Failure analysis process in TRAVEO™ T2G devices

About this document

Scope and purpose

This application note describes the failure analysis flow in TRAVEO™ T2G automotive microcontroller devices. It provides a comprehensive description of how the customer should handle a failing sample and how the device must be prepared to allow the regular failure analysis procedure at Infineon. It covers also the 'No Trouble Found' (NTF) process after all standard tests have passed.

Intended audience

This document is intended for anyone who uses the TRAVEO™ T2G microcontrollers to configure a secured system and to prepare and apply to a failure analysis process for any kind of claim.

Associated part family

TRAVEO™ T2G family

Table of contents

Table of contents

1 Introduction

This application note describes how to properly handle devices which failed in the field at the end customer, during production at the OEM as a ‘0km failure’, or during production at TIER1.

Infineon strives for getting a 0-ppm failure rate of its automotive products. To reach the highest standard of automotive quality, each failing device must be analyzed systematically with various analysis techniques to solve manufacturing defects, material imperfections, or design errors.

The ownership of all activities during the whole failure analysis (FA) procedure is associated with Infineon’s quality division (QA) who is responsible for interacting with customers and internal departments such as Product Engineering, Failure Analysis Engineering, and Product Design. Each case is managed in the “My Cases” in the SFDC system where all interactions are collected and distributed to all collaborators attending to the case. During the whole process, Infineon provides regular intermediate reports which include the latest results of failure investigation, which in turn defines new steps along with respective schedules. A case will be closed with a final 8D-report after the root cause was identified and correction actions are defined. If all standard tests have passed, the claimed sample will be returned to the customer with the result as “No Trouble Found” (NTF).

TRAVEO™ T2G devices include an efficient and reliable security mechanism which supports state-of-the-art safety and security standards for the automotive industry.

The protection state of a TRAVEO™ T2G device is determined by irreversible burning of eFuses. Apart from different configuration parameters and hash values, nonvolatile lifecycle stages are also defined by eFuses. These describe a protection status in which the device is running. A transition from one lifecycle stage to the next is irreversible. The focus in this application note is on the transition to “RMA” lifecycle stage and what the customer must consider to allow the standard FA flow and how the sample can be reused during the NTF process.

This application note refers to AN228680 – Secure system configuration in TRAVEO™ T2G family, which provides detailed information on handling of security and debugging. Review this document carefully to develop a secured system in your application which allows the device to be fully testable during the FA process.

2 Overview of the standard FA flow

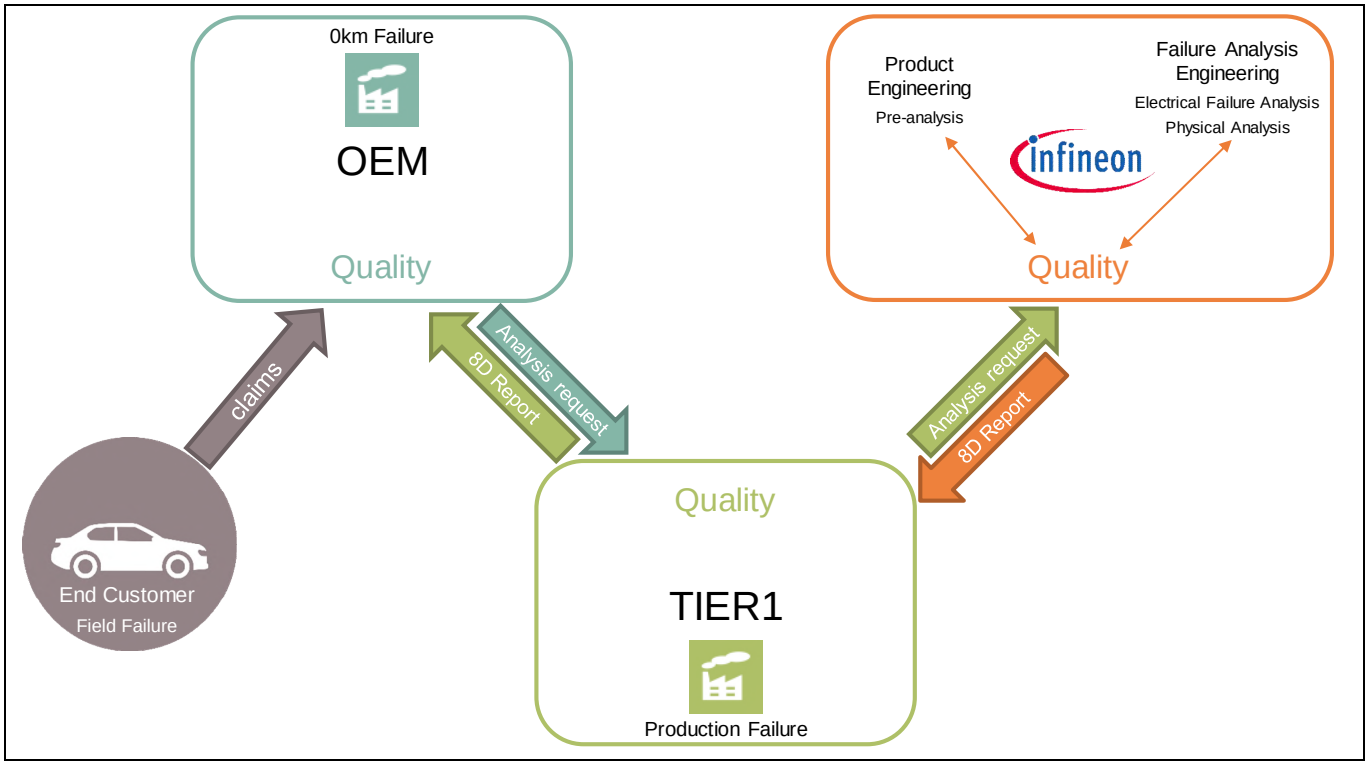

Figure 1 Standard FA flow for automotive microcontrollers

During the lifetime of a TRAVEO™ T2G product, the device can fail in various stages, in the production facilities, or in the final product. Figure 1 shows the standard FA flow for how the customer can claim a defective device. Basically, the TIER1 customer is the direct client of Infineon, and there is no different process depending on where the device failed. For each analysis request, the device will follow a strict analysis procedure which ends with a final 8D-report which summarizes the analysis results; if a defect was found it will explain the root cause in detail. Table 1 summarizes the different failure stages along with the responsibilities to perform a proper FA flow.

Table 1 Life stages where a TRAVEO™ T2G product can fail and how the FA flow is handled

Case	Failure stage	Description	Actions	Responsibility
1	TIER1 production plant (production failure)	The TRAVEO™ T2G device failed during TIER1 EOL test	Perform a swap test to identify if the failure moves with the microcontroller.	TIER1
			Perform initial failure analysis	TIER1
			Create a failure report	TIER1
			Define a clear and comprehensive failure description	TIER1
			QA of TIER1 contacts regional quality partner at Infineon to hand over all required information (see section 2.1)	TIER1 / Infineon
			Infineon creates a new FA case in salesforce.com	Infineon
			TIER1 sends the sample to Infineon	TIER1
			Steps followed are described in Failure analysis procedure at Infineon	Infineon
2	OEM production plant (0km failure)	TIER1 application including TRAVEO™ T2G device failed during OEM EOL test	OEM identifies an issue on the TIER application unit	OEM
			Same procedure as in Case 1	–
3	Final product (field failure)	TIER1 application including TRAVEO™ T2G device failed in the final product in the field	End-consumer claims the product (e.g., vehicle) at the dealership	End-customer/ dealership
			Same procedure as in Case 2	–

2.1 Minimum information to be provided to Infineon

To ensure a fully qualified and fast support process, all needed information must be provided directly during the first contact with Infineon.

See Customer request handout for the complete list of required information. It can be used as a blueprint to hand in support requests.

2.2 Procedure at Infineon

All external examinations, electrical characterizations, and other non-destructive tests must be completed before device decapsulation or performing other destructive tests.

Expendable control samples should be used to test “risky” analysis procedures before attempting them on the devices under evaluation. They have must be used while verifying test setups and electrical failures.

Table 2 Failure analysis procedure at Infineon

Responsible	Tasks
Quality assurance (QA)	New claim • Open new case in the MyCases system • Provide the LOT history
Product engineering (PE)	Pre-analysis • Visual inspection • Cleaning • Pin alignment • Pre-analysis on an eval board (check lifecycle stage[1], perform OpenRMA command[2], perform standard RAM march and galloping test) • Provide first intermediate report
Failure analysis engineering (FA)	Electrical failure analysis (EFA) • Standard outgoing test on ATE tester • DC test • Flash test • Operation margin tests • Provide intermediate 8D report • Provide final 8D report
Failure analysis engineering	Physical failure analysis (PFA) • Provide final 8D report

2.3 Reporting

2.3.1 Reporting tool

For each claim, a dedicated FA case must be opened in the 'MyCases' system at Salesforce.com. It is the customer relationship management tool used to track customer returns and communicate with the customer. It includes the history of communication between the customer and Infineon and collaborators who are involved in the FA process. The ownership of each case is assigned to a customer related QA representative.

2.3.2 Reporting schedules

The case closure target is 35 days (case opened to case closed). The target analysis stage cycle time to achieve this target is shown in Table 3.

These targets are general guidelines to help manage the analysis cycle time. The exact time at each stage will vary depending on the complexity from case to case.

[1] Customer should have ideally transitioned the device to "RMA" lifecycle stage before sending to Infineon. See section 3.4.

[2] OpenRMA command must to be performed after the lifecycle stage has been transitioned to "RMA". See section 4.2.

Overview of the standard FA flow

Table 3 Analysis stage cycle time

Stage	Target cycle time (days)	Stage owner	Remarks
TRIAGE/SHIPPING	3	Regional QA	This stage includes the initial contact with the customer, understanding the customer issue, and the assignment of a shipping address. Monitoring of the shipping time of the units to Infineon is the responsibility of the customer quality engineer (CQE). If the return does not arrive or the customer does not provide a shipment tracking number within five days, the case may be put on pending status for four additional days. If no units or a tracking number are received after nine days, the case may be cancelled after notification to the customer.
RECEIVING	1	FA	This stage includes receiving the units at Infineon and the initial non-destructive analysis such as visual inspection (ensure the part(s) received match the MPN / part marking provided in the SFDC case), X-ray, CSAM, and/or lead conditioning. This stage is owned by the failure analysis group.
Pre-analysis	1	PE	Pre-analysis is done in case the MCU is not directly sent to Infineon's backend to Failure analysis engineering to perform EOL test. It includes some basic tests which are dependent on debug access restriction settings. If possible, standard RAM (March, Galloping) tests can be done. At the minimum, the sample is checked whether the provided unlocking codes (certificate along with the signature) can be used for the OpenRMA command which is required to test the device on the ATE. It will also check whether the device is in DEAD state. Note that this stage might be skipped if the device is directly sent to Infineon's Failure analysis engineering where the device is tested against the test specification in a dedicated test mode.
Electrical fault isolation	7	PE/FA	This stage includes all electrical testing and electrical fault isolation on the returned unit(s). Initial testing should be performed using the current QA and production-level ATE programs. This should include any temperature testing that is done in the production flow or the temperature related to the customer's reported failure. All automotive returns must be tested at room, hot, and cold temperatures. In-depth electrical analysis is used to isolate the electrical failure location. This may include bitmap- and design-related debug to determine the failure's physical location within the die.
PFA	7	FA	This stage includes the physical analysis process to identify the defect causing the failure.

Overview of the standard FA flow

Stage	Target cycle time (days)	Stage owner	Remarks
MANUFACTURING 8D CAR[3]	10	CAR Owner	This stage includes the completion of the CAR issued to manufacturing or engineering. A CCAR should be issued when: • A customer returns a failing unit with a failure mode not covered by a prior CCAR. • The returned unit fails for a failure mode on material manufactured after a CA was implemented to eliminate the failure. This is considered a recurrence CCAR. • A new customer is affected by a known failure mode - each customer affected by a failure mode should get a CCAR assigned. Multiple returns for a single customer (same event) can be combined into the same CCAR. If a CCAR already exists, it is reasonable to request the CCAR owner to revise the CCAR with the newest customer failing quantity, lot trace analysis, containment actions, and risk assessment.
FINAL REPORT	1	FA	This stage includes gathering all the analysis data from the various groups (FA/PE/Apps) and writing a complete analysis report for the customer. The final report should be approved by the failure analysis director or designate. Upon completion, the report is posted to the case for the customer. This stage is owned by the failure analysis group with inputs from the application and/or product engineers.
CUSTOMER CLOSURE	5	Regional QA	This stage includes the final closure with the customer. This may include answering follow-up questions from the customer, translation of reports to the local language, or providing additional analysis details. This stage requires the customer's agreement to close the case. While waiting for the customer's response to the final closure, the case shall be put on pending status after two days.

Communication with the customer of the analysis findings and status will be provided on a regular time schedule in the form of CRM external interactions or interim reports until the analysis and corrective actions are completed. All communications will be sent to the customer using the CRM system.

- Cycle time begins at case creation
- Provide (as required) the shipping information to the Infineon analysis site as soon as possible
- 24 hours – Containment for IATF16949 CSR compliance only
- Notify the customer immediately of receipt of the customer's shipment at the Infineon analysis site
- 48 hours – Initial report
- 10 days – First interim report
- >10 days – The interval for sending a written response to the customer will not exceed 10 days.

[3] Corrective Action Request.

Overview of the standard FA flow

These targets are general guidelines to help manage the analysis cycle time. The exact time at each stage will vary on the complexity from case to case.

3 General information of TRAVEO™ T2G security concept

The TRAVEO™ T2G MCU introduces a new concept of device security, which aims to prevent the device from being hacked via direct access to the debug port, via direct connection to a communication port (such as SPI, I2C, CAN, LIN, UART, etc.), or by wireless connection such as OTA. In addition, it provides a verification process to ensure the authenticity of the firmware and to execute it only if it has not been corrupted.

There are various combinations how the device can be secured. This application note provides an overview of the most critical options and the subsequences during the FA process of a failing device. The customer must be aware that depending on the chosen software-related security configuration, EFA[4] on a claimed sample can be done in one of the following ways:

- Very detailed (full FA flow possible)
- Partly (FA flow possible with restrictions[5])
- Not at all (locked condition)

In all cases, X-ray, CSAM, and DC analysis are possible as non-destructive analysis methods.

If the claimed MCU is provided in the “RMA” lifecycle stage, all tests can be executed on the ATE test machine. If the transition to the “RMA” lifecycle stage is not possible, some investigations can be done if access to the device and to the memories and IPs is granted.

The device is completely locked if the “RMA” lifecycle stage cannot be entered and no access to the device can be granted via the debug interface.

3.1 Lifecycle stages

The most significant change in security functionalities compared to former MCU families is the introduction of different security levels based on the device lifecycle stage. In each stage, the accessibility and protection behavior of the device can be configured separately. Lifecycle stages are defined by the irreversible blowing of eFuses. The customer can define the security level.

Figure 2 shows the relevant lifecycle stages for TRAVEO™ T2G products.

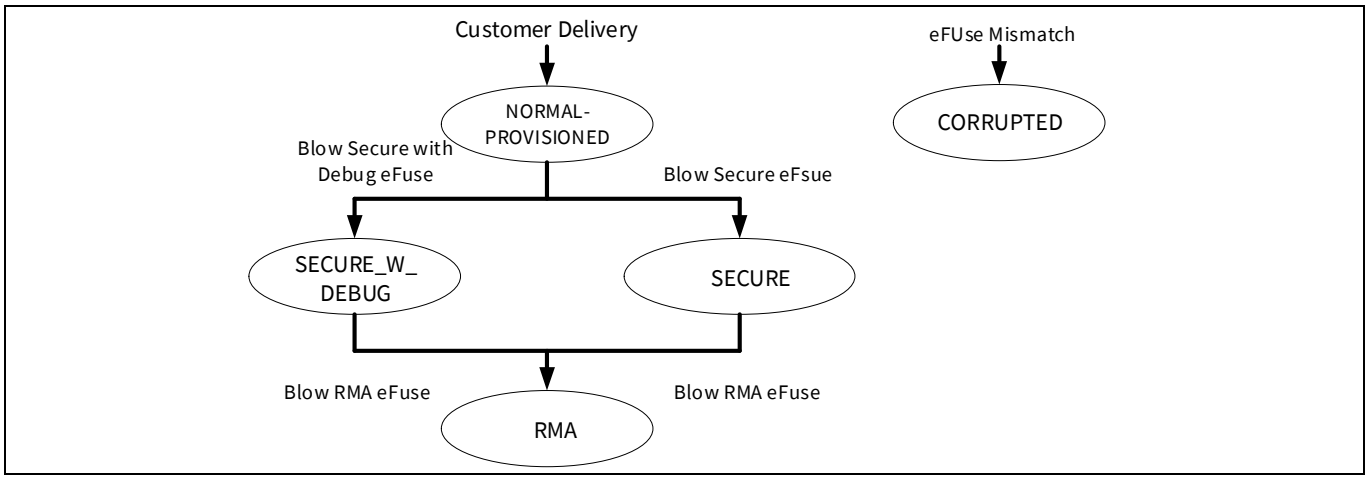

Figure 2 Device lifecycle stages

[4] Electrical failure analysis.

[5] Analysis possible only on the evaluation board if access to the device and its IPs and memories is granted.

3.1.1 “NORMAL_PROVISIONED”

This is the lifecycle stage of a device after trimming and testing are complete in the factory. All configuration and trimming information are complete. Valid flash boot code has been programmed in the SFlash. To allow boot of devices only with the ensured integrity of trims, flash boot, and other objects from the factory, a hash (SHA-256 truncated to 128 bits/SHAKE-128) of these objects is stored in eFuse. This hash is referred to as FACTORY_HASH. Customers receive parts in this lifecycle stage from Infineon.

3.1.2 “SECURE”

This stage is the life cycle stage of a “secure” device. Prior to transitioning to this stage, the SECURE_HASH must have been blown in eFuse and a valid application code must have been programmed in the code flash. In this stage, the protection state is set to “SECURE” and “SECURE” access restrictions are deployed. A “SECURE” device will boot only when the authentication of its flash boot and application code succeeds. The SECURE_HASH is calculated and written to the eFuse by the SROM firmware when transitioning to the “SECURE” or “SECURE_W_DEBUG” lifecycle stage from the “NORMAL_PROVISIONED” lifecycle stage. Customers should not receive parts in this lifecycle stage from Infineon.

3.1.3 “SECURE_W_DEBUG”

This stage is the same as the “SECURE” lifecycle stage, except the device allows debugging. Before transitioning to this stage, the SECURE_HASH must have been blown in eFuse and a valid application code must have been programmed in the code flash. In this stage, the protection state is set to “SECURE”, but NORMAL access restrictions are deployed to enable debugging. When there is an authentication failure during ROM boot or flash boot, SWD/JTAG pins are enabled, the protection state is set to “SECURE” but NORMAL access restrictions are deployed and is not transitioned to “DEAD” state. Transition from “SECURE_W_DEBUG” to “SECURE” is not allowed. “SECURE_W_DEBUG” parts are used only by the developers and software testers. Customers should not receive parts in this lifecycle stage from Infineon.

3.1.4 “RMA”

This stage allows performing FA. The customer transitions the part to the “RMA” lifecycle stage when FA by Infineon is required on the part (see section 3.4 for details). Infineon recommends erasing all the sensitive data in flashes before invoking the system call that transitions the part to “RMA”. Customers should not receive parts in this lifecycle stage from Infineon.

Before invoking the system call to transition to “RMA”, the customer must create an offline certificate that authorizes the transition of the part with a specific Unique ID to “RMA” lifecycle stage. The certificate will be signed by the customer using the same private key that is used in signing the user application image. The verification of the signature uses the same algorithm used by flash boot in authenticating the user application. The same public key (injected by the OEM) stored in SFlash is used for the verification. See section 4.2 for more information.

When a part is reset in the “RMA” lifecycle stage, the boot will set the access restrictions such that the DAP has access only to the System AP, IPC, and MMIO registers for making system calls, and adequate RAM for communication. It will then wait for the “OpenRMA” system call from the DAP along with the certificate of authorization. The boot process will not initiate any firmware until it successfully executes the “Open-RMA” command. After the command is successful, the lifecycle stage stored in eFuse cannot be changed from “RMA”. Every time the part is reset, it must execute the “OpenRMA” command successfully before the part can be used. To execute the "OpenRMA" command, the customer must provide the certificate signed by the customer using their private key to Infineon. This certificate is different from the one used for transitioning the device to “RMA”

stage. If sending the device(s) to Infineon, the device(s) must be traceable to each individual OpenRMA certificate.

Note that the "OpenRMA" system call is executed by Infineon during the pre-analysis process using the provided certificate. In addition, it is used while performing the root cause analysis during the NTF flow.

3.1.5 CORRUPTED

The device is in this lifecycle stage if a read error is detected when reading the eFuse bits that determine the lifecycle stage. The device will enter the DEAD protection state and only IPC MMIOs can be read via SYSTEM-AP. No other accesses are allowed.

3.2 Authentication

TRAVEO™ T2G MCUs follow the chain of trust (CoT) concept. During the boot execution, different areas of the non-volatile memory are validated to fulfill the highest standard of security. Code is executed only if the validation process is passed. If a mismatch occurs, the device enters the DEAD protection state, remains in the boot code and in an endless loop, and application code is not executed.

After RESET, the CM0+ core executes the boot sequence in the following order:

- ROM boot: Validation of SFlash is done by the HASH calculation and comparing to the FACTORY_HASH (NORMAL and “SECURE" protection states) or SECURE_HASH (“SECURE” protection state). Both hash values are stored in eFuses. In case of mismatch, the ROM boot code remains in an endless idle loop.
- Flash boot: Validation of the “secure” image. A signature is calculated by the RSA-2048 encrypting process based on the public key and the code flash memory content (“secure” image code). This signature is compared with the “secure” image digital signature which was defined in a separate flash area during the design phase. If a mismatch occurs, the flash boot code remains in an endless idle loop. The flash area to be authenticated is defined in TOC2.

Note: For RSA 2048, 3072, and 4096 support, see the device-specific datasheet (under the Part Number/Ordering Code Nomenclature section, Hardware option).

Authentication can be enabled in TOC2. The state must be provided to Infineon.

Note: If authentication fails and device enters the DEAD protection state, the lifecycle stage cannot be transitioned to “RMA”. Failure analysis will be restricted. Test mode cannot be entered, and standard analysis cannot be done on ATE test machine.

One way to solve this issue is to have a second application software whose only job is to allow the transition to the “RMA” lifecycle stage. The start address is defined in TOC2. If the first application software fails, this second application can be loaded by flash boot, which can then allow the transition to “RMA”. Of course, if authentication of the second application software also fails, there is no option to change to the “RMA” lifecycle stage.

There might also be a requirement to erase all sensitive or proprietary code stored in the device before transitioning to “RMA”. When authentication is enabled for “secure” image, the device transitions to the DEAD state by erasing the “secure” image or digital signature for authentication. As a result, you cannot transition to the “RMA” lifecycle stage. If you need to erase your code, you will need to prepare and write a signed dummy code and digital signature.

Note: *If the device already has a second application for "RMA" management, the first application software can be erased if it has sensitive data. The second application can be launched by programming a dummy application header (if needed) for the first application. See the application note "Secure system configuration in TRAVEO™ T2G family" for details.*

3.3 Access restrictions

TRAVEO™ T2G devices support various sets of configurations for accessing the debug port and internal components. Depending on this setup, enhanced FA on evaluation board for pre-analysis process can be restricted. The configuration must be provided by the customer before creating a new FA case. Note that it is the customer QA's responsibility to contact the project development team to collect all required information.

3.3.1 Debug access port (DAP) configuration

The DAP consists of a combined SWD/JTAG interface. The debug port connects to one of the three access ports (AP):

- CM0-AP: Access to CM0+ internal debug components and to the rest of the system via the AHB master interface. This provides the debug host the same view as an application running on the CM0+ core.
- CM4/7-AP: Access to the CM4/7 internal debug components. The CM4/7-AP also allows access to the rest of the system through the CM4/7 AHB/AXI master interfaces. This provides the debug host the same view as an application running on the CM4/7 core. Additionally, the CM4/7-AP provides access to the debug components in the CM4/7 core through the external peripheral bus (EPB). These debug components can also be accessed by the Cortex-M4 CPU but cannot be reached through the other APs or by the Cortex-M0+ core.
- SYSTEM-AP: Access to the rest of the system through an AHB mux. This allows access to the system access port ROM table (SYSAP_ROMTABLE), which is not intended to be reached any other way. The system ROM table provides the chip ID but is otherwise empty. In addition, the SYSTEM-AP is also protected by a MPU. Different peripheral IPs can be separately secured through the DAP MPU structures.

For security reasons, all three APs have their own access restrictions and can be independently disabled. In addition, the configuration can be defined for each protection state: Normal, "Secure", and Dead. Each of these access restrictions may be configured by the user. The NORMAL access restrictions and normal DEAD access restrictions are stored in SFlash, but "SECURE" access restrictions and "SECURE" dead access restrictions are stored in the one-time programmable eFuse.

The access restrictions are evaluated and applied during boot process through ROM/Flash boot.

There are three options for the initial AP configuration:

- Enable AP: Always accessible
- Disable AP: Can be enabled by code execution
- Permanently disable AP: Completely locked

Note: *In the "RMA" lifecycle stage, all access ports are always enabled if the OpenRMA command was executed successfully and full access to flash, Work Flash, SRAM, and MMIO is granted. This means that during the pre-analysis, standard RAM (March and Galloping) and flash tests can be executed. Table 3 provides an overview of the access restrictions in different lifecycle stages and how they can be configured.*

Table 4 Access restrictions overview

Lifecycle stage	Access restrictions	How to configure access restrictions
“NORMAL_PROVISIONED”	Normal access restrictions, stored in SFlash	Write to the SFlash row 13 using the “WriteRow” system call. Note that the access restrictions once configured cannot be widened.
“SECURE_W_DEBUG”	Normal access restrictions, stored in SFlash	Write to the SFlash row 13 using the “WriteRow” system call. Note that the access restrictions once configured cannot be widened. Writing to SFlash must be done before transitioning the device to “Secure with Debug”.
“SECURE”	“Secure” access restrictions, stored in eFuse	Access restrictions specified in the “TransitionToSecure” system call
“RMA”	Special access restrictions until OpenRMA is executed successfully. No access restrictions after OpenRMA is successful.	Not applicable

3.3.2 Access restrictions to internal components

The restrictions define how the access ports need to behave. If system access port MPU is additionally enabled, the settings in this table will be applied to define how the DAP can access these resources via system AP.

Table 5 Overview of access restriction configurations via SYSTEM-AP MPU

Component/memory	Options
FLASH	Only a portion of the flash starting at the bottom of the area is exposed. • Entire region • Seven-eighth • Three-fourth • One-half • One-eighth • One-sixteenth • No access
SRAM	Only a portion of the SRAM starting at the bottom of the area is exposed. Uses the same encoding as for flash.
WORK_FLASH	This field indicates what portion of the work flash is accessible through the system access port. Only a portion of the work flash starting at the bottom of the area is exposed. • Entire region • One-half • One-quarter • No access

Component/memory	Options
SFLASH	This field indicates what portion of the flash supervisory region is accessible through the system debug port. Only a portion of the SFlash starting at the bottom of the area is exposed. • Entire region • One-half • One-quarter • No access
MMIO	This field indicates what portion of the MMIO region is accessible through the system debug port. Encoding is as follows: • All MMIO registers • Only IPC MMIO registers accessible (system calls) • No MMIO access

The configuration set for the SYSTEM-AP MPU depends on the current lifecycle stage.

“NORMAL_PROVISIONED”/“SECURED_W_DEBUG”: MPU settings defined in SFlash (row 13).

“SECURE”: MPU settings defined in eFuses.

Note that in the “RMA” lifecycle stage after successful execution of the OpenRMA command, full access to the device is granted independent of the MPU configuration. However, note that if other internal protection units (like SMPU or PPU) are configured through the currently executing application/HSM Software, access to various resources can still be blocked.

3.3.2.1 Access restrictions to flash, eFuse, SRAM, and MMIO

In addition to the MPU restrictions, there are two further configuration options to define the access to flash, SRAM, and MMIOs.

Software protection unit (SWPU)

The flash/eFuse permissions can be set by SWPU which is stored in SFlash. It prevents malicious or unintended modification of the flash or eFuse, and reading of customer-specific, sensitive eFuse data.

Shared memory protection unit (SMPU)

The SMPU is shared by all bus masters. It distinguishes between different protection contexts; it also distinguishes “secure” from “non-secure” accesses and user mode from privileged mode accesses.

The access restrictions can be defined different for TRAVEO™ T2G derivatives:

- B-E (body-entry)
- MMIO
- All memory (SRAM access, FLASH read)
- B-H/C (body-high and cluster)
- CM0+: same as B-E
- CM7: All memory

Note: *In CM7-based devices, SMPUs cannot be used for protecting peripherals (MMIOs) against CM7 accesses.*

PPU

The PPUs are situated in the peripheral block and are associated with a peripheral group (peripherals with a shared AHB-Lite bus infrastructure). A PPU is shared by all bus masters. The PPU distinguishes between different protection contexts; it also distinguishes “secure” from “non-secure” accesses and user mode from privileged mode accesses.

See the TRAVEO™ T2G architecture technical reference manual (TRM) for more information.

3.3.2.2 Access to debug interface (DAP)

To allow access to any access port, the port pins of debug access port (DAP) must be configured correctly.

There are two DAP security configurations which define the debug port pins. Note that debug port pins are shared port pins and can also be used for other functionalities. Do the following to enable the debug port as part of the TRAVEO™ T2G internal boot process:

- At least one access port (CM0+-AP, CM4/7-AP, SYSTEM-AP) must be enabled.

AND

- TOC2 configuration in SFlash: TOC2_FLAGS.SWJ_PINS_CTL set to ‘2’ (Enable SWJ pins in Flash boot). Note that SWJ pins may also be enabled in the user code.

The configuration of the debug interface is done during the TRAVEO™ T2G internal boot process, where depending on the present lifecycle stage, the configuration in SFlash or eFuses is evaluated and the port pins are set accordingly.

In the user code, the connection status of a debugger can be read in the SWJ_CONNECTED bit of the CPUSS_DP_STATUS register.

Note that for various reasons (like authentication failure or hash mismatch), the TRAVEO™ T2G device can enter DEAD state from boot. DEAD access restrictions (DAR – Normal_DAR or Secure_DAR) will then decide how an external debugger can have access to the device. If there is a failure during ROM boot execution, the DAP pins are always configured by ROM boot. If there is a failure during flash boot execution, the DAR (Normal_DAR or Secure_DAR) defines whether the DAP pins need to be configured (i.e., the DAR should allow access to at least one access port if DAP pins need to be configured by boot). As a recommendation, the DAR can be configured in such a way that the system access port is enabled and access for IPC MMIOs are granted. This will then allow the debugger to read the possible reason for DEAD state entry from either of the boot processes.

3.3.3 System call requirements

To execute a lifecycle stage transition, a system call is required which is writing the data related to the system call into the SRAM to set the SRAM scratch address in the IPC data register and to trigger the system call by writing to the IPC notification register. See section 4.2 for an example using the OpenRMA system call.

Note that SRAM access restrictions are set in the MPU and SMPU settings.

After the system call is initiated, the related API is called in the SROM by an interrupt handler. The related API is executed by the CM0+ core. To allow a proper interrupt execution, IRQ0 and IRQ1 of CM0+ must be enabled and the correct priorities must be set in the user code. In addition, the interrupt vector table must be prepared accordingly. See the technical reference manual for details.

System calls can be triggered by CM0+, CM4/7, or DAP.

Depending on the TRAVEO™ T2G family, the following inter-processor communication (IPC) structure must be considered:

TRAVEO™ T2G B-E

- CM0+ access: IPC structure 0
- CM4 access: IPC structure 1
- DAP access: IPC structure 2

TRAVEO™ T2G B-H / Cluster (not all cluster derivates have two CM7 cores)

- CM0+ access: IPC structure 0
- CM7_0 access: IPC structure 1
- CM7_1 access: IPC structure 2
- DAP access: IPC structure 3

Thus, IPC structure 2 should be used by DAP in CM4-based devices and single CM7-based devices, while IPC structure 3 should be used for dual CM7-based devices for triggering system calls.

3.4 Lifecycle stage transition to “RMA”

To allow full access to the device and to fulfill the TRAVEO™ T2G FA requirements, the customer must ensure performing a lifecycle stage transition to “RMA”. As shown in Figure 2, there is a strict sequence of lifecycle stage transitions. For devices in “NORMAL_PROVISIONED” lifecycle stage (delivery condition), first a transition to “SECURE” or “SECURE_W_DEBUG” lifecycle stage must be performed.

Depending on the access restrictions mentioned in section 3.3, an implementation must be applied to the application software which allows performing a lifecycle transition. Figure 3 illustrates how the transition to “RMA” lifecycle stage can be triggered by the user via a defined interface (e.g., CAN diagnosis tool). This procedure requires a failing device which is fully functional (in stable communication to external I/Os is possible and all required software code can be executed). Note that cyclic resets in a very early stage of software execution (start-up code) may not allow triggering a “RMA” lifecycle stage transition. Certificate 1 is used for transition to the “RMA” lifecycle stage. It is different from the Certificate 2, which is used for the OpenRMA command (see also section 4.2).

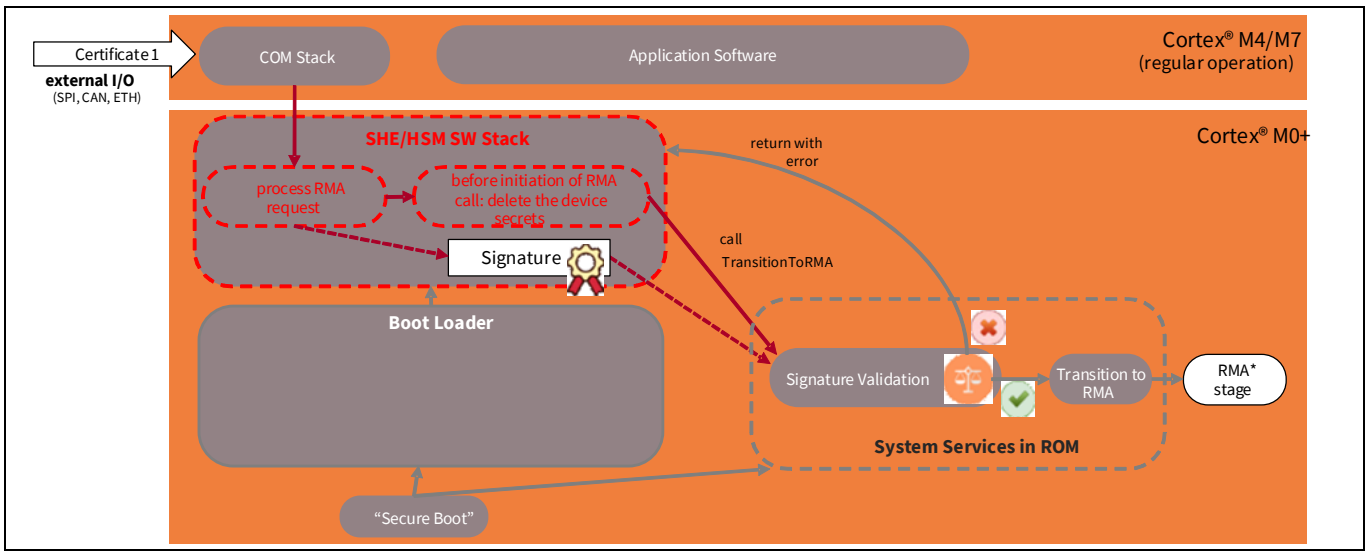

Figure 3 Transition to "RMA" lifecycle stage (option 1) via standard communication interface (e.g., CAN, Ethernet)

*Note that the device cannot be transitioned from "RMA" lifecycle stage to "NORMAL_PROVISIONED" or any other lifecycle stage because of use of eFuse cells for storing the lifecycle stage.

If no function for the lifecycle transition is implemented OR cannot be executed for various reasons, the transition can also made via the debug interface (JTAG/SWD). Consider the access restrictions described in section 3.3 to allow performing the system call for lifecycle stage transition. Figure 4 illustrates how the transition to "RMA" lifecycle stage can be triggered in a very early stage of software execution (start-up code) via an interface such as JTAG. The advantage of this procedure is that the probability to get the system into a deadlock condition (e.g., cyclic resets) is minimized and the device can be prepared for FA process to analyze a possible hardware failure of the device. Keep in mind that only the Arm® Cortex® M0+ core must be used for this option.

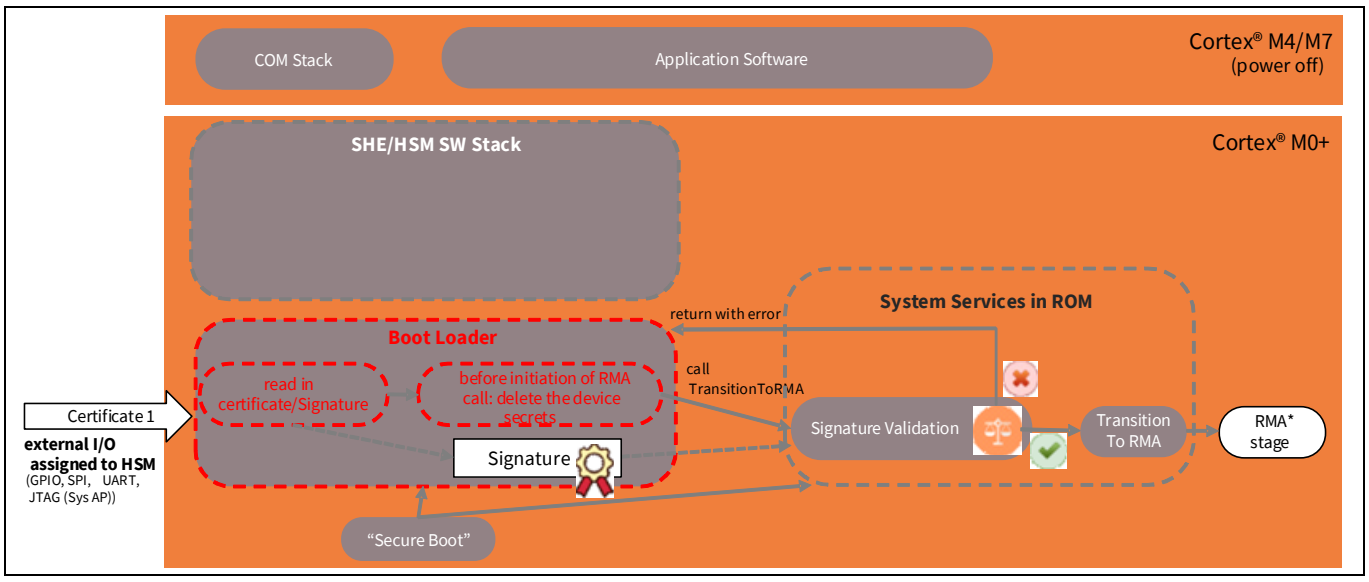

Figure 4 Transition to "RMA" lifecycle stage (option 2) via dedicated I/O (e.g. JTAG)

*Note that the device cannot be transitioned from the “RMA” lifecycle stage to “NORMAL_PROVISIONED” or any other lifecycle stage because of use of eFuse cells for storing lifecycle stage.

Note that it is recommended to trigger transition to “RMA” from DAP. See section 7.1 for details.

For guidelines for software implementation for lifecycle stage transition, see Appendix H 16.1, “Sample code” of AN228680.

3.5 Authenticated debugger techniques

In a secured system, non-authorized access to the MCU is prohibited. For FA purpose, access to the target is required to halt the cores and to allow the debugging of MMIO IPs and memories. In the “RMA” lifecycle stage, this is possible after a successful OpenRMA command. In other lifecycle stages, access restrictions defined in SFlash or in eFuses define the initial accessibility to internal modules via the debugging interface (JTAG/SWD). To allow such access, the restrictions can be recovered by the user software code.

The implementation can provide for a concept in which the external access to the SYSTEM-AP and the associated IPC registers can trigger a debug mode that must then be configured by the customer application software which can then enable the core APs. It will allow entering a unique authorization key. If this key is valid, access restrictions defined in the MPU, SMPU, SWPU, and PPU can be defined to allow partial or full access to the target MCU.

See Appendix H 16.1.1 of AN228680 for detailed information where the debugging capabilities are enabled after a valid 128-bit key is provided via debug interface.

4 Reuse a sample in the NTF process

The ‘No Trouble Found’ (NTF) process is initiated after all standard tests have passed at Infineon. Combined with the final 8D report, the claimed sample is sent back to the TIER1 customer to continue the root cause analysis.

The customer must solder the sample on a known good PCB and the application software must be reprogrammed into the code flash of the MCU. Depending on the nature of the failure, different tests will be done to prove whether the original failure is still reproducible. This need to be decided by the customer based on the original failure description and their analysis reports.

For TRAVEO™ T2G devices, a strict handling is required which allows testing the device in the same way as the device was in production state. Note that this procedure is different compared to legacy MCU families from Infineon. If the NTF process is continued in customer’s production plants, different items must be considered and prepared to allow a regular handling of devices in the “RMA” lifecycle stage.

Figure 5 illustrates the test bench to be used during the NTF process when the MCU is in the “RMA” lifecycle stage. After each RESET, a script must be executed to send the OpenRMA command to the microcontroller. Because the correct data are transferred, the boot process is completed and customer code can be executed. The OpenRMA command must also be executed to allow flash programming.

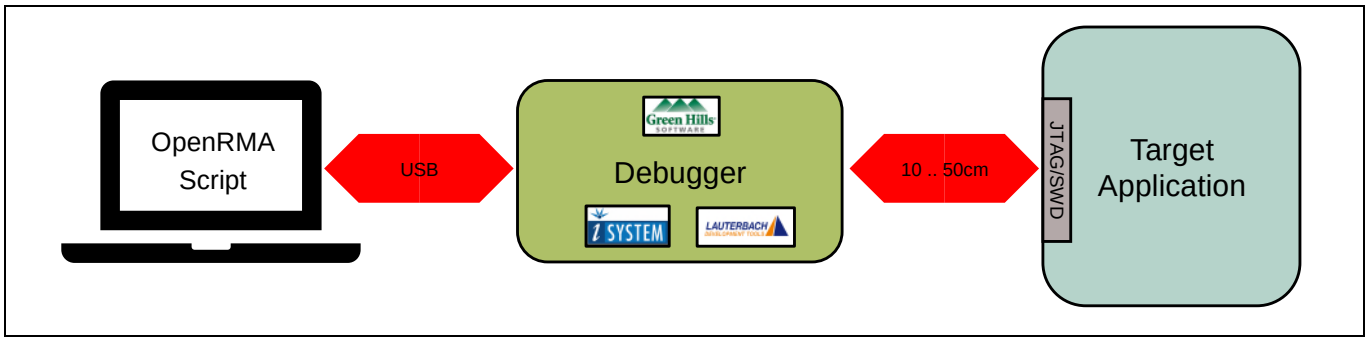

Figure 5 Test bench setup for the OpenRMA procedure

4.1 Flash programming and application execution in the “RMA” lifecycle stage

After the device was transitioned from “SECURE” or “SECURE_W_DEBUG” to the “RMA” lifecycle stage, the behavior of the MCU is different. In the “RMA” lifecycle stage, the MCU waits in an endless loop for an OpenRMA command, which allows to execute any code in flash. A dedicated protocol must be sent to the device via the JTAG/SWD interface including a certificate and the signature. This procedure unlocks the device.

Note that this procedure is required to be done after each RESET.

The certificate and the related signature are the same codes as already provided to Infineon.

Table 6 “RMA” certificate format

Object	Bytes
Object Size	4 bytes: 0x00000014
Command ID	4 bytes TransitiontoRMA: 0x120028F0 OpenRMA: 0x120029F0
Unique ID	11 bytes
Zero Padding	1 byte
Digital Signature	256 bytes

Note that the certificates and digital signatures for TransitionToRMA and OpenRMA are different because the command ID is different.

4.2 OpenRMA procedure

The execution of the OpenRMA command is mandatory for FA procedures. After any RESET, a script must be executed which sends the OpenRMA protocol to the MCU. After this is successfully done, all operations on the MCU can be done including flash programming.

The OpenRMA command is a system call executed by accessing the System-API. For running the OpenRMA API, the correct certificate and digital signature are required. See Appendix H 16.1, “Sample code” of the application note, AN228680. Figure 6 illustrates how the device can be opened with the OpenRMA command. It requires Certificate 2 which is different from Certificate 1 that is used for the transition to “RMA” lifecycle stage (see also section 3.4).

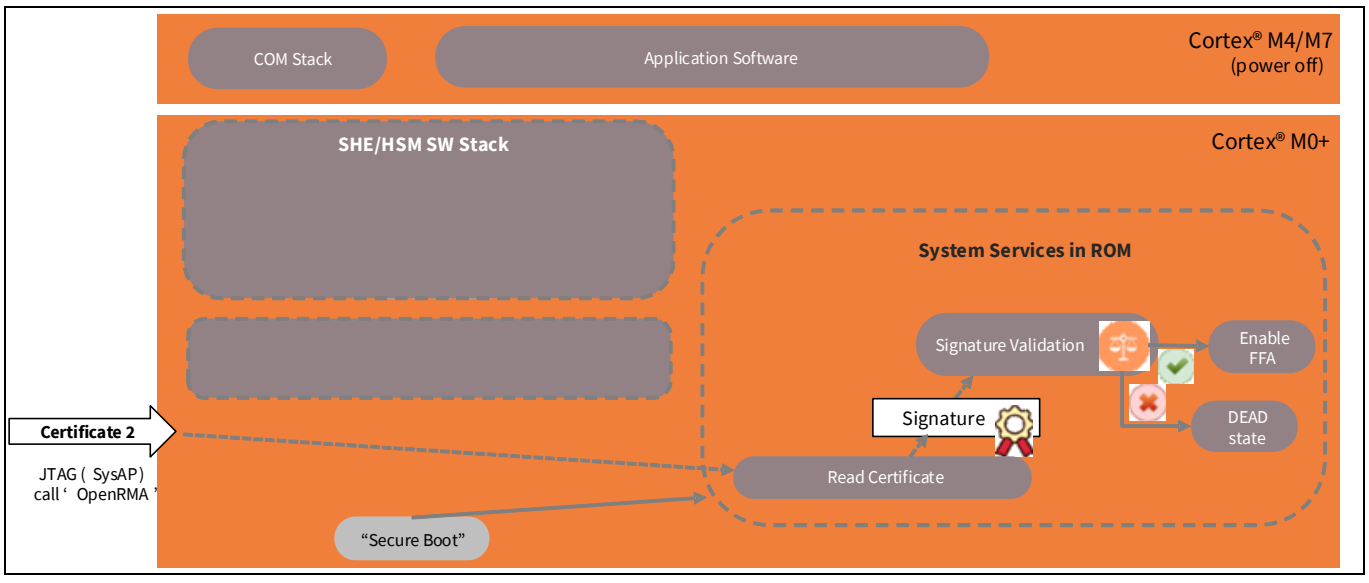

Figure 6 Opening the device in RMA lifecycle stages

Note that in the “RMA” lifecycle stage, IPC reserved for DAP (IPC2 for B-E device, for example) must be used to trigger the OpenRMA system call. Also, 1/16th of SRAM0 can be accessed until the OpenRMA command is successfully executed. When using the OpenRMA API, the parameters such as certificate and digital signature must be placed as follows:

- Devices with SRAM0 size larger than 64 KB: parameters must be placed from [SRAM0 start address + 4 KB] to [SRAM0 start address + 1/16 of SRAM0 size].
- Devices with SRAM0 of 64 KB or less: parameters must be placed within 600 bytes from [SRAM0 start address + 2 KB]. The certificate and signature addresses are 24 bytes, and digital signature is 512 bytes. (RSA-4K).

In the example shown in section 4.2.2, a device with the SRAM0 size greater than 64 KB is considered.

4.2.1 Step1: preparation of the certificate and signature

Like the transition to “RMA”, the first step is to generate the certificate which needs the following information:

- Object size (0x14 bytes)
- OpenRMA command ID (0x120029F0)
- 12-byte Unique ID[6] (11-byte ID and 1-byte zero padding)
- Digital signature

The signature and the “RMA” certificate can be generated in Linux environment (e.g. Git Bash). Figure 7 shows a script which creates the certificate and the signature based on the private key, which is associated with the same public key stored in SFlash, which was used for authenticating the application in “SECURE” mode.

```
echo 14000000 f0290012 1ab09707 0a350f00 23057000 | xxd -r -p > _data.bin
     Object Size  Command ID   Unique ID   Zero Padding

openssl dgst -sha256 -sign rsa_private.txt _data.bin > _signature_openrma.bin

cat _signature_openrma.bin | xxd -p -c 64 > _tmp1.hex

echo 00000029 > _tmp2.hex
     RMA OP Code

echo 14000000 f0290012 1ab09707 0a350f00 23057000 >> _tmp2.hex

cat _tmp1.hex >> _tmp2.hex

sed -r "s/\s*(\w{2})(\w{2})(\w{2})(\w{2})/0x\4\3\2\1\n/g" _tmp2.hex | sed -r
"/^$/d" > _output_openrma.hex

rm _data.bin _tmp1.hex _tmp2.hex
```

Figure 7 Create RMA certificate and signature with Git Bash

Note that this certificate and signature must be provided to Infineon. It will be proved in pre-analysis for correctness and to open the device for first analysis steps (for example, RAM tests).

[6] See AN228680 (Secure system configuration in TRAVEO™ T2G family) and the TRM for details on obtaining the device unique ID.

4.2.2 Step 2: definition of system call data in related locations in the SRAM

The OpenRMA API requires to store a parameter list, certificate, and signature in two separate areas in the SRAM. The start address of the SROM API parameter list (*SRAM_SCRATCH_ADDR*) is handled by the IPC data0 register. The area where the signature is stored in the SRAM is defined by a pointer that is part of the parameter shown in Figure 9. Both data fields must be downloaded to the SRAM during the execution of the OpenRMA script. Figure 8 illustrates an example for possible locations in the SRAM for the OpenRMA data used for the OpenRMA system call.

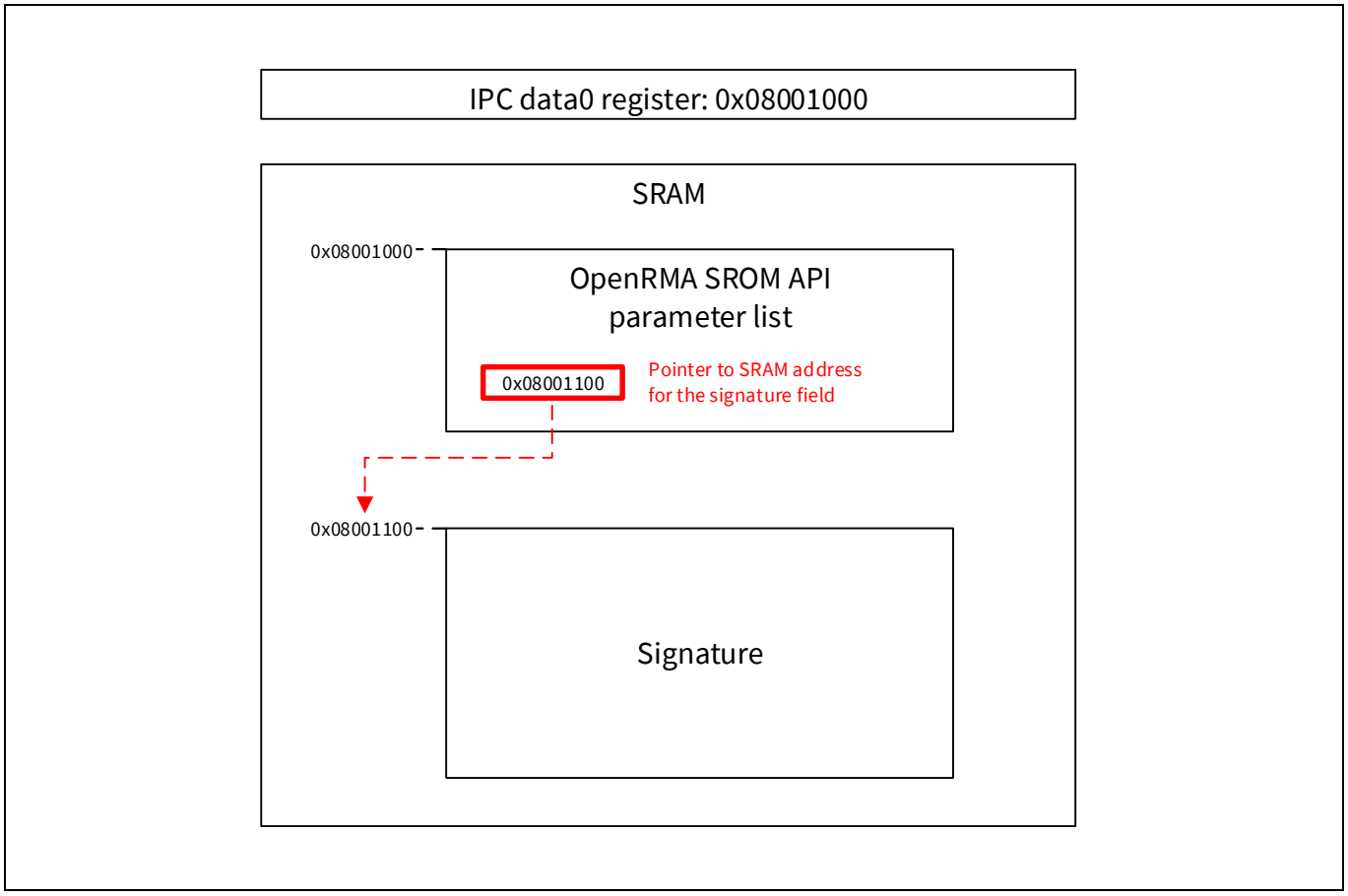

Figure 8 Certificate and signature fields located in SRAM

Note that the start addresses in the SRAM can be defined by the user.

Based on the certificate file, two further files must be created manually as the location in SRAM is customer-specific. Figure 10 shows how the files are built.

- *"open_rma_sign.txt"*
  This file consists of the signature (256 bytes) only.
- *"open_rma_parameter_list.txt"*
  The OpenRMA API in TRAVEO™ T2G requires a dedicated parameter list to be stored in the SRAM. Along with the certificate data, some additional information is required in the following format:

Reuse a sample in the NTF process

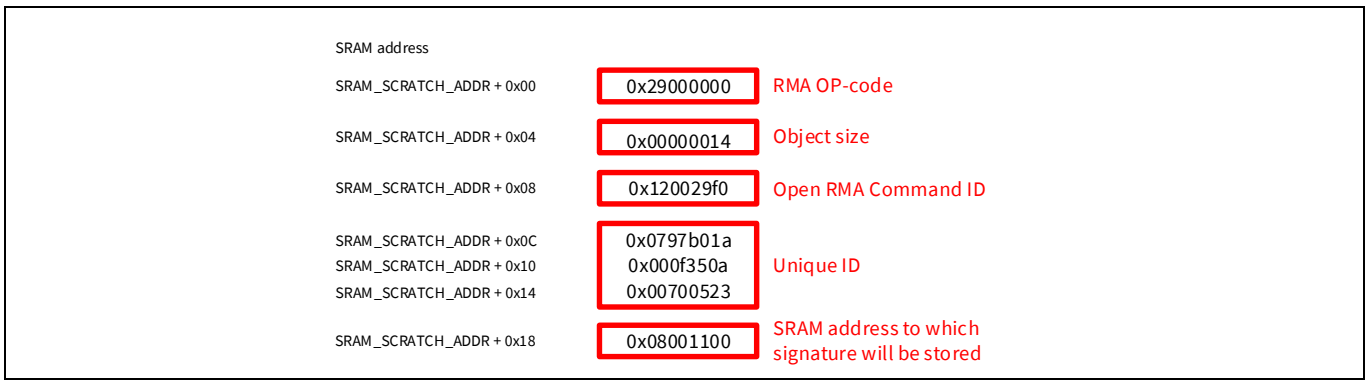

Figure 9 Format of SROM API parameters for the OpenRMA system call

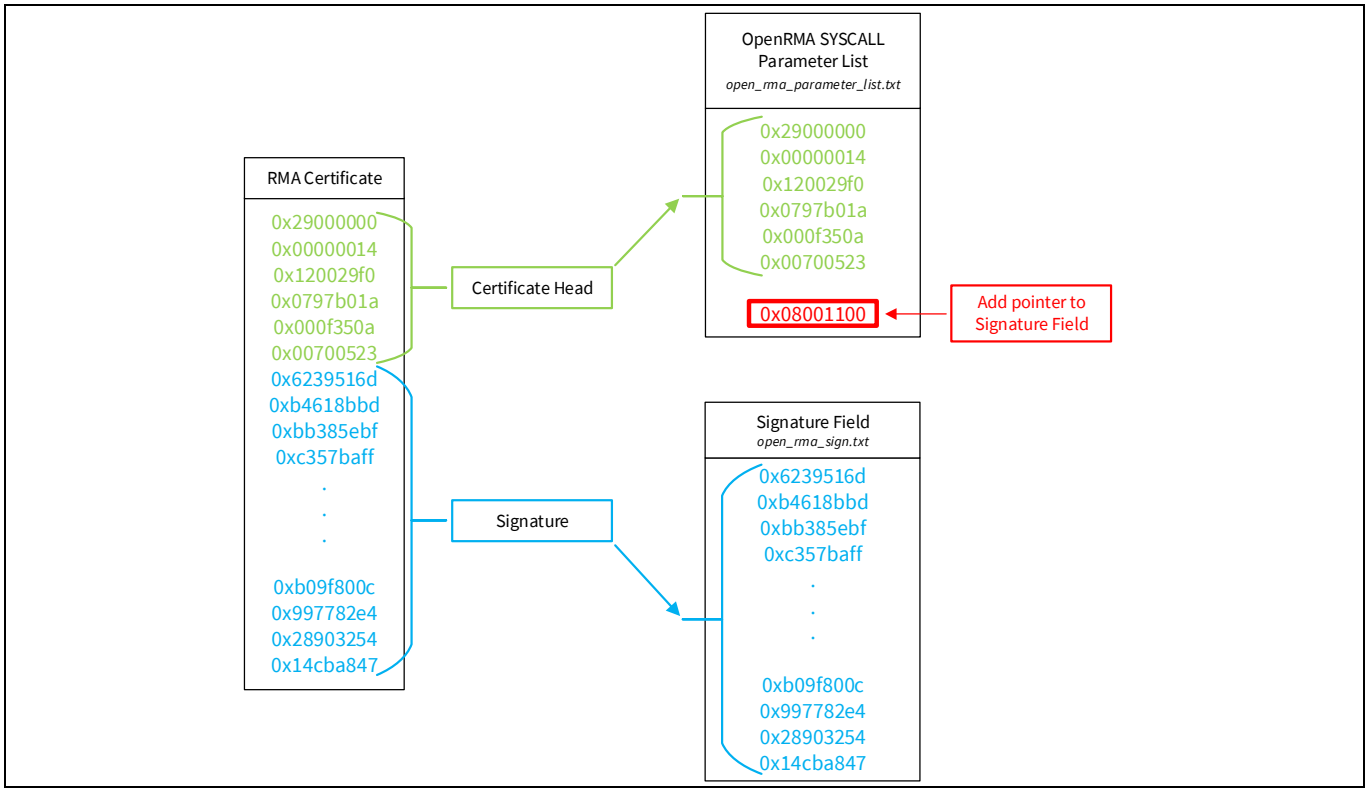

Figure 10 Creation of system call parameter list and signature field

4.2.3 Step 3: execution of the OpenRMA script

Now it is time to execute the OpenRMA script. In the following example, the script is called within a GHS MULTI debugger session. After connecting to the target MCU via a debugger session, the customer application code will not be executed. The error “Read Command Not successful” will be reported. This is because ROM boot waits in a loop until the OpenRMA system call is executed successfully. Also, if the core is tried to be halted, the error “Process won’t halt” will be reported.

Figure 11 shows the sequence to trigger the OpenRMA system call.

1. Download the certificate to the start address *SRAM_SCRATCH_ADDR* in the SRAM.
2. Download the signature to the start address defined in the parameter list of the OpenRMA system call at SRAM_SCRATCH_ADDR + 0x18.
3. Write *SRAM_SCRATCH_ADDR* to the IPC_data0[7] register.
4. Generate a notification event by writing to the IPC_NOTIFY register.

```
inFileName = sys.argv[1]
targetAddress = int(sys.argv[2], 0)
returnAddress = int(sys.argv[2], 0)

IPC2Data0Address = 0x4022004C                       # IPC2 register addresses used for TVII-B-E
IPC2NotifyAddress = 0x40220048
IPC2AcquireAddress = 0x40220040

NotifyNo = 0x00000001

WriteViaSysAp(IPC2AcquireAddress, 0)                # aquire IPC channel

WriteViaSysAp(IPC2Data0Address, targetAddress)      # write SRAM_SCRATCH_ADDR into
                                                    # IPC data0 register

inFile = open(inFileName,"r")

fileLines = inFile.readlines()

for line in fileLines:                              # write OpenRMA certificate into SRAM
   valueToBeWritten = int(line.rstrip("\n"),0)
   WriteViaSysAp(targetAddress, valueToBeWritten)
   targetAddress += 4
inFile.close()

inFileName_sig = sys.argv[3]
targetAddress_sig = int(sys.argv[4], 0)
returnAddress_sig = int(sys.argv[4], 0)

inFile_sig = open(inFileName_sig,"r")

fileLines_sig = inFile_sig.readlines()              # write signature data into SRAM
for line in fileLines_sig:
   valueToBeWritten_sig = int(line.rstrip("\n"),0)
   WriteViaSysAp(targetAddress_sig, valueToBeWritten_sig)
   targetAddress_sig += 4
inFile_sig.close()

WriteViaSysAp(IPC2NotifyAddress, NotifyNo)          # set notify bit in IPC notify register
                                                    # to trigger OpenRMA syscall
```

Figure 11 OpenRMA script UserDataLoadToRam.py for GHS MULTI debugger

The script is called in GHS MULTI debugger by using the following command line option:

[7] IPC structure is dependent on the TRAVEO™ T2G derivative.

Reuse a sample in the NTF process

```
python UserDataLoadToRam.py open_rma_cert.txt 0x08001000 open_rma_sign.txt
0x08001100
```

The cores can be halted after the OpenRMA system call is executed successfully.

4.2.4 Verification of a successful OpenRMA system call

To verify the proper execution of the OpenRMA system call, you can use the status stored by the OpenRMA API at *SRAM_SCRATCH_ADDR* in the SRAM.

Table 7 System call status codes

Status code	Description
0xA0000000	OpenRMA system call successfully completed
0xA0000009	OpenRMA system call in progress
0xF00000F1	SRAM_SCRATCH is protected
0xF00000B3	Invalid unique ID is passed during OpenRMA system call
0xF00000B4	Invalid signature found during OpenRMA system call

4.3 Required hardware

To perform the OpenRMA command, a dedicated test bench setup is required. TRAVEO™ T2G MCU supports JTAG and SWD interfaces. They cannot be used in parallel because the MCU port pins are shared. One of these interfaces must be available on the application hardware through a dedicated connector. The SWD listener decides whether the JTAG interface (default) or SWD interface is active.

The OpenRMA script is executed on a host which is connected to a debugger.

The following common debugger tools for TRAVEO™ T2G products are used in the automotive industries:

- Lauterbach – Trace32
- GHS – Green Hills Probe (Probe v3, v4, Super Trace Probe v3)
- iSystem – For example, ic5500

Note that depending on the clock speed used for the JTAG/SWD interface, the cable length from the debugger hardware to the target application must be chosen properly. Read the instruction manual from the debugger system for more details. Also consider the physical requirements for testing the unit in a climate chamber.

4.4 Reset sources

The OpenRMA command sequence must be applied to the MCU after each reset (e.g., power-on reset, external RESET XRES, software reset (triggered by the AIRCR core register), watchdog reset, etc.).

5 Restrictions on performing the standard FA flow

TRAVEO™ T2G MCUs provide highest security standard. The customer has a broad range of configurable access restrictions that have a direct impact to the testability of a device. This must be considered for a failing device that need to be analyzed during the FA flow. A minimum requirement to perform the standard FA flow is to perform a transition to “RMA” lifecycle stage before a device is sent to Infineon.

5.1 Matrix of possible security combinations

Table 8 shows a matrix which defines the possible FA options for TRAVEO™ T2G MCUs. It shows which tests can be done on the evaluation board and on ATE test machine considering the security settings used. It assumes a typical secured system where all the access ports are temporarily disabled, and the system access port is re-enabled by the user application as explained in section 7.1.

Note that for testing the device in other than “RMA” lifecycle stage, the access to peripheral is granted only if the corresponding IP is enabled in the MPU-MMIO settings.

The boot parameters mentioned are enabled during the boot process and may be overwritten by the user software. However, note that if access ports are locked permanently as part of access restriction configuration, reconfiguration is not possible. Also, DAP MPU structures cannot be reconfigured through the user software if initially configured by ROM/flash boot as part of access restriction configuration.

Table 8 Testability matrix

Lifecycle stage	DAP SYSTEM-AP (SFlash/eFuses)	MPU of SYSTEM-AP MMIO - IPC	MPU of SYSTEM-AP SRAM	SMPU - SRAM	MPU of SYSTEM-AP FLASH	SWPU FLASH	Certificate provided	Transition to “RMA”	Testability on ATE tester	Testability on eval board
NORMAL PROVISIONED	–	–	–	–	–	–	–	n	–	n
	n	–	–	–	–	–	–	n	n	n
	–	n	–	–	–	–	–	n	n	n
	y	y	y	y	–	–	–	n	n	SRAM tests possible
	y	y	n	–	–	–	–	n	n	No SRAM test
	y	y	–	n	–	–	–	n	n	No SRAM test
	y	y	–	–	y	y	–	n	n	FLASH tests possible
	y	y	–	–	n	–	–	n	n	No FLASH test
	y	y	–	–	–	n	–	n	n	No FLASH test
“SECURE”/“SECURE_W_DEBUG”	–	–	–	–	–	–	–	n	n	n
	n	–	–	–	–	–	–	n	n	n

Restrictions on performing the standard FA flow

Lifecycle stage	DAP SYSTEM-AP (SFlash/eFuses)	MPU of SYSTEM-AP MMIO - IPC	MPU of SYSTEM-AP SRAM	SMPU - SRAM	MPU of SYSTEM-AP FLASH	SWPU FLASH	Certificate provided	Transition to "RMA"	Testability on ATE tester	Testability on eval board
	–	n	–	–	–	–	–	n	n	n
	y	y	y	y	–	–	–	y	n	SRAM tests possible
	y	y	n	–	–	–	–	n	n	No SRAM test
	y	y	–	n	–	–	–	n	n	No SRAM test
	y	y	–	–	y	y	–	n	n	FLASH tests possible
	y	y	–	–	n	–	–	n	n	No FLASH test
	y	y	–	–	–	n	–	n	n	No FLASH test
"RMA"	–	–	–	–	–	–	y	–	All tests	All tests
	–	–	–	–	–	–	n	–	Structural tests[8]	n

Note that the SRAM and flash tests on the evaluation board are tests executed on the application level. They evaluate the behavior of the related memory in terms of parameters such as ECC errors. The tests will not cover tests like Vt analysis of flash cells which can only performed in an ATE tester.

[8] Structural tests are limited test procedures executed on the ATE test machine which covers SCAN and BIST (built-in self-test) tests and give no full test coverage compared to the test capabilities which are provided if the correct OpenRMA certificate is provided and can open the device properly.

6 Preparations before sending a device for FA

There are some preparations required before a sample should go through the standard FA flow. Table 9 summarizes the steps to be performed in advance before an FA procedure is initiated. It includes the items to be provided to Infineon to guarantee a smooth process and to avoid any delay in the stages of the FA flow.

Table 9 Checklist to prepare a failing sample for FA at Infineon

#	Item	Description
1	Perform the swap test	1. De-solder the failing sample from the failing PCB. 2. Solder the failing sample on a known good PCB. 3. Prove whether the failure moves with the MCU. 4. Re-check the behavior of a known good MCU on the failing PCB. Note that if a failure is assumed in flash, you should not de-solder the MCU from the PCB. Directly contact your regional QA partner to discuss the next steps. Any heating stress may lead to destroy the failure in flash.
2	Erase sensitive data in the Work Flash	Erase related flash sectors in Work Flash which contain the sensitive data.
3	Erase sensitive data in FLASH	Erase related flash sectors in Code Flash which contain the sensitive data. Note that if authentication is activated, a defined code flash area is proved for integrity. If data is changed by the erase process, after next reset, the device enters the DEAD state. In this state, transition to “RMA” lifecycle stage is not possible. Se section 3.2 for details.
4	Provide certificate and signature	The calculation is based on the Unique ID which can be obtained from the MCU by using the ReadUniqueID API (System call 0x1F). See Appendix E.1 of AN228680 for information on generating the certificate.
5	Provide access restriction configuration	If the claimed device cannot be provided in the “RMA” lifecycle stage, contact your QA partner. A checklist will be provided to find all debug-relevant access restrictions in the used customer application.
6	Perform transition to the “RMA” lifecycle stage	See section 3.4.
7	Perform the OpenRMA system call	Perform the OpenRMA sequence to verify whether OpenRMA command is successful. See section 4.2.
8	Claim the device at QA/Infineon	1. Provide detailed MPN number of the product. 2. Provide a clear failure description and all conditions where the device failed. Avoid common descriptions like “black screen” or “no CAN communication” because this behavior of the application is a consequence of a completely different root cause. 3. Follow the guideline in section 2.1
9	Provide all analysis reports	Collect all analysis results found during the investigations done on the customer side in a report including the findings discussed with the OEM. Upload all files into MyCases system.
10	Send the device to Infineon	If no standard shipping address is defined, contact your regional QA partner at Infineon for details.

Note that it is the customer QA’s responsibility to contact the project development team to collect all required information.

7 Procedure when transition to “RMA” lifecycle stage cannot be done

There are various scenarios where the transition to “RMA” lifecycle stage can fail. For example:

- Regular software flow is not executed; external “RMA” trigger via an interface (e.g., CAN diagnosis) is not possible.
- MCU in DEAD state because of reasons such as corrupted eFuses or failure in software authentication, and chain of trust cannot be ensured.
- MCU in hard fault
- MCU performs resets continuously

In all these cases, the system call for lifecycle stage transition cannot be triggered. A detailed EFA on ATE test machine will be refused by Infineon.

To overcome this situation, perform the lifecycle stage transition via the debug interface (JTAG/SWD). According to different restrictions which are discussed in section 3.3, the customer must implement a concept into the user software to allow the access to the MCU and to trigger the system call for the lifecycle stage transition.

In this section, some possible options are introduced. These proposals must be carefully evaluated by the customer in terms of security, safety, and additional start-up timing. The customer must take care of how the concept can be implemented in their system.

Depending on the nature of the failure mechanism, the user software might be not executed in a very early stage of the regular software flow. To allow access to the MCU via the debug interface, some configuration and evaluations must be done during the startup code which enables the proper handling of the TransitionToRMA system call.

In addition, keep in mind that during HSM protection rollout[9] , additional protection configurations (with PPUs and SMPUs) might get applied, which prevents a proper handling of the system call. The cores cannot be halted during the access via SYSTEM-AP.

Note: *If this procedure fails, no transition to “RMA” lifecycle stage is possible. In this case, contact your QA partner to discuss further limited analysis steps.*

[9] HSM rollout consists of the execution of software library functions of the hardware security module.

7.1 Minimum requirements for a system call initiated by a debugger

Consider the following to enable required accesses to perform the transition to “RMA” system call.

Considering a secured system, it is recommended to temporarily disable all the access ports (AP) as part of the initial access restriction configuration. It is also important to keep in mind that enabled CM4/7 AP allows the debugger to power up the CM4/7 CPU and start code execution from an arbitrary memory address. This is possible as soon as ROM/Flash boot configures SWD/JTAG pins. In the worst-case scenario, this can happen even before the CM0+ application starts.

If the CM0+ application (e. g., hardware security module (HSM) software) has specific assets which must be protected from the main application (e. g., keys stored in the HSM portion of flash), it is recommended not to enable CPU access ports through the debug and test access restrictions. Otherwise, CM4/7 CPU could be used to access the HSM memory before CM0+ could enable protection.

Assuming that all APs are temporarily disabled and SYS_AP_MPU is not enabled as part of the access restriction configuration, the following setup must be done by the user application software which would then allow the possibility to trigger transition to “RMA” from DAP.

1. Configure all necessary DAP pins.
2. Configure the DAP MPU structures to give access to required resources (for example: IPC MMIOs, SRAM, etc.)
3. Configure a DAP MPU structure to not to give any access to the SRAM area which is additionally used by the TransitionToRMA API. This is the 2 KB of SRAM starting from (SRAM0 + 2KB).
4. PPU/SMPU settings must allow required IPC MMIO and SRAM access.
5. Interrupt initialization in the user software: Enable IRQ0 and IRQ1 to allow the handling of system call interrupts. Define interrupt vectors for both interrupts.
6. User software can then re-enable the access ports and trigger the TransitionToRMA system call.

Note: *If one of the mentioned configuration items fails during the boot process or startup code, the system call cannot be applied.*

In addition, note that because of improper initialization of the Crypto memory buffer and internal SRAM0; Crypto and SRAM0 ECC errors may be set after the TransitionToRMA call. Also, the PERI_GROUP_VIO_2 fault may get set. To avoid this issue, do not configure the fault structure for Crypto and SRAM0 ECC errors and PERI_GROUP_VIO_2 fault before triggering the TransitionToRMA system call or ignore the ECC faults reported during the TransitionToRMA system call execution.

Note: *The device in the “RMA” lifecycle stage requires OpenRMA API on every reset. If the device-specific failure, such as hardware failure, triggers a reset by the fault report after OpenRMA execution, the device cannot be opened from the “RMA” stage. There are two options to handle this case:*

1. *Mask the device-specific failure that triggers a reset in advance before triggering the TransitiontoRMA API. The application software that runs after the OpenRMA system call also needs to be masked. The protection state of the “RMA” lifecycle stage indicates VIRGIN. Therefore, the software can know if the device is in the “RMA” lifecycle stage using the CPUSS_PROTECTION register.*
2. *Reprogram a code that performs only “RMA” management before triggering the TransitiontoRMA API.*

7.2 Debugger detection method

After the boot process has successfully performed, the entry address of the customer-specific code is called. Considering a malfunction of the MCU in a very early stage of software code execution, all configurations and preparations for a possible lifecycle transition must be done in the startup code.

To identify if a lifecycle transition should be initiated, an external event is required to enter a TransitionToRMA mode which is a customer-specific mode in the application software to handle the transition to “RMA” lifecycle stage.

The main difference of the hardware setup is that during the regular execution of the customer code, no debugger is connected to the JTAG/SWD interface. This means that if the TransitionToRMA mode should be entered, a connected debugger can be used as an external event.

The status of a proper connection of the debugger to the DAP can be proved by checking of the SWJ_CONNECTED bit of the CPUSS_DP_STATUS register.

If a debugger is identified in the startup code, the software can enter the TransitionToRMA mode. Note that the debug port pins need to be defined during boot. See section 3.3.2.2 for details.

In that mode, the user application software remains in a “while(1)”-loop after IRQ0 and IRQ1 are enabled; the transition to “RMA” lifecycle stage can be done via DAP.

To make the system more robust, a second requirement can be implemented where a dedicated pattern need to be written into a dedicated SRAM area.

If both criteria are fulfilled, a transition to “RMA” lifecycle stage can be initiated.

7.3 Port pin detection method

A similar procedure as described in section 7.2 with the difference that one or more port pins will be used as an external event.

Depending on the pattern or a dedicated frequency on the used port pins, the TransitionToRMA mode can be entered and the transition to “RMA” lifecycle stage can be initiated.

7.4 Debug script for transition to “RMA”

Basically, the procedure for TransitionToRMA system call is very similar to the OpenRMA command explained in section 4.2.

The main differences are:

- COMMAND ID: 0x120028F0
- OP-code: 0x28000000

Thus, certificate and signature will be different.

8 Glossary

Table 10 Definitions, acronyms, and abbreviations

Acronyms and abbreviations	Definition
8D-report	Documentation of analysis results for a claimed device
AP	Access Port
ATE	Automatic Test Equipment – Test Machine
CA	Correction Action
CAR	Correction Action Request
CCAR	Customer Corrective Action Request
CPU	Central Processing Unit
CSR	Customer Specific Requirements
CQE	Customer Quality Engineering
CSAM	Confocal Scanning Acoustic Microscopy
DAR	Dead Access Restrictions
DAP	Debug Access Port
EFA	Electrical Failure Analysis
EOL	End of line, production test
FA	Failure Analysis
HSM	Hardware Security Module
IP	Intellectual Property / Peripheral Module in the context of this AN
MCU	Microcontroller
MPU	Memory Protection Unit
NAR	Normal Access Restrictions
NOK	Not ok, failing device
NTF	No trouble found
PCB	Printed circuit board
QA	Quality assurance
SAR	Secure Access Restrictions
TOC2	Table Of Contents 2. This is an area in SFlash that is used to store pointers to two applications blocks: Secure Image and Main User Application. It also contains some boot parameters that are settable by the system designer.
TRM	Technical reference manual

References

The following are the TRAVEO™ T2G family series datasheets and technical reference manuals. Contact Technical support to obtain these documents.

[1] Device datasheet

- CYT2B7 datasheet 32-bit Arm® Cortex®-M4F microcontroller TRAVEO™ T2G family
- CYT2B9 datasheet 32-bit Arm® Cortex®-M4F microcontroller TRAVEO™ T2G family
- CYT4BF datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family
- CYT4DN datasheet 32-bit Arm® Cortex®-M7 microcontroller TRAVEO™ T2G family

[2] Technical reference manual

- CYT2B series
  - TRAVEO™ T2G automotive body controller entry family architecture technical reference manual (TRM)
  - TRAVEO™ T2G automotive body controller entry registers technical reference manual (TRM) for CYT2B7
  - TRAVEO™ T2G automotive body controller entry registers technical reference manual (TRM) for CYT2B9
- CYT4B series
  - TRAVEO™ T2G automotive body controller high family architecture technical reference manual (TRM)
  - TRAVEO™ T2G automotive body controller high registers technical reference manual (TRM)
- CYT4D series
  - TRAVEO™ T2G automotive cluster 2D family architecture technical reference manual (TRM)
  - TRAVEO™ T2G automotive cluster 2D registers technical reference manual (TRM)

[3] Application note

- AN228680 – Secure system configuration in TRAVEO™ T2G family

Customer request handout

This document can be used to send in a customer request for support. It provides all necessary information to ensure a qualified and fast answer to the request

Table 11 Customer request handout

Needed information	Additional information
Contact person	–
End customer	–
Mark code and lot number	Lot code, date code information printed on the MCU. Provide a photograph if possible.
Project name	Project name of the final product
Failure rate and conditions	How likely the failure is reproduced and under which conditions (temperature, supply voltage, CAN messages, etc.)
Quantity of return samples	–
Failure found at • Incoming • Programmer (provide third-party vendor name) • On-board programming • Board level – ICT • Qualification test • System level – FT • 0km Field (mileage)	–
CCAR priority • Catastrophic • High • Standard	Provide the business justification for this priority classification. For example: details of the customer's lines down situation and/or how this has potential to severely affect Infineon's revenue.
Is there any special laser marking or X-ray inspection step in the production? If yes, provide details.	–
Did the customer SMT reflow profile comply with JEDEC standard?	Provide reflow profile if necessary.
Did the customer update or change their system design or manufacture process flow recently?	If yes, provide details.
Did the customer cross-check the failed units on good boards with exchange test?	If yes, provide details.
MPN number	Detailed part number

Customer request handout

Needed information	Additional information
Detailed failure description • when the problem was detected • provide the failure category (read/programming/erase or booting/reset fail) • Operating conditions (VCC, temperature, test duration) • Memory map in the customer application • Provide details of the flash: failing address, sector information, and expect data vs actual data or data dump • Is the return unit in the original failure state or did it go through the replication test in customer lab? • Can the flash be erased/ re-programmed for further analysis? • If failure is expected in the flash, do not desolder the sample from the PCB to avoid thermal stress which might have an impact on the failure	–
If the device is transitioned to the “RMA” lifecycle stage, provide the certificate to open the device using the OpenRMA system call.	This certificate is unique for each device in the “RMA” lifecycle stage. See section 6 for additional information.

Revision history

Document revision	Date	Description of changes
**	2021-07-02	Initial release
*A	2023-03-07	Updated Table 8 and content in 7.1 section

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

Edition 2023-03-07
Published by

Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
002-30102 Rev. *A

Important notice
The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.