RTA-OS Zynq/RVDS

Release Note - Version 5.0.8 (12-05-2016)

Copyright

Document: 10558-RN-5.0.8 EN-05-2016(12-05-2016)

Safety Notice

This ETAS product fulfills standard quality management requirements. If requirements of specific safety standards (e.g. IEC 61508, ISO 26262) need to be fulfilled, these requirements must be explicitly defined and ordered by the customer. Before use of the product, customer must verify the compliance with specific safety standards.

Contents

1 Introduction

RTA-OS is an AUTOSAR compliant Operating System and associated tooling. This document provides release information for the RTA-OS Zynq/RVDS port plug-in that customizes the RTA-OS development tools for the Xilinx Zynq-7020 with the ARM DS-5 v5 compiler. It supplements the more general information you can find in the *Release Note*.

1.1 Version Information

This is Version 5.0.8 of the RTA-OS Zynq/RVDS plug-in.

1.2 Installation

The installation process is covered in detail in the *ZynqRVDS Port Guide*.

2 Open EHI Calls

Open issues are referred to by their call number in the ETAS Helpdesk International (EHI) system.

No EHI calls are open.

3 Change History

3.1 Version 5.0.8

Additional Features

The following features have been added to this release:

- Seventh fully featured and tested release.

Modified Features

The following features have been modified in this release:

- Updated to add fix for EHI issues 480164, 495725 and 51186.
- Cross-core interrupt routines have been updated to remove an unnecessary (but safe) interrupt priority manipulation.
- The default implementation of Os_Cbk_GetAbortStack() used in for multicore applications that do not include untrusted objects (i.e. Tasks ISRs and functions) has been updated to support each core with a separate abort stack.
- Volatile modifiers have been added to pointers in Os_InitializeVectorTable() to support higher levels of compiler optimization.
- The capitalization of the target option 'Read CoreID from ICDIPTR0' has been updated to match the other target options.

Removed Features

No features have been removed from this release.

3.2 Version 5.0.7

Additional Features

The following features have been added to this release:

- Sixth fully featured and tested release.
- Os_IntChannel_x macros.
- Target option to make Os_Get_CoreID() detect the core ID by reading from the memory mapped ICDIPTR0 register in the Generic Interrupt Controller.

Modified Features

The following features have been modified in this release:

- The code to detect the core ID in multicore applications has been updated. The OS now reads from the MPIDR register in CP15 by default. As this can only be read by trusted code a mode swap is required if the core ID needs to be read in untrusted code. A target option allows untrusted code to avoid this mode swap and read from ICDIPTR0 in the GIC instead.

Removed Features

No features have been removed from this release.

3.3 Version 5.0.6

Additional Features

The following features have been added to this release:

- Fifth fully featured and tested release.
- Target option to allow the default interrupt to be run at a low IPL.
- Target option to always use Os_Cbk_GetAbortStack() in the abort ISR rather than just in SC3/SC4 applications.

Modified Features

The following features have been modified in this release:

- The code to support the 'enable stack repositioning' target option has been updated. The assembly language instructions generated now do not rely on values stored in the CPU general purpose registers to be preserved over the call to untrusted code.
- The support for stack and execution time measurement has been updated to prevent a possible miscalculation when a higher priority Category 2 interrupt occurs during the calculations.
- The code to detect the core ID in multicore applications has been updated. If the application contains untrusted code the OS no longer uses the MPIDR register to determine the core ID as this can only be read in trusted code and a mode swap is required if the core ID needs to be read in untrusted code. Now the ICDIPTR0 register is used when there is untrusted code in the application as this register can be be read in both untrusted and trusted code. In multicore applications ICDIPTR0 read access must not be restricted by memory protection.

8 Change History

- Updated the default implementation of Os_Cbk_GetAbortStack() so that no stack is used in both single and multicore applications.

Removed Features

No features have been removed from this release.

3.4 Version 5.0.5

Additional Features

The following features have been added to this release:

- Fourth fully featured and tested release.

Modified Features

The following features have been modified in this release:

- Updated to add fix for EHI issue 436568 and 436570.

Removed Features

No features have been removed from this release.

3.5 Version 5.0.4 (Preview Release)

Additional Features

The following features have been added to this release:

- Interim Preview Release.

Modified Features

The following features have been modified in this release:

- Updated to add fix for EHI issue 427311.

Removed Features

No features have been removed from this release.

3.6 Version 5.0.3 (Preview Release)

Additional Features

The following features have been added to this release:

- Interim Preview Release.

Modified Features

The following features have been modified in this release:

- Update to ’Enable untrusted stack check’ support.

Removed Features

No features have been removed from this release.

3.7 Version 5.0.2

Additional Features

The following features have been added to this release:

- Third fully featured and tested release.
- Calls to the __schedule_barrier() intrinsic function have been added to code sequence points to prevent operations with side effects from being re-ordered by the compiler.

Modified Features

The following features have been modified in this release:

- Supports the v5.04 ARM DS-5 compiler update 2 tools.
- The optimizer setting to use -Otime following the update to the compiler.
- The compiler options –autoinline and –forceinline no longer used.

Removed Features

No features have been removed from this release.

3.8 Version 5.0.1

Additional Features

The following features have been added to this release:

- Second fully featured and tested release.
- Support for the Zynq-7015 and Zynq-7100 variants
- A target option to select the -fpmode compiler option as either none or fast

10 Change History

Modified Features

The following features have been modified in this release:

- Supports the v5.04 ARM DS-5 compiler tools.
- The compiler version test.
- The optimizer setting to use -Ospace rather than -Otime due to issues with the generated code.

Removed Features

No features have been removed from this release.

3.9 Version 5.0.0

Additional Features

The following features have been added to this release:

- First fully featured and tested release.
- Interrupt configuration macros added (i.e. Os_Set_Edge_Triggered_x(), Os_Set_Level_Sensitive_x()).
- Support for user selection of Multicore Cross-core interrupts

Modified Features

The following features have been modified in this release:

- Multicore support now fully tested.

Removed Features

No features have been removed from this release.

3.10 Version 1.99.4 (Preview Release)

Additional Features

The following features have been added to this release:

- Fifth Early Access Release.

Modified Features

The following features have been modified in this release:

- Spinlock release code now modified according to the Cortex A/R ARM Architecture Reference Manual (ARM DDI 0406C).
- Supports the v5.02 ARM DS-5 compiler tools.
- Zynq-7045 replaces Zynq-7040 variant.
- Multicore libraries modified to pass lint/MISRA tests.
- Testsuite updated to support the v5.1.1 RTA-OS tools.

Removed Features

No features have been removed from this release.

3.11 Version 1.99.3 (Preview Release)

Additional Features

The following features have been added to this release:

- Fourth Early Access Release.

Modified Features

The following features have been modified in this release:

- Fix support for spurious interrupt.

Removed Features

No features have been removed from this release.

3.12 Version 1.99.2 (Preview Release)

Additional Features

The following features have been added to this release:

- Third Early Access Release.
- Multi Core support.

12 Change History

Modified Features

The following features have been modified in this release:

- FIQ interrupt support removed.

Removed Features

No features have been removed from this release.

3.13 Version 1.99.1 (Preview Release)

Additional Features

The following features have been added to this release:

- Second Early Access Release.
- ECC task support.
- Timing protection.
- Memory protection.
- Support for aligning stack to MPU regions in tasks and ISRs.
- Stack corruption testing in ISRs.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

3.14 Version 1.99.0 (Preview Release)

Additional Features

The following features have been added to this release:

- Initial Early Access.
- Category 1 and 2 interrupts (IRQ only).
- Category 1 GIC spurious interrupt support.
- BCC task support.
- SC1 Autosar conformance.
- RTA-Trace support.

Modified Features

No features have been modified in this release.

Removed Features

No features have been removed from this release.

4 Fixed EHI Calls

Bugs that have been fixed are referred to by their call number in the ETAS Helpdesk International (EHI) system.

4.1 Version 5.0.8

EHI 480164

Status: Fixed

Title: Backward compatibility when using RTA-OS Tools version 5.4.2

Description: When RTA-OS Tools v5.4.2 is used a compilation error is seen in Os_Wrapper.c stating that variable isr_needs_termination_callback is undefined. RTA-OS Tools v5.4.3 creates this variable but RTA-OS Tools v5.4.2 does not and thus it cannot be referred to by the port. This release has updated Os_Wrapper.c to fix this issue.

EHI 495725

Status: Fixed

Title: Default RTA-OS Os_Cbk_GetAbortStack() code can return NULL

Description: In configurations that use the Os_Cbk_SetMemoryAccess callback to update the memory protection settings for untrusted code, but where the stack value is not actually passed to the callback (i.e. Stack Monitoring is disabled AND target option 'Enable stack repositioning' is false) a NULL value can be returned. This release has updated the default implementation of Os_Cbk_GetAbortStack() to fix this issue.

EHI 511186

Status: Fixed

Title: Cross-core interrupts running at the highest Category 2 ISR priority

Description: When an RTA-OS configuration has either stack measurement or timing execution measurement enabled then the cross-core ISRs are erroneously given the highest Category 2 ISR priority. This release has updated the interrupt initialization values applied in Os_InitializeVectorTable() to fix this issue.

4.2 Version 5.0.5

EHI 436568

Status: Fixed

Title: Build error in SC3 systems: CallTrustedFunction.c Os_untrusted_function_alignment_handler

Description: In multicore SC3 applications local variables are not inserted in the Os_untrusted_function_alignment_handler function when ORTI is disabled. The RTA-OS library fails to build. This release has updated Os_untrusted_function_alignment_handler() to fix this issue.

EHI 436570

Status: Fixed

Title: Build error GetTaskActivationTime.c

Description: In applications with time monitoring enabled a local variable is declared but not used in GetTaskActivationTime.c. As all warnings are treated as errors this prevents the library from building. This release has updated GetTaskActivationTime.c to fix this issue.

4.3 Version 5.0.4 (Preview Release)

EHI 427311

Status: Fixed

Title: Register overwritten in Os_SVCHandler()

Description: When the 'Enable untrusted stack check' target option is enabled the test to determine if interrupted untrusted code has an illegal stack pointer caused a register to be corrupted. This release has updated the Os_SVCHandler() to fix this issue.

4.4 Version 5.0.2

EHI 413764

Status: Fixed

Title: Missing declarations in Os.h/Os_Cfg.h

Description: Os_Cfg.h updated to contain declarations for Os_trigger_SVC and Os_CPSR_val.

4.5 Version 5.0.1

EHI 358338

Status:	Fixed
Title:	Potential optimization of API calls
Description:	In SC3 configurations the end of all API calls has been optimized to remove unnecessary trust mode changes.

5 Limitations

5.1 Installer

There are the following limitations for the installer:

Limitation	None.
Workaround	None.

5.2 ORTI Support

There are the following limitations for this tool:

Limitation	Versions of RTA-OS tools earlier than v5.4.3 may generate ORTI files with symbols that have no debug information. This prevents Trace32 from displaying some ORTI fields.
Workaround	None.

5.3 ZynqRVDS DLL

There are the following limitations for this tool:

Limitation	None.
Workaround	None.

6 Contacting ETAS

6.1 Technical Support

Technical support is available to all users with a valid support contract. If you do not have a valid support contract, please contact your regional sales office (see Section 6.2.2).

The best way to get technical support is by email. Any problems or questions about the use of the product should be sent to:

rta.hotline.uk@etas.com

If you prefer to discuss your problem with the technical support team, you call the support hotline on:

+44 (0)1904 562624.

The hotline is available during normal office hours (0900-1730 GMT/BST).

In either case, it is helpful if you can provide technical support with the following information:

- Your support contract number
- Your `.xml`, `.arxml`, `.rtaos` and/or `.stc` files
- The command line which caused the error
- The version of the ETAS tools you are using
- The version of the compiler tool chain you are using
- The error message you received (if any)
- The file `Diagnostic.dmp` if it was generated

6.2 General Enquiries

6.2.1 ETAS Global Headquarters

ETAS GmbH

Borsigstrasse 14	Phone:	+49 711 3423-0
70469 Stuttgart	Fax:	+49 711 3423-2106
Germany	WWW:	www.etas.com

6.2.2 ETAS Local Sales & Support Offices

Contact details for your local sales office and local technical support team (where available) can be found on the ETAS web site:

ETAS subsidiaries	www.etas.com/en/contact.php
ETAS technical support	www.etas.com/en/hotlines.php