# X-CUBE-CLASSB

Errata sheet

# X-CUBE-CLASSB self test library software errata

## Introduction

This document applies to the X-CUBE-CLASSB STM32Cube Expansion Package. The X-CUBE-CLASSB references concerned by the errata in this document are summarized in Table 1 together with the related Arm® processors.

This errata sheet is dedicated to firmware versions v4.0.0 and higher. However, if specifically indicated, some proposed workarounds can also be applied to older versions and adapted to their source code. To address known issues in these older packages, users should primarily implement the measures described in section 3.10 of application note *Guidelines for obtaining UL/CSA/IEC 60730-1/60335-1 Class B certification in any STM32 application* (AN4435).

**Table 1. Impacted variant summary**

| X-CUBE-CLASSB reference(1) | Arm® processor |
|---|---|
| X-CUBE-CLASSB-G4 | Cortex®-M4 |
| X-CUBE-CLASSB-L4 | |
| X-CUBE-CLASSB-MP1 (Cortex®-M4) | |
| X-CUBE-CLASSB-WL | |
| X-CUBE-CLASSB-F7 | Cortex®-M7 |
| X-CUBE-CLASSB-H7 | |
| X-CUBE-CLASSB-L5 | Cortex®-M33 |
| X-CUBE-CLASSB-U5 | |

1. *Refer to each erratum for the references and versions considered.*

# 1 General information

The X-CUBE-CLASSB STM32Cube Expansion Package runs on STM32 microcontrollers and microprocessors based on the Arm® Cortex® processor.

Table 2 defines the acronyms needed for a better understanding of this document.

**Table 2. List of acronyms**

| Acronym | Description |
|---|---|
| API | Application programming interface |
| DC | Diagnostic coverage |
| DCache | Arm® Cortex®-M7 data cache |
| STL | Self-test library |
| TM | Test module |

*Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.*

# 2 Summary of errata

Table 3 gives a reference to the software errata and their status:

- A = workaround available
- N = no workaround available
- P = partial workaround available
  An erratum workaround is deemed partial if
  - either it only reduces the rate of occurrence, or the consequences, or both
  - or if it is effective only for a subset of applicable references or versions
  - or it is a combination of both

**Table 3. Summary of software errata**

| Domain | Section | Erratum | Status |
|---|---|---|---|
| CPU | 3.1 | The CPU TMCB function might return a wrong test failure status | A |
| RAM | 3.2 | Risk of data cache and RAM misalignment during the RAM TM execution when the data cache is enabled | A |

# 3 Description of software errata

## 3.1 The CPU TMCB function might return a wrong test failure status

### 3.1.1 Applicability

**Table 4. Applicability of the CPU TMCB erratum**

| X-CUBE-CLASSB reference | X-CUBE-CLASSB versions | Arm® processor |
|---|---|---|
| X-CUBE-CLASSB-G4 | Up to v4.0.0 included | Cortex®-M4(1) |
| X-CUBE-CLASSB-L4 | Up to v4.0.0 included | |
| X-CUBE-CLASSB-MP1 (Cortex®-M4) | Up to v4.0.0 included | |
| X-CUBE-CLASSB-WL | Up to v4.0.0 included(2) | |
| X-CUBE-CLASSB-F7 | Up to v4.0.0 included | Cortex®-M7(1) |
| X-CUBE-CLASSB-H7 | Up to v4.0.0 included | |
| X-CUBE-CLASSB-L5 | Up to v4.0.0 included | Cortex®-M33(1) |
| X-CUBE-CLASSB-U5 | Up to v4.0.0 included | |

1. *For library versions older than v4.0.0 provided in open-source format for Cortex®-M4, Cortex®-M7, and Cortex®-M33 core-based products, users must implement the appropriate workaround at the beginning of the STL_StartUpCPUTest procedure. This is necessary because the APSR register content is incorrectly assumed to have all GE bits cleared during the test.*
2. *There is no such issue in any X-CUBE-CLASSB release for the STM32 microcontrollers based on the Cortex®-M0 and Cortex®-M0+ processors.*

### 3.1.2 Erratum details

#### *3.1.2.1 Description*

The CPU TMCB function might return the STL_FAILED value even when the test is executed correctly on a nonfaulty hardware. In such a case, the test result is a false positive.

Inside the CPU TMCB, some internal computations depend on the GE bits from the xPSR register:

- If one of the GE bits, GE[0:3] (APSR[19:16]), is set to 1 before CPU TMCB execution (or set by an interrupt handler during CPU TMCB execution), the test result is set to STL_FAILED, false positive.
- If the GE bits are set to 0, the test result is set to STL_PASSED. No abnormal behavior is noticed.

*Note: The above CPU TMCB behavior is deterministic.*

#### *3.1.2.2 Workaround*

Apply in the application software one of the following measures just before calling the CPU TMCB function. The choice between the two measures depends on the application software, whether it relies on GE bits values or not.

- Either simply reset the GE bits from xPSR
- Or save the xPSR, reset the GE bits from xPSR, and restore the xPSR value after the test execution

#### *3.1.2.3 Environment issue description*

The issue happens only when the two following conditions are met:

- At least one instruction from SADD8, SADD16, SSUB8, SSUB16, SSAX, SASX, UADD8, UADD16, USUB8, USUB16, USAX, and UASX is used inside the application software (outside STL own functions)
- The result of these instructions sets to 1 one or multiple GE bits, just before the CPU TMCB function call

If these instructions are not directly programmed by using corresponding intrinsic functions (such as __UADD8() or others) or in assembler (inline or in assembler code), their usage might also depend on the compiler and potentially on the compiler version. Note that the instructions can possibly be used in specific libraries included in the final project.

## 3.2 Risk of data cache and RAM misalignment during the RAM TM execution when the data cache is enabled

### 3.2.1 Applicability

**Table 5. Applicability of the RAM TM erratum**

| X-CUBE-CLASSB reference | X-CUBE-CLASSB versions | Arm® processor |
|---|---|---|
| X-CUBE-CLASSB-F7 | Up to v4.0.0 included | Cortex®-M7 |
| X-CUBE-CLASSB-H7 | Up to v4.0.0 included | |

*Note: The RAM test module erratum applies to the Cortex®-M7 only, when the data cache is enabled by the application.*

### 3.2.2 Erratum details

#### *3.2.2.1 Description*

The issue described below only happens if the data cache (DCache) is enabled on the product. If the DCache is not enabled, there is no issue and the workaround is not needed.

The DCache is disabled and reenabled during RAM test execution by specific STL internal utilities to ensure that the RAM test is really executed on the RAM and not on the DCache. This is explained in the corresponding user guide in *Chapter 4.3.10*. Older versions of the library designed for Cortex®-M7 based products do not include the DCache disabling protection feature. Therefore, users must ensure proper handling of DCache enabling once the RAM test is executed.

If an interrupt handler that uses cacheable data appears at a specific timing during the execution of these DCache disabling and enabling utilities, a RAM/DCache misalignment might occur, and the interrupt handler might use incorrect cacheable data (such as variables, stack, and others).

X-CUBE-CLASSB provides no notification to detect if the issue did happen.

#### *3.2.2.2 Workaround*

The main principle of the workaround is to prevent any interrupt handling when executing the disable and reenable DCache flow around the RAM test execution. To do so, the application must follow the mandatory steps below while performing the RAM test:

- Disable the DCache before running the RAM test module
- Enable the DCache just after returning from the RAM TM execution
- Encapsulate both DCache disabling and enabling function calls into a disabled interrupt window

Workaround proposal with the use of CMSIS procedures calls (to be adapted to the user's application):

```
STL_UTIL_Disable_IT(); /* be sure that flag STL_ENABLE_IT is not activated to really disable/
enable interrupt with the use of STL functions, or use   __disable_irq(); or…*/
SCB_DisableDCache();
STL_UTIL_Enable_IT();

yyyyy = STL_SCH_RunRamTM(xxxxxx);

STL_UTIL_Disable_IT();
SCB_EnableDCache();
STL_UTIL_Enable_IT();
```

*Important: If STL dedicated functions for disabling and enabling interrupts are used (STL_UTIL_Disable_IT() and STL_UTIL_Enable_IT()), be sure that the flag STL_ENABLE_IT is not activated. If the flag STL_ENABLE_IT is activated, functions like intrinsic __disable_irq() must be used instead.*

*Note: The intrinsic functions SCB_DisableDCache and SCB_EnableDCache are executed in loops that service each data cache set and way individually. Therefore, their execution depends on the product through its data cache size. Refer to the definitions of these functions in the file `core_cm7.h`.*

**STL performance and execution impact**

The use of the workaround has no impact on the execution timing of the STL_SCH_RunRamTM() function. Without the workaround, the main part of the function is already executed with DCache disabled. With the workaround, only a few additional instructions are executed with DCache disabled.

The use of the workaround implies that the execution of the interrupt handler happening during the run of RAM TM is executed with the DCache disabled.

#### *3.2.2.3 Environment issue description*

The issue might occur only for the X-CUBE-CLASSB running on the Cortex®-M7 during the RAM test when DCache is activated. Because this issue is related to a critical run with intervening interrupts, it can be difficult to reproduce it in a deterministic way.

# Revision history

Table 6. **Document revision history**

| Date | Revision | Changes |
|---|---|---|
| 21-Nov-2024 | 1 | Initial release. |

# Contents

# List of tables

**IMPORTANT NOTICE – READ CAREFULLY**

STMicroelectronics International NV and its affiliates (“ST”) reserve the right to make changes corrections, enhancements, modifications, and improvements to ST products and/or to this document any time without notice.

This document is provided solely for the purpose of obtaining general information relating to an ST product. Accordingly, you hereby agree to make use of this document solely for the purpose of obtaining general information relating to the ST product. You further acknowledge and agree that this document may not be used in or in connection with any legal or administrative proceeding in any court, arbitration, agency, commission or other tribunal or in connection with any action, cause of action, litigation, claim, allegation, demand or dispute of any kind. You further acknowledge and agree that this document shall not be construed as an admission, acknowledgment or evidence of any kind, including, without limitation, as to the liability, fault or responsibility whatsoever of ST or any of its affiliates, or as to the accuracy or validity of the information contained herein, or concerning any alleged product issue, failure, or defect. ST does not promise that this document is accurate or error free and specifically disclaims all warranties, express or implied, as to the accuracy of the information contained herein. Accordingly, you agree that in no event will ST or its affiliates be liable to you for any direct, indirect, consequential, exemplary, incidental, punitive, or other damages, including lost profits, arising from or relating to your reliance upon or use of this document.

Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment, including, without limitation, the warranty provisions thereunder.

In that respect, note that ST products are not designed for use in some specific applications or environments described in above mentioned terms and conditions.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers’ products.

Information furnished is believed to be accurate and reliable. However, ST assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.