic that is calculated over 256 bit and is capable of:

- Detect and correct single-bit errors (SBE): This triggers an alarm and log the error in the SBABx (x=0-16) record buffer.

Note: When the SBAB becomes full, the SMU is informed and triggers SBAB full alarm. In this case, the DBAB limit is halved from 2 to 1.

- Detect and correct double-bit errors (DBE): This triggers an alarm and log the error in the DBAB x (x=0-1) record buffer

Note: A double bit error can be an indication of a word line (aligned 1 KB) failure and a higher risk of having an uncorrected multiple bits error. The software can run a test by reading pages around the DBE address and checking for other ECC errors.

Note: When the DBAB becomes full, the SMU is informed and triggers DBAB full alarm.

- Detect up to 3-bit error (TBE) which is considered as uncorrected multiple bit error and tracked in the MBAB record buffer.

*Note: There is only one MBAB record buffer **MBABRECORD0** and when it becomes full (with the first multiple bit error detected) the SMU triggers an alarm.*

- Detect all zero errors (reading an erased sector) which are tracked simultaneously in ZBAB **x (x=0-3)** and MBAB record buffer.

Note: when reading an erased sector, the user can use the erase verify command to check if the sector was correctly erased without triggering the ECC error.

- Detect address and all one error, these are reported and recorded as uncorrected multiple bit error

Note: The default configuration of register FLASHCON2 is set to enable the ECC and track the errors in the record buffer and unless changed for test purposes, the ECC must remain enabled, and the errors tracked.

2.1.1.1 Managing PFlash ECC errors alarms

When an alarm is triggered due to an ECC error or because one of the record buffers is full, the application is expected to take the right reaction depending on the criticality of the error and the safety level of the application.

Important: The choice of alarms reactions remains under the responsibility of the user; the following are suggested guidelines for a typical safety application.

Corrected errors (single- and double-bit errors) have no impact on the data integrity and so their individual alarms (SBE and DBE alarms) can be ignored, then a reaction can be implemented on SBAB and DBAB record buffer full alarms since having multiple single- and double-bit errors is a risk indication.

Uncorrected errors alarms (MBAB full due to multi-bit error, address errors...) must be reacted to with appropriate measures and consider the NVM as non-functional.

Note: These alarms are level type alarms and to clear them, the record buffers need to be cleared.

2.1.2 Monitoring the ECC mechanism

The PFLASH ECC detection and correction logic is continuously monitored and checked so that the ECC operations are fault free. If an error is detected, then an SMU alarm is generated.

2.1.3 PFlash banks monitoring

Each PFlash bank is monitored for unintended selection and operations like program and erase, if this unintentional pflash bank selection is detected then an alarm is triggered.

2.2 PFI-related safety mechanisms

As part of safety critical operations (CPU reading and fetching from PFlash) the PFI is equipped with dedicated safety mechanisms.

2.2.1 Read-path redundancy

The PFI have lockstep mechanism that detects failures of the CPU-PFLASH read-path via hardware redundancy.

The lockstep error is reported to the local CPU, then it flags the PFLASH read-path monitor alarm.

This mechanism has a continuously running background self-test.

2.2.2 PFlash incorrect operation

The PFI checks if the FSI is performing an operation on Pflash which is not expected by the DMU. If an incorrect operation is detected, then an error is generated to the CPU, resulting in PFLASH read-path monitor alarm.

2.3 DMU

Operating the PFLASH bank follows a specific sequencing, any unintended operation is detected and rejected with a triggered alarm.

2.3.1 SRI transaction integrity

The SRI interface can detect integrity errors of an SRI transaction and in particular the DMU transactions through the SRI are covered.

2.3.2 Wait cycles protection

The wait cycles configured in DMU_HF_PWAIT register adds a latency to PFlash reads, this latency is needed due to the CPU operations being much faster than the PFlash operations.

A corrupted wait cycles value can lead to a corrupted read operation and so the PWAIT value is continuously monitored and protected. In case an unintentional change is detected an alarm is triggered.

2.3.3 Safety endinit mechanism

The safety endinit is a mechanism that restrict the modification of safety critical registers, like the DMU configuration registers. A PROER error flag is generated if there is a modifying access to the PFLASH without removal of Safety Endinit protection. For more information, see [2].

2.3.4 Master-specific access control

The DMU enforces master-specific access control for its own configuration registers and for PFLASH read operations, restricting access to only authorized bus masters.

2.3.5 Read/Write Protection

user can enable different types of PFlash protections using the UCB_Flash configuration block, and then to allow for legitimate software updates, the protections can be temporarily disabled using a password-based mechanism.

2.3.5.1 Write protection

PFLASH write protection prevents unauthorized programming or erasing of PFlash sectors.

The UCB_PFLASH contains configurations for sector-level write protection, which are transferred into the DMU_HP_PROCONP registers.

2.3.5.2 Read protection

PFLASH read protection restricts read access to flash memory. This is particularly effective when the device is not booting from the internal flash.

Read protection is configured by setting the RPRO bit to 1B in the DMU_HF_PROCONPF register, using the value stored in UCB_PFLASH_ORIG / COPY, this will result in the setting of DDFP and DDFD bits in HF_CONTROL register.

HF_CONTROL
Flash Control Register **(0000014$_H$)** **Application Reset Value: 0000 0320$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													RES	CERASE	CPROG
r													r	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						DDFD	DDFP	WSERRINJ	RES	FSIENPE		DUCG	RES		
r						rwh	rwh	rw	r	rw		rw	r		

DDFP	8	rwh	**Disable Read from PFLASH** This bit enables/disables the read access to PFLASH. **DMU ramp up and SSW start up** This bit is automatically set with reset and is cleared during startup, if no Read Protection installed, and during startup SSW in case of internal start out of Flash. **User Software: Clearing PFLASH Read Protection** Once set, **HF_CONTROL**.DDFP can only be cleared when **HF_PROTECT**.PRODISP or **HF_PROTECT**.PRODISP0-5 or not **HF_PROCONPF**.RPRO **User Software: Setting PFLASH Read Protection** Software must write **HF_CONTROL**.DDFP to 1_B 0_B Read access from the PFLASH memory area is allowed. 1_B Read access from the PFLASH memory area is not allowed.

DDFD	9	rwh	**Disable Data Fetch from DF0_EEPROM** This bit enables/disables the data fetch from DF0_EEPROM. **DMU ramp up and SSW start up** This bit is automatically set with reset and is cleared during startup, if no Read Protection installed, and during startup (BROM SW) in case of internal start out of Flash. **User Software: Clearing DF0_EEPROM Read Protection** Once set, **HF_CONTROL**.DDFD can only be cleared when **HF_PROTECT**.PRODISD or not **HF_PROCONDF**.RPRO **User Software: Setting DF0_EEPROM Read Protection** Software must write **HF_CONTROL**.DDFD to 1_B 0_B Read access from the DF0_EEPROM memory area is allowed. 1_B Read access from the DF0_EEPROM memory area is not allowed.

For more information, see [2].

Attention: ***A global write protection is applied over the entire PFLASH when global read protection is enabled.***

2.4 DFLASH

The Data Flash part of the NVM subsystem is not considered as safety critical, and the following mechanisms and guidelines are proposed for robust system purposes.

2.4.1 DFLASH ECC

The reliability of the data stored in Dflash is increased by the ECC mechanism type TECQED: correction of single-bit, double-bit, and triple-bit errors and detection of quad-bit errors calculated over 64 data bits.

Note: *Unlike PFlash the Address is not included in the ECC calculation, therefore the addressing faults are not detected.*

Reading an erased data Flash sector does not trigger an ECC error (in single-ended sensing mode).

2.4.2 EEPROM emulation

EEPROM emulation is a software algorithm used to spread the update frequency seen by the memory cells over a large range of Data Flash, effectively increasing the endurance of DFLASH sectors.

This algorithm is required to be resistant against operation aborts (Power off, resets…) and be able to recover and resume at the next startup.

Infineon provides within its AUTOSAR MCAL drivers a robust EEPROM emulation, for further information, visit Infineon AUTOSAR web page or contact Infineon sales.

2.5 Software hints for safety applications

The following guidelines describe assumptions of use that help detect and prevent a faulty flash operation or faulty code execution.

2.5.1 CRC integrity checks on safety relevant registers

Before entering the run mode, application SW shall monitor the data stored in the following safety relevant registers by calculating a cyclic redundancy check (CRC) and comparing it to an expected checksum.

- DMU_HF_PROCONPF.RPRO
- DMU_HF_PROCONUSR
- DMU_HF_PROCONDF
- DMU_HF_PROCONRAM
- DMU_HF_PROCONDBG (if used)
- DMU_SP_PROCONHSM (if used)
- DMU_SF_PROCONUSR
- DMU_SP_PROCONHSMCBS (if used)
- DMU_SP_PROCONHSMCX0/1 (if used)
- DMU_SP_PROCONHSMCOTP0/1 (if used)
- DMU_SP_PROCONHSMCFG (if used)
- DMU_HP_PROCONOTPpx (p,x=0..5)
- DMU_HP_PROCONWOPpx (p,x=0..5)
- DMU_HF_PROCONTP

2.5.2 CRC integrity checks on the PFLASH

During the fault diagnostic time intervals, a cyclic redundancy check (CRC) shall be calculated on safety-related code ranges of the PFlash (or on the entire PFLASH). Then, the result is compared to an expected checksum.

The purpose of this software diagnostic is to detect latent faults that can accumulate into multi bits uncorrectable errors on PFlash and stay undetected by the ECC since they may not be accessed by a read operation.

In case of unmatched CRC with the expected value or if an ECC error is triggered during the CRC calculation, an alarm should be triggered, and the user should react depending on the system safety architecture.

To reduce the load on the CPU, the CRC can be calculated using the DMA once during the driving cycle.

2.5.3 CRC update checks on the PFLASH

After an update on the PFlash, it is necessary to read the entire PFlash and calculate a CRC checksum that is compared to an expected value.

This operation must be executed before starting the safety application.

2.5.4 CPU memory protection system

This protection method incorporates hardware mechanisms related to the TriCore™ that protect user-specified memory ranges from unauthorized read, write, or instruction fetch accesses.

For further information about the CPU MPU, refer to [3].

2.5.5 PFlash erase counters

Erase operations are logged into the erase counters and as a method to detect intrusions and unintended erases, the address of the first erased sector and the number of erased ones, can be checked and compared to a checksum to detect any unintended operation.

The erase counters are not a functional safety mechanism, but they can be used by the software to track unintended erase operations.

Note: *The number of erases that can be logged is limited to up to 256 low-priority entries and 256 high priority entries, critical PFlash sectors can be assigned to higher priority with UCB_ECPRIO. Refer to the user manual for further details.*

3 Conclusion

This document explained the NVM subsystem in the AURIX TC3xx microcontroller family and goes over, in a simplified way, the available hardware safety mechanisms and guidelines that ensure the programmed data integrity and correct functioning of the different presented components.

4 Related resources

- Product family: AURIX™ TC3xx
- Product datasheet: AURIX™ TC3xx Datasheets
- Developer Community: AURIX™ Community

References

[1] Infineon Technologies AG: AN1000 - FuSa in a Nutshell - release note; Available online

[2] Infineon Technologies AG: AURIX™ TC3xx User manual; Available online

[3] Infineon Technologies AG: TriCore™ TC1.6.2 core architecture manual; Available online

Revision history

Document revision	Date	Description of changes
1.0	2025-11-27	Initial release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2025-11-27

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2025 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

AN1008

Important Notice

Products which may also include samples and may be comprised of hardware or software or both ("Product(s)") are sold or provided and delivered by Infineon Technologies AG and its affiliates ("Infineon") subject to the terms and conditions of the frame supply contract or other written agreement(s) executed by a customer and Infineon or, in the absence of the foregoing, the applicable Sales Conditions of Infineon. General terms and conditions of a customer or deviations from applicable Sales Conditions of Infineon shall only be binding for Infineon if and to the extent Infineon has given its express written consent.

For the avoidance of doubt, Infineon disclaims all warranties of non-infringement of third-party rights and implied warranties such as warranties of fitness for a specific use/purpose or merchantability.

Infineon shall not be responsible for any information with respect to samples, the application or customer's specific use of any Product or for any examples or typical values given in this document.

The data contained in this document is exclusively intended for technically qualified and skilled customer representatives. It is the responsibility of the customer to evaluate the suitability of the Product for the intended application and the customer's specific use and to verify all relevant technical data contained in this document in the intended application and the customer's specific use. The customer is responsible for properly designing, programming, and testing the functionality and safety of the intended application, as well as complying with any legal requirements related to its use.

Unless otherwise explicitly approved by Infineon, Products may not be used in any application where a failure of the Products or any consequences of the use thereof can reasonably be expected to result in personal injury. However, the foregoing shall not prevent the customer from using any Product in such fields of use that Infineon has explicitly designed and sold it for, provided that the overall responsibility for the application lies with the customer.

Infineon expressly reserves the right to use its content for commercial text and data mining (TDM) according to applicable laws, e.g. Section 44b of the German Copyright Act (UrhG).

If the Product includes security features:

Because no computing device can be absolutely secure, and despite security measures implemented in the Product, Infineon does not guarantee that the Product will be free from intrusion, data theft or loss, or other breaches ("Security Breaches"), and Infineon shall have no liability arising out of any Security Breaches.

If this document includes or references software:

The software is owned by Infineon under the intellectual property laws and treaties of the United States, Germany, and other countries worldwide. All rights reserved. Therefore, you may use the software only as provided in the software license agreement accompanying the software.

If no software license agreement applies, Infineon hereby grants you a personal, non-exclusive, non-transferable license (without the right to sublicense) under its intellectual property rights in the software (a) for software provided in source code form, to modify and reproduce the software solely for use with Infineon hardware products, only internally within your organization, and (b) to distribute the software in binary code form externally to end users, solely for use on Infineon hardware products. Any other use, reproduction, modification, translation, or compilation of the software is prohibited. For further information on the Product, technology, delivery terms and conditions, and prices, please contact your nearest Infineon office or visit https://www.infineon.com