

TASKING™

VX-toolset for ARM Cortex-M and R



The TASKING VX-toolset for ARM Cortex-M/R consists of:

- ISO C++ compiler, scalable to EC++
- C compiler, ISO C'99 compliant, with integrated MISRA C and CERT C static code analyzer
- Assembler/Linker/Locator
- C/C++ libraries, run-time libraries, floating point libraries
- Integrated Development Environment (IDE) based on Eclipse™
- Debugger integrated into Eclipse, with two execution environments:
 - Simulator
 - On-Chip (OCDS) debugger, using JTAG
- Support for Cortex-M and Cortex-R



Support for STMicroelectronics

Altium and STMicroelectronics have been partners since the first ST10 microcontrollers became available. The TASKING development tools for the ST10 are used extensively in many automotive and hard disk drive applications.

This relationship has culminated in a strong focus from Altium to support STMicroelectronics' range of STM32 microcontrollers and its ST-Link debug probe with the TASKING VX-toolset for ARM Cortex-M/R.

Users of the VX-toolset for ST10 will find an excellent migration path in the VX-toolset for ARM Cortex-M/R to the world of 32-bit performance.

The TASKING ARM toolset

Altium's embedded software development tools for the ARM Cortex processors offers industry leading code performance and quality at the accessible price levels that Altium is famous for.

Based on Altium's sophisticated Viper compiler technology, this suite of tools is released as the TASKING VX-toolset for ARM Cortex-M/R. The VX-toolset takes a major step forward by offering unparalleled code optimization performance, a totally new debugger, which is ready for the latest industry trends such as multi-core application development, plus toolset integration into the popular Eclipse™ platform as an Integrated Development Environment (IDE).

Eclipse integrates the compiler, assembler and linker seamlessly into a single IDE and comes with wizard functionality to set up the developer's application and configuration for target debugging. The new debugger is integrated into Eclipse through a plug-in. In this way the user has all development tools available within one state-of-the-art and industry-standard IDE, with the option to plug-in additional modules from third parties.

Altium's TASKING VX-toolset is available in targeted bundles – Standard, Professional and Premium Edition – allowing you to choose the best fit for your application development activities. Along with the standard Eclipse, C compiler toolset and simulator modules, additional functionality include options such as a C++ compiler, CERT C secure code analysis, on-hardware debugging through an on-chip debug solution, USB-to-JTAG debug probe plus flash memory programming support.

Eclipse IDE

The Integrated Development Environment (IDE) that is built on the Eclipse framework provides a seamless workbench for the complete tool chain including the debugger of the VX-toolset. The IDE provides facilities for project configuration and management, C/C++ and assembly code aware editing, build management, debugging, profiling and more. It provides wizards to help you to set up your embedded ARM project and to configure your target board settings to debug your project on hardware.

The Eclipse editor supports C, C++, assembly language and header files with syntax highlighting, auto completion, context assistance and tool tips. As you would expect from a de facto standard IDE, it provides full support for all relevant source code version control systems. The Eclipse environment provides you with a single platform for many different embedded product toolsets from different vendors. The standardization on an industry-wide IDE significantly reduces your learning curve, removes the barriers of changing development tools for different architectures, increases your productivity and ultimately reduces the time to market with your end product. The availability of plug-in modules to enhance or extend the feature set of the Eclipse IDE ensures that you can build the workbench according to your development needs. With the concept of the open Eclipse framework, third-party tool

vendors can now develop plug-ins that tightly integrate into various IDEs from different vendors, contrary to proprietary IDEs for which custom connectivity needs to be created.

The VX-toolset's IDE is based on the v3.5 release of Eclipse and the C/C++ Development Tools (CDT) plug-in version 6.0. Altium has built the integration blocks for the toolset and extensions to Eclipse to make the whole environment a coherent workbench.

C compiler

Based upon Altium's latest DSP-C compiler technologies, the VX-toolset C compiler is reliable, compliant, competitive, complete, easy to use and generates the most optimal code possible to allow you to take full advantage of the ARM architecture. The TASKING VX-compiler for ARM is tested for ISO C'99 and ISO C++ conformity against authoritative validation suites, such as Perennial and Plum Hall. In addition, the optimization techniques of the compilers are tested with various large real-world applications (for example, audio/GSM codec suites), as well as industry benchmark standards such as Nullstone and EEMBC.

Fast and compact

Altium understands that you expect your ARM compiler to produce the most optimal code possible with no fuss. With its Viper compiler technology, the TASKING VX-toolset for ARM Cortex-M/R, in its default configuration, generates code with the smallest footprint and fastest execution possible. Depending on the specific requirements of

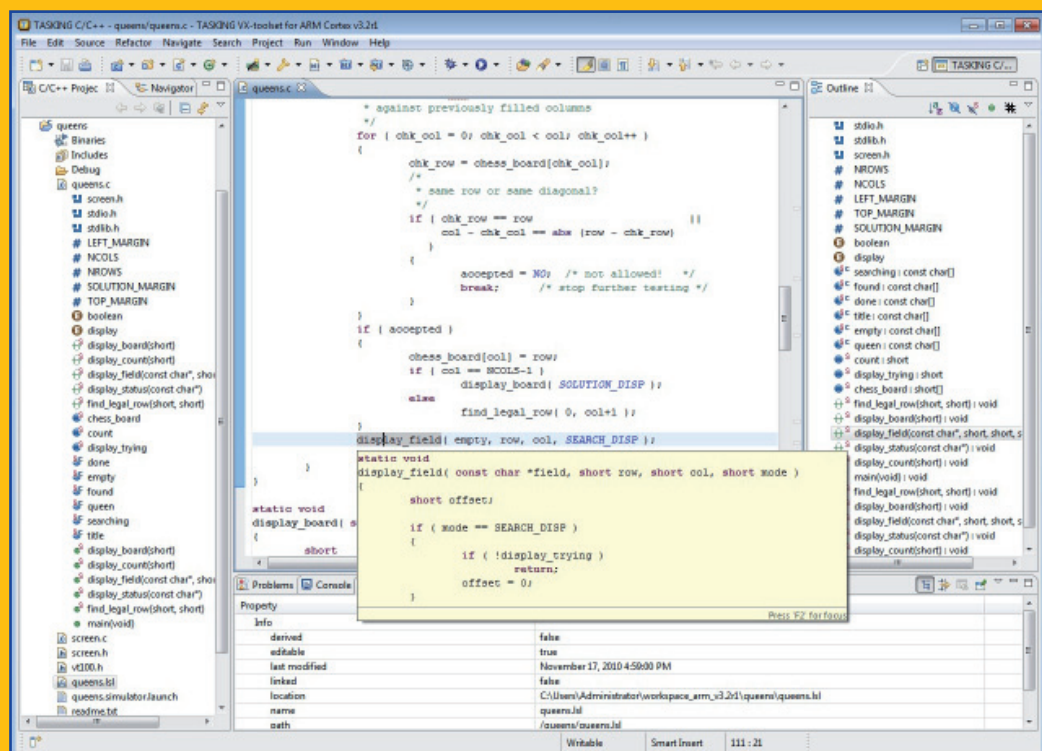
your ARM application, optimizations can then be further tweaked for smaller code size or higher execution speed.

Compiler optimizations include:

- Partial Redundancy Elimination (PRE) detects and eliminates repeating (sub-) expressions
- Various Loop and Jump optimizations speed up execution and reduce code size
- Control-flow and code-reduction optimizations remove dead code and perform transformations to minimize jumps
- Function inlining replaces calls to small functions with inlined copies of the function code
- Peephole optimizations replace instruction sequences with equivalent but faster and/or shorter sequences, or remove obsolete instructions
- Inter-procedural register allocation

Code profiling

In addition to the profiling features built into the debugger, the compiler is equipped with a profiler that uses code instrumentation. Code profiling can be used to determine which pieces of your code execute slower than expected and which functions contribute most to the overall execution time of a program. A profile can also tell you which functions are called more or less often than expected. The advantage of this code profiling option in the compiler is that it can give a complete call graph of the application annotated with the time spent in each function and basic block.



Eclipse Integrated
Development Environment
for editing and project
management.

Target architecture support

Altium's TASKING VX-toolset supports Cortex-M0, Cortex-M3 and Cortex-M4 based microcontrollers, as well as the Cortex-M1 based softcore for FPGA based applications. From within the Eclipse IDE you can easily select the device of your choice for your project, such as:

NXP:	STMicroelectronics:
LPC11xx (M0)	STM32F100
LPC13xx (M3)	STM32F101
LPC17xx (M3)	STM32F102
	STM32F103
Texas Instruments	STM32F105
(Luminary)	STM32F107
Stellaris family:	STM32L151
LM3Sx00	STM32L152
LM3S1000	STM32F205
LM3S2000	STM32F207
LM3S3000	STM32F215
LM3S5000	STM32F217
LM3S6000	
LM3S8000	
LM3S9000	

Support for new variants and new semiconductor vendors is constantly being added. Support for Cortex-R will be available in H1 2011.



Industry standards support and conformance

The various parts of the VX-toolset conform to or provide support for relevant applicable standards, such as ISO/IEC 9899:1999(E), ANSI-C X3.159-1989, ISO/IEC 14882:1998(E), MISRA-C:2004, MISRA-C:1998, IEEE-754, ELF/DWARF 3, S-Record, CERT C, ARM ABI, CMSIS and ORTI.

Several forms of profiling output can be obtained:

- Flat profile shows how much time is spent in each function, how many times that function has been called, and optionally how often each lexical block within the function is executed. This is very useful if you want to know which functions or lexical blocks consume most cycles
- Call graph profile shows, for each function, which functions called it, which other functions it called, and how many times. There is also an estimate of how much time was spent in the subroutines of each function

Syntax and semantic checks

The compiler offers a vast array of syntax and semantic checks that warn about potential undesirable effects or bugs in your program. Early fixing of source code problems when reported by the compiler generally only takes minutes compared to hours, or days, when the problem is discovered at run time.

Examples of compile-time checks include:

- Validating printf and scanf format strings against the type of the actual arguments
- Using uninitialized memory locations
- Detecting unused variables
- Value tracking, which is used to detect errors such as
 - array subscript out of bounds
 - division by zero
 - constant conditions

Run-time error checking

The TASKING VX-toolset's run-time error checking capabilities in the compiler offer a wealth of checks that reveal run-time errors when they first occur. The kind of errors found by run-time error checking are typically hard to find since they manifest themselves through secondary effects or, in the worst case, will not manifest at all prior to your product being shipped. By identifying the source line where the error first occurs, the run-time error checking facilities reduce the time spent in the debugger, and increase the quality of your software. You can specify whether the application will terminate or continue when an error is detected.

These optional checks are implemented by generating additional code and/or enabling additional code in the standard C library. Run-time error checking has a nominal effect on code size and execution speed and can be enabled on a module-by-module basis, making it practical for use in debugging large applications.

The following types of checks are provided:

- Bounds checking verifies all pointer operations to detect buffer overflows and other illegal operations such as
 - accessing uninitialized or null pointers
 - accessing objects outside their declared bounds
 - illegal pointer arithmetic
- Malloc / free checks uncover dynamic memory allocation errors such as:
 - buffer overflow
 - write to freed memory
 - multiple calls to free
 - passing an invalid pointer to free

- Report an unhandled case value in a switch without a default part
- Stack overflow detects when the stack grows beyond its allocated size
- Divide by zero issues a message when a division by zero is attempted

Static Code Analysis

Static code analysis is a method to verify all possible paths within a software program without actually executing the program. A static code analysis tool can efficiently locate defects such as out of bound array accesses, memory allocation errors, arithmetic over and under flows, and inconsistent code fragments that go unnoticed during dynamic tests or peer reviews. Static code analysis can be applied early in the software development process, it can be applied on incomplete and incorrect code bases and when no test-cases have to be developed.

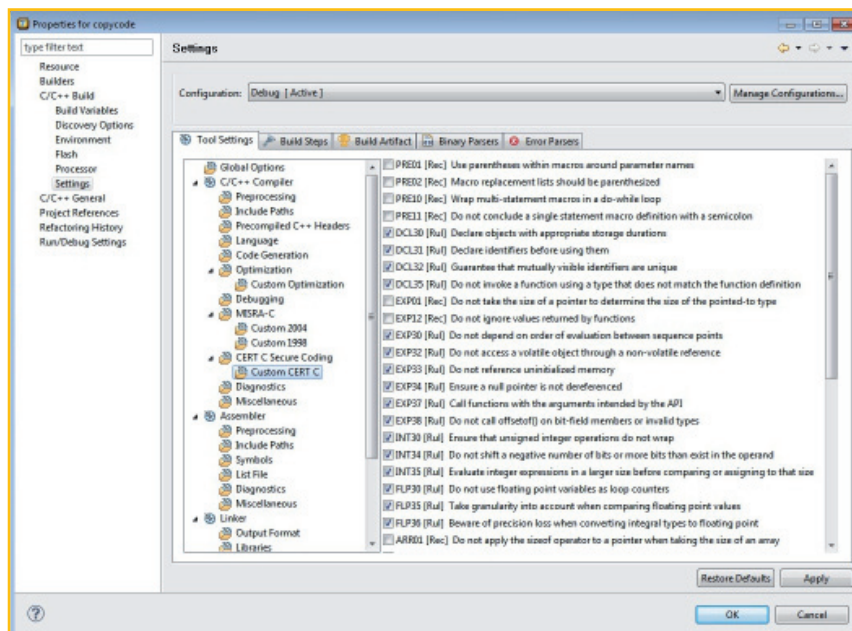
Altium has integrated static code analysis functionality for CERT C and MISRA C in its C compilers, with the advantage that such an analyzer is aware about specific embedded software issues such as: the existence of special function registers, the use of in-line assembly language, C-language extensions such as pointer and memory space qualifiers to address multiple address spaces, and DSP specific data types such as circular buffers, and fixed point data types.

CERT C

The CERT C/C++ secure coding standard is defined by the Computer Emergency Readiness Team (CERT), founded by the US government. Altium is one of the first vendors to provide a CERT C coding guidelines analyzer built into a C compiler for embedded software development.

MISRA C

MISRA C is driven by the Motor Industry Software Reliability Association and guides programmers in writing more robust C-code by defining selectable C-usage restriction rules. Through a system of strict error checking, the use of error-prone C-constructs can be prevented. The TASKING



Unique features TASKING VX-toolset

- The code generation chain compiler-assembler-linker is not based on open-source code
- Very stable and reliable code-generation compiler technology, with proven use at many automotive vendors
- Efficient optimizations for fast and small code
- Support for MISRA C guidelines
- Static Code Analysis for conformance to CERT C secure coding standard
- This ARM Cortex compiler is also part of Altium Designer
- Very cost attractive solution, due to shared investment costs with Altium Designer
- Parallel build support for efficient compilation on multi-core systems
- Eclipse based IDE with integrated debugger
- Legacy version support services available

C compiler supports the MISRA-C:1998 and MISRA-C:2004 guidelines.

CPU functional problem support

Semiconductor vendors regularly publish microcontroller errata sheets reporting deviations from the electrical and timing specifications. As an integral part of best practice architecture support, Altium's TASKING VX-toolset for ARM Cortex-M/R provides bypasses and checks for identified silicon defects. CPU functional problem support is provided throughout the complete toolset:

- C-compiler bypasses adapt code generation in order to avoid the identified erratic instruction sequences
- Assembler checks warn the assembly programmer for suspicious or erroneous instruction sequences
- Protected C-library sets are built with bypasses for all identified CPU functional problems

If reliability of your embedded application is essential, be sure to put support for CPU functional problems on your list of compiler selection criteria. Through its close co-operation with semiconductor vendors, Altium offers the most comprehensive support for this with its TASKING compilers.

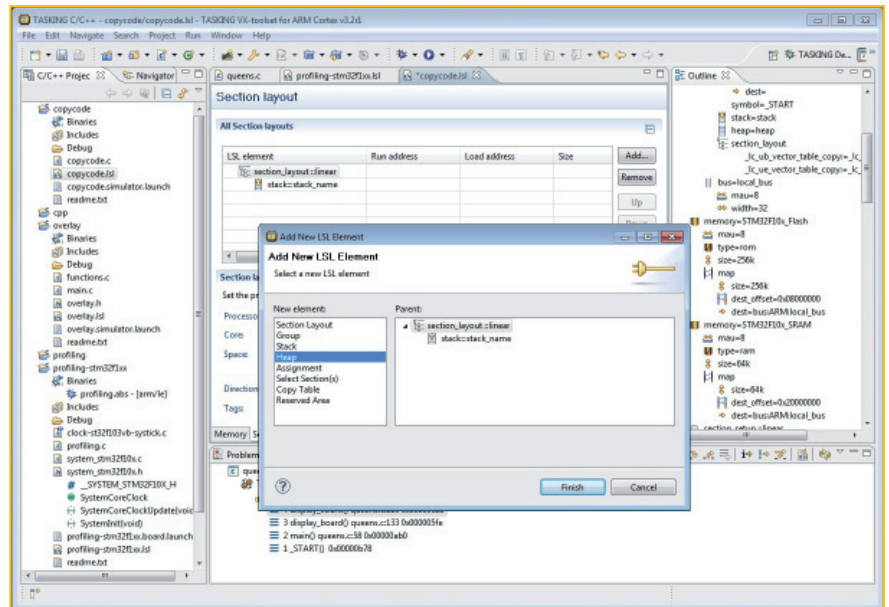
C++/EC++ compiler

Fully aware of the undeniable trend towards higher level language programming, the TASKING VX-toolset for ARM Cortex-M/R offers the full range of C++, C and assembly programming languages. Its ISO C++ compliant compiler allows developers to utilize the power of object-oriented design and coding techniques for the ARM family. The object-oriented benefits of C++ can be incorporated into your ARM application one module at a time, providing appropriate use of assembly, C and C++.

Fully compatible with the Embedded C++ (EC++) standard, the VX-toolset's C++ compiler can be configured to selectively disable C++ features that may not be essential for your embedded application. By selecting full or partial compliance with the EC++ standard, code-size overhead and run-time inefficiency can be minimized.

Essential tools

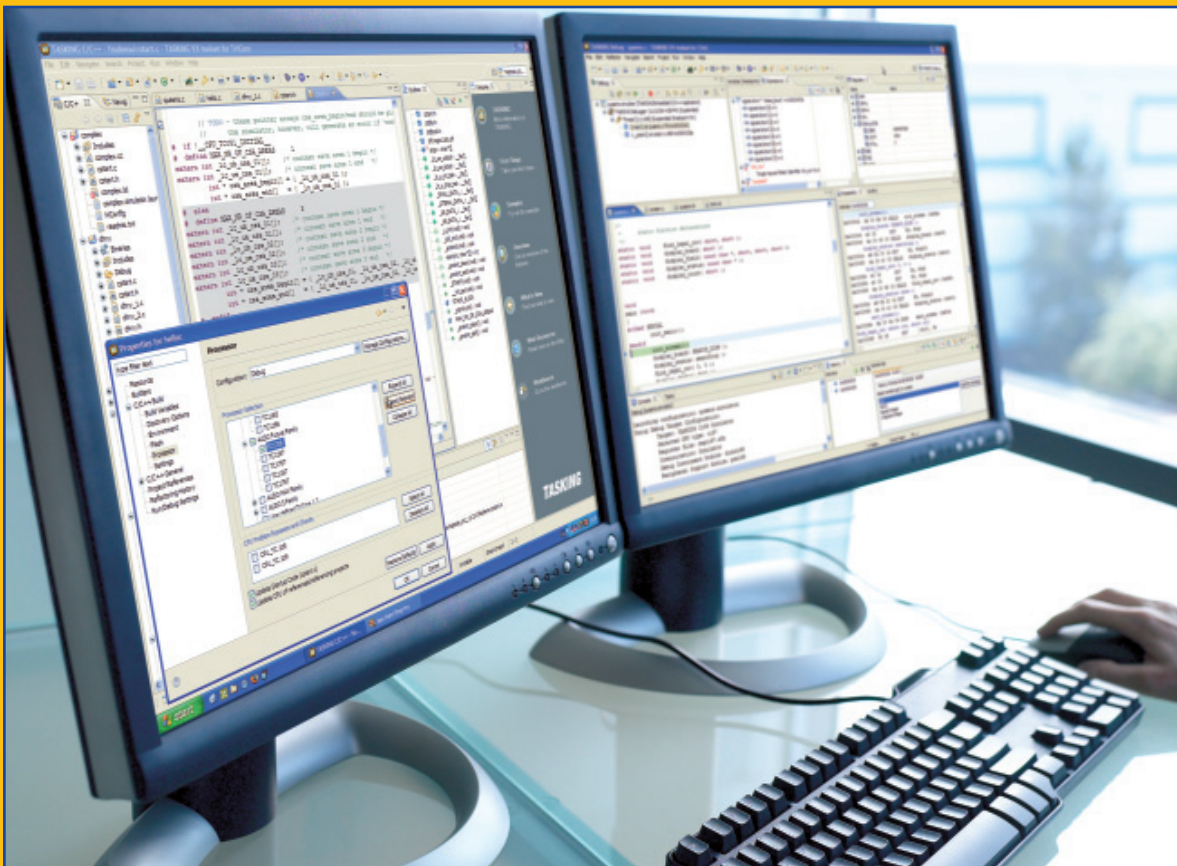
The VX-toolset includes a macro assembler, a linker/locator, C/C++/run-time/floating-point libraries with source code, a librarian for creating and maintaining user libraries, object file converter and many example projects or reference



Controlling the allocation, positioning and aligning of code and data.

designs. These projects and reference designs help you to get started with the VX-toolset quickly.

Through a versatile linker-script language the behavior of the linker/locator can be controlled for allocating, positioning and aligning of code and data. For tool interoperability, it supports the ELF/DWARF 3 standard and debugger specialists like Hitex, Lauterbach and PLS



Full control over editing and debugging your code.

offer compatibility to the VX-toolset with their respective products for debugging your embedded application.

A new generation *make* with support for faster parallel builds on multi-core platforms is also included.

Flash memory programming

Through the Eclipse IDE and the debugger you can download an application file to flash memory. It allows versatile configuration of external devices as well as microcontroller on-chip flash memory, and the debugger controls the actual flash programming algorithm through a small programming monitor.

Integrated debugger

The VX-toolset's debugger is based on Altium's latest debugger technologies released in 2006 with the TASKING VX-toolset for C166/ST10. The debugger has been redesigned from the ground up and made ready for market trends like integrated kernel-awareness and multi-core debugging. Utilizing the Eclipse IDE workbench it comes as a plug-in with a seamless integration to the editing environment. With the VX-toolset for ARM Cortex-M/R it provides two execution environments serving various debugging needs.

ARM instruction set simulator debugging

The ARM simulator debugger features instruction set simulation, allowing you to extensively debug your application on the host platform, even before your target hardware is available.

On-Chip debugging

Making the most of the On-Chip-Debug-Support facilities built into the microcontrollers, our debugger offers high quality in-circuit-emulation functionality at low cost. The VX-toolset has been tested and qualified with USB-JTAG debug probes from leading vendors like SEGGER and STMicroelectronics. The company SEGGER provides high-quality debug probes in various configurations, such as the cost attractive standard J-Link, the fast J-Link Ultra and the versatile J-Link Pro with Ethernet adapter.

From STMicroelectronics the extremely cost attractive ST-Link debug probe is supported. This probe can be used for all STM32 variants.

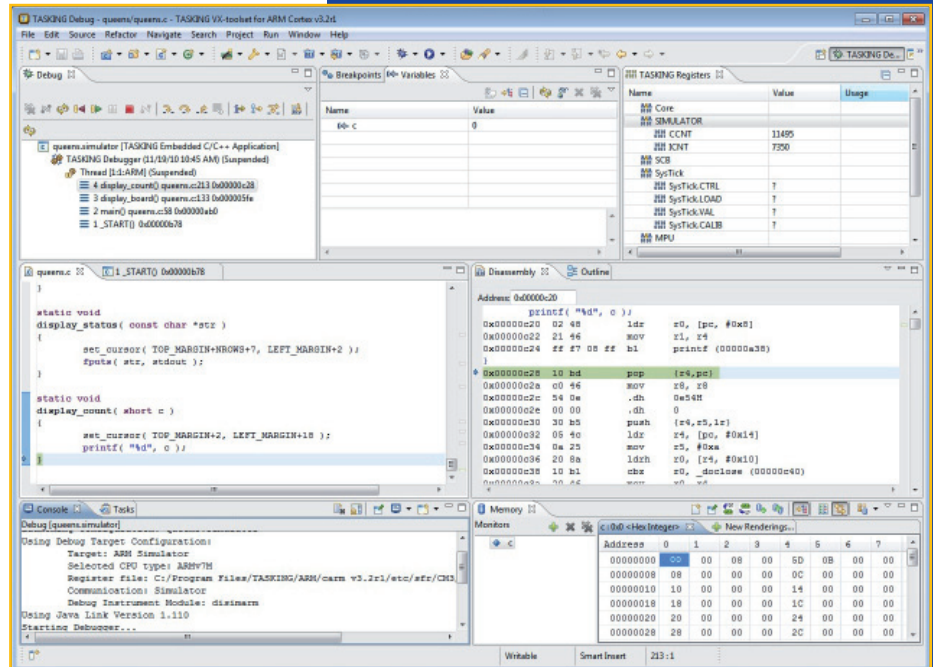
Several standard off-the-shelf evaluation boards are equipped with an on-board debug probe, such as the STM32 Discovery board. These boards can directly be connected to a PC through a simple USB cable for on-chip debugging.

Altium offers the J-Link and ST-Link debug probe variants as an option to the VX-toolset.

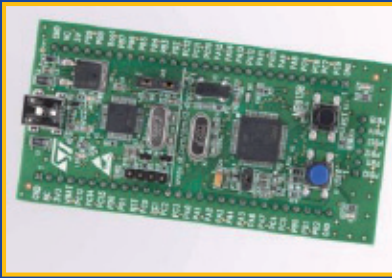
Toolset compatibility

The TASKING VX-toolset uses Altium's Viper compiler technology, which has a proven track record for generating stable, reliable, extremely compact and high-performance code. Because of its technical resemblance with for example the VX-toolsets for C166 and TriCore, the VX-toolset for the ARM Cortex-M/R provides an excellent migration path to or from Infineon's 16-bit or 32-bit microcontrollers.

When comparing Altium's TASKING VX-toolset for ARM with other vendor's tools, the major differentiator is that with the TASKING VX-toolset, the user is provided with a future roadmap into complete FPGA embedded systems design and a guarantee of software compatibility with Altium Designer. This is truly a unique position for the TASKING VX-toolset and provides you with the confidence that your decision future proofs you by providing you with a tool that meets today's needs, plus an entry point into Altium Designer and all the benefits that a unified electronic product development system offers for the future.



Eclipse debugger perspective.



Evaluation board and custom board support

The on-chip debugger in the TASKING VX-toolset supports a wide range of standard evaluation boards and starter kits for rapid and simplified verification of your embedded project on hardware. Standard boards are supported from respected vendors such as:

- STMicroelectronics
- Texas Instruments

A helpful wizard is provided in the Eclipse IDE to set up the debug configuration for your custom hardware.

Depending on the debugging platform of your choice, you can connect to hardware through a USB cable, or a debug probe from SEGGER or STMicroelectronics (both available from Altium).

Easy debugging of RTOS-based applications

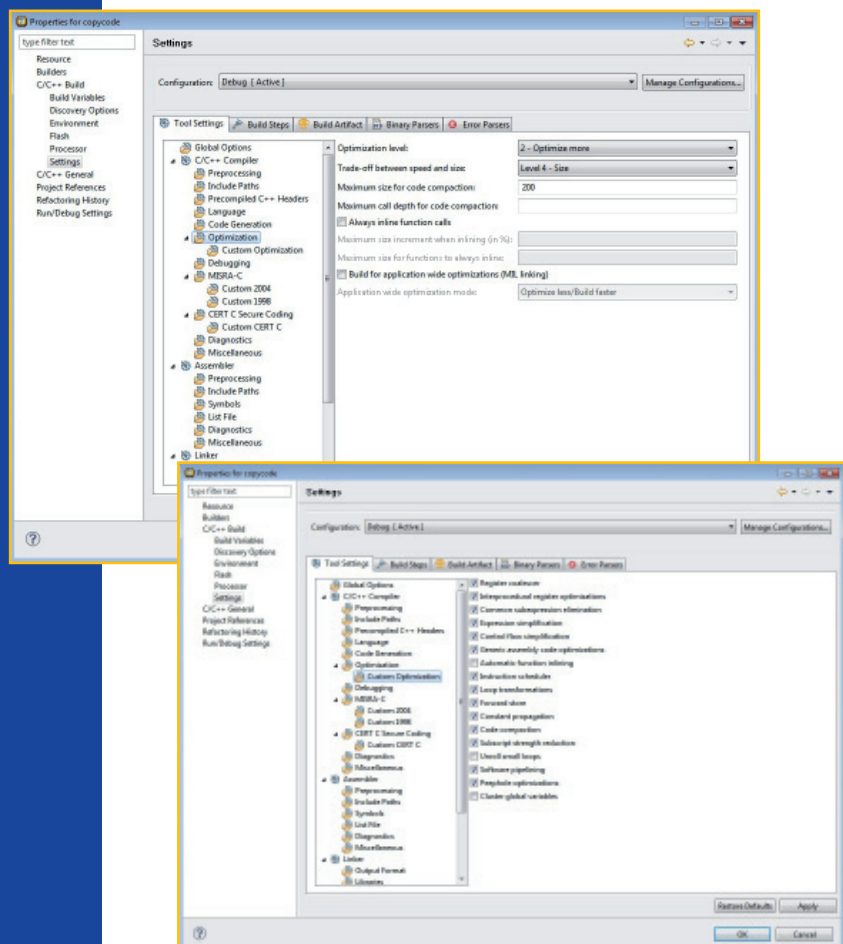
Altium's Kernel-aware Debugging Interface (KDI) defines an open standard interface between our debugger and an RTOS-Aware Debug Module (RADM). The RADM adds debugger capabilities to read, format and report kernel data structures for any commercial or proprietary RTOS. Our generic RADM for OSEK kernels, which is included in the toolset, is based on the ORTI 2.0 and 2.1 language specification.

The RADM extends the debugger with impressive kernel-aware debugging. Features include:

- Display levels of kernel information
- Examine and modify kernel data structures
- Obtain a summary of all tasks
- View contexts of tasks
- Inspect message contents (pipes, queues, mailboxes)
- Status of synchronization mechanisms
- Interrupt service routine status

Tool partner support for the VX-toolset

Our active and extensive third-party tool vendor program ensures that you have access to the tools you need to be most productive. Altium works closely together with all relevant manufacturers of In-Circuit-Emulators, Real-Time Operating Systems, evaluation boards, communication protocols (CAN, Flexray), and CASE and UML tools for the ARM architecture. Contact the tool supplier of your choice for information on Altium's TASKING VX-toolset compatible products, or consult our up-to-date third-party tool vendor catalog on our website.



SEGGER J-Link

Which bundle is the best for you?

Altium's TASKING VX-toolset is available in targeted bundles – Standard, Professional and Premium Edition, allowing you to choose the best fit for your application development activities.

The Professional Edition is the best choice if you develop a Cortex-M based application and if you want to have all essential development tools around the C/C++ compiler integrated into one environment. In addition to a software simulator, it offers you an on-hardware debugging solution – using an On-Chip Debug System (OCDS) solution with a USB-to-JTAG debug probe or wiggler. The OCDS debugger is the most cost-effective on-hardware debug solution you can get.

The Premium Edition* provides everything from the Professional Edition, but also supports the Cortex-R next to the Cortex-M. So if you develop your application based on a Cortex-R, this Premium Edition is a full suite solution for you.

Add one of the optional USB-to-JTAG debug probes to the Premium or Professional Edition in case your hardware board comes without an on-board wiggler. Altium supports and delivers various debug probes from SEGGER and STMicroelectronics.

The Standard Edition for Cortex-M is the most cost-attractive choice for C/C++ programming and debugging with a simulator. For debugging your code on hardware or programming your flash memory, you will require alternative tools from third parties. You can upgrade to the Professional or Premium Editions at a later time, offering you all included functionality under one single user interface.

* The Premium Edition will be available in H1 2011.

TASKING VX-toolset Editions for ARM Cortex-M/R

	Standard Edition	Professional Edition	Premium Edition
Eclipse based IDE	✓	✓	✓
Assembler, Linker/Locator	✓	✓	✓
C/C++ Compiler	✓	✓	✓
Support for Cortex-M	✓	✓	✓
CERT C secure code analyzer	✓	✓	✓
MISRA C code analyzer	✓	✓	✓
Simulator debugger	✓	✓	✓
On-Chip debugger		✓	✓
Flash programming support		✓	✓
Support for Cortex-R			✓
USB to JTAG debug probe		✓ ¹	✓ ¹

¹ Optional item available from Altium, manufactured by SEGGER or STMicroelectronics

Customer support

Altium is dedicated to providing quality products and support worldwide. This support includes program quality control, product update service, and support personnel ready to answer your questions by telephone, fax or email.

A 90-day technical support period is included with the purchase of TASKING toolsets and entitles you to enhancements and improvements as well as individual response to problems. Annual support agreements are available to extend this initial support period.

Availability of legacy versions

Upon ordering a product from Altium one normally receives the latest release, but Altium is also able to deliver popular legacy versions for existing projects that require the availability of compatible compiler releases. Special support and improvement services for legacy versions are also available from Altium.

License management

Altium's TASKING VX-toolset for ARM Cortex-M/R includes the industry standard FLEXlm license manager, offering stability as well as flexibility. Its license 'borrowing' functionality is a popular feature, allowing laptop users to take a license from the network license pool for the period of time they are off-site, saving on cost for individual licenses.

The FLEXlm license manager also allows you to upgrade at a later stage to one of the more extensive VX-toolset Editions, giving you additional functionality. This is done by issuing you a new FLEXlm license key which simply unlocks the functionality in the existing installation on your system.



Product Information

Standard Edition for Cortex-M

Product code: 07-200-103-804

Package contents: Eclipse IDE, C/C++ compiler, assembler, linker, simulator debugger

Professional Edition for Cortex-M

Product code: 07-200-160-806

Package contents: Eclipse IDE, C/C++ compiler, assembler, linker, simulator and on-chip debugger

Premium Edition for Cortex-M and R*

Product code: 07-200-160-808

Package contents: Eclipse IDE, C/C++ compiler, assembler, linker, simulator and on-chip debugger

Please check the table on page 9 for a detailed overview of the contents/features in the various VX-toolset Editions.

*planned for H1 2011

ST-Link USB-JTAG debug probe

Product code: 07-290-103-001

STMicroelectronics' In-Circuit Debugger/ Programmer for STM32 microcontrollers

J-Link debug probe

Product code: 07-290-103-011

SEGGER's USB driven JTAG interface

J-Link Pro debug probe

Product code: 07-290-103-012

SEGGER's USB and Ethernet driven JTAG interface

J-Link Ultra debug probe

Product code: 07-290-103-013

SEGGER's Hi-Speed USB 2.0 driven JTAG interface

The debug probes are available as an option to the Professional or Premium Edition.

Fully-functional trial version

A fully-functional 15-day trial version of the TASKING VX-toolset for ARM Cortex-M/R is available on CD-ROM or downloadable from TASKING's website: www.tasking.com/arm

The trial version only counts the days of effective use, which allows for an extensive evaluation period. The on-chip debugger can be evaluated using a J-Link debug-probe from SEGGER, or the ST-Link from STMicroelectronics. Also select STM32 evaluation boards include an on-board ST-Link that connect directly to the on-chip debugger.

Minimum system requirements

PC/Windows platform:

- 2 GHz Pentium-class CPU
- 3 GB free disk space
- 2 GB memory
- 1024 x 768 display

Supported platforms and operating systems:

- Windows XP, Windows Vista, Windows 7 (32-bit, 64-bit¹)
- Sun/Solaris
- PC/Linux²

1. The VX-toolset is provided as 32-bit executable.
2. Selected versions of the VX-toolset only.

For more details contact your local Altium Sales and Support Center or Reseller.

More Information

- Contact your local Altium Sales and Support Center or Reseller
www.tasking.com/contacts
- To order Altium's TASKING VX-toolset for ARM Cortex-M/R
www.tasking.com/arm

Altium offices worldwide

North America

Altium Inc.
3207 Grey Hawk Court
Suite 100
Carlsbad, CA 92010
Ph: +1 760-231-0760
Fax: +1 760-231-0761
sales.na@altium.com
support.na@altium.com

Germany

Altium Europe GmbH
Philipp-Reis-Straße 3
76137 Karlsruhe
Ph: +49 (0) 721 8244 300
Fax: +49 (0) 721 8244 320
sales.de@altium.com
support.eu@altium.com

France

Protel SA
(Filiale du groupe Altium Limited)
121 rue d'Aguesseau
92100 Boulogne-Billancourt
Ph: 0800 88 05 06
Fax: 0800 82 85 92
email.info.fr@altium.com
support.eu@altium.com

Australia

Altium Limited
3 Minna Close, Belrose
NSW 2085
Ph: +61 2 8622 8100
Fax: +61 2 8622 8140
sales.au@altium.com
support.au@altium.com

China

Altium Information Technology
(Shanghai) Co., Ltd
9C, East Hope Plaza
No.1777 Century Avenue
Shanghai 200122
Ph: +86 21 6182 3900
Fax: +86 21 6876 4015
sales.cn@altium.com
support.cn@altium.com

Japan

Altium Japan K.K.
Nomura Fudosan Yotsuya Bldg 7F
2-12-1 Yotsuya
Shinjuku-ku, Tokyo
160-0004
Phone: +81 3 6672 6155
Facsimile: +81 3 6672 6159
sales.japan@altium.com
support.japan@altium.com

The Netherlands

Altium Technology Centre
Altium BV
Saturnus 2
3824 ME Amersfoort
Ph: +31 33 4558584
Fax: +31 33 4550033
tasking@altium.com

For a full list of TASKING resellers visit **www.tasking.com/contacts**

The Altium logo, featuring the word "Altium" in a bold, italicized, sans-serif font, with a small trademark symbol (TM) to the right.

Copyright © 2011 Altium Limited. Altium, Altium Designer, Board Insight, DXP, Innovation Station, Live Design, NanoBoard, NanoTalk, OpenBus, P-CAD, SimCode, Situs, TASKING, and Topological Autorouting and their respective logos are trademarks or registered trademarks of Altium Limited or its subsidiaries. All other registered or unregistered trademarks referenced herein are the property of their respective owners, and no trademark rights to the same are claimed.