

HIGHLIGHTS

- Incorporates next generation Viper compiler technology
- State-of-the-art C compiler
 - Fast and compact
- MISRA C code checking
- CrossView Pro debugger
 - R8C/Tiny instruction set simulator
 - TASKING ROM monitor
 - Renesas E7 emulator interface
- Integrated R8C/Tiny Flasher
- Complete integrated development environment
- Available for PC/Windows

THE TASKING R8C/TINY SOFTWARE DEVELOPMENT TOOLSET

The R8C/Tiny family from Renesas (a Mitsubishi Electric and Hitachi company) is one of the most versatile ranges of 16-bit microcontrollers on the market, providing developers of embedded software with features such as high performance, flash memory and a wide range of peripherals.

Altium's TASKING R8C/Tiny toolset brings to developers the power of Altium's latest in-house, next generation compiler technology framework, allowing them to take full advantage of this high-performance architecture. Codenamed Viper, it generates code with the level of execution speed and code density needed for tomorrow's industrial, security and consumer applications.

EMBEDDED DEVELOPMENT ENVIRONMENT

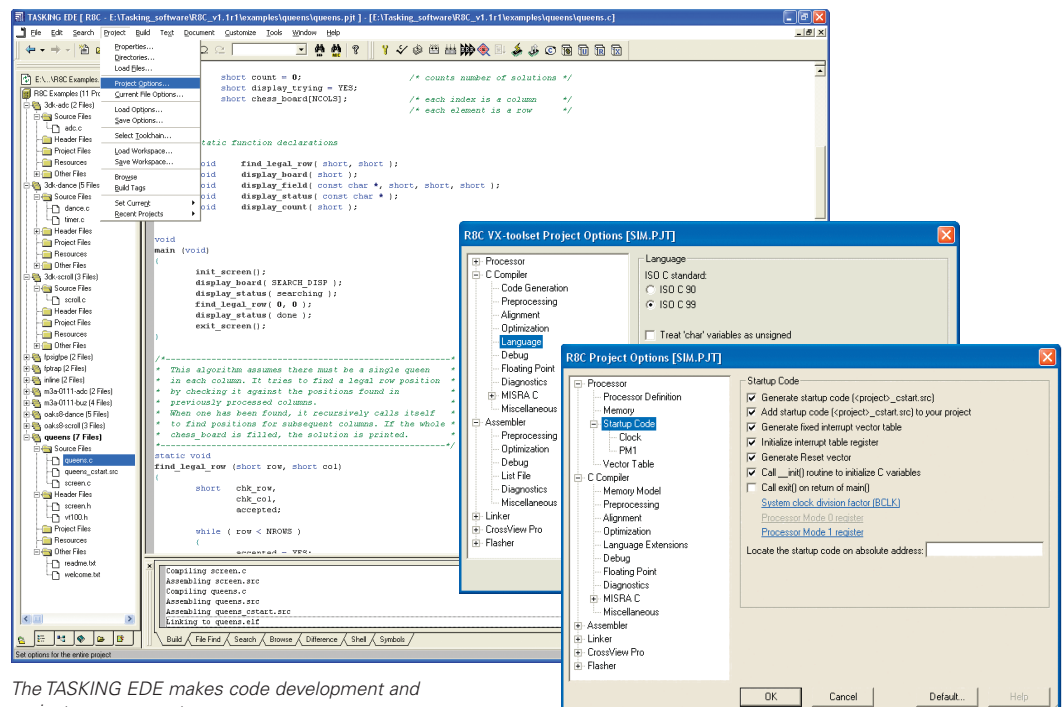
With the TASKING R8C/Tiny Embedded Development Environment (EDE), you can create and maintain projects easily. All project related aspects, such as the application source files, the tool options (compiler, assembler, linker/locator, CrossView Pro debugger), file management, and the options of the build process, are managed from one central point. File dependencies, as well as the sequence of operations required to build the application, are handled automatically.

The R8C/Tiny EDE offers many productive features for application and code development, which include:

- An explorer-like treeview control that allows simplified configuration of the TASKING tools and the R8C/Tiny target processor for the experienced, as well as the novice, user
- Menu structure that is tuned according to the development work flow, offering an intuitive project management setup
- Easy selection of target processor and new edit controls for project settings
- HTML View Window allowing you to browse through the product manuals, project or code documentation, or even surf the net without leaving the EDE
- Project Spaces that enable you to group multiple projects in one view, offering improved project management for more complex developments
- CodeSense advanced coding assistance that offers rich type-ahead features, which help you in selecting the next expected function parameter or available structure members. When positioning your mouse pointer over a function name, the function prototype will be displayed
- CodeFolio to enable easy insertion of template code, adding to coding efficiency and consistency. It allows macro expansion and prompted input as you insert the code
- Split Windows that provide full control over source code by allowing you to split your file horizontally or vertically into as many as four edit windows

RENESAS

R8C
Tiny



The TASKING EDE makes code development and project management easy

C COMPILER

Based upon Altium's latest C compiler technologies, the TASKING R8C/Tiny 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 R8C/Tiny architecture.

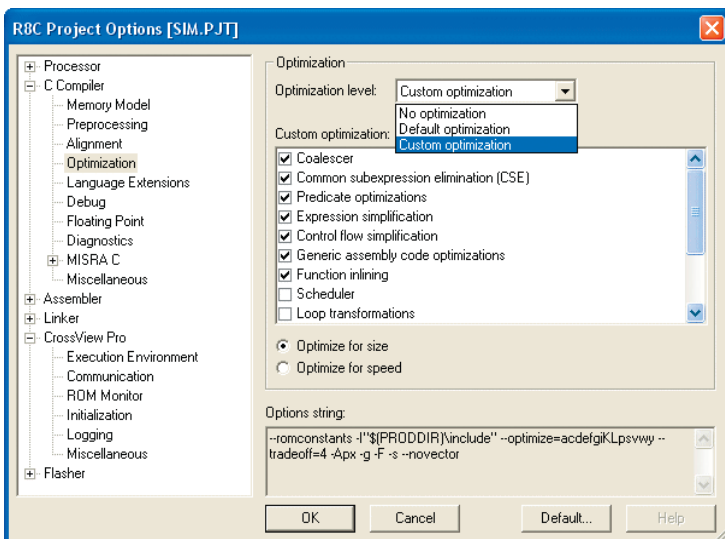
The TASKING compilers are tested for ISO C'99 and ANSI C conformity against authoritative validation suites such as Plum Hall and Perennial. Additionally, the optimization techniques of the compilers are tested with various large real-world applications as well as industry benchmark standards such as Nullstone.

Fast and Compact Code

The R8C/Tiny compiler generates the most optimal code. Based on Altium's new Viper compiler technology, the R8C/Tiny toolset produces a compiler generated code in its default configuration code, which is small and has a fast execution speed. Depending on the specific requirements of your R8C/Tiny application, optimizations can be tweaked towards smaller code size or higher execution speed.

Compiler optimizations include:

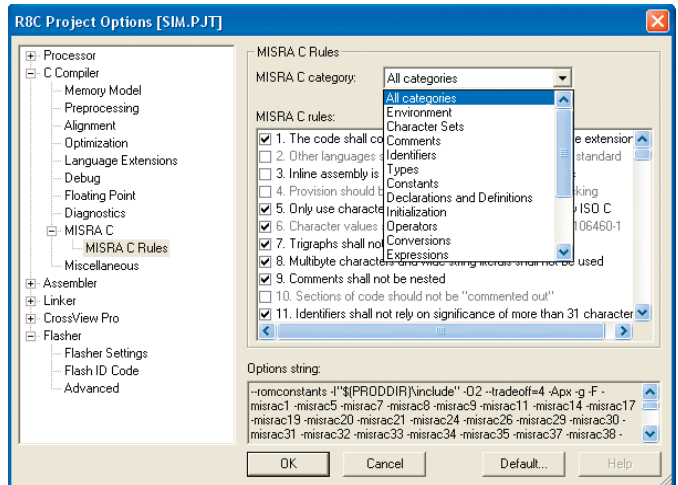
- **Partial Redundancy and Elimination (PRE), which detects and eliminates repeating (sub-) expressions**
- **Various Loops and Jump optimizations, which speed up execution and reduce code size**
- **Control-flow and code reduction optimizations, which remove dead code and perform transformations to minimize jumps**
- **Function inlining, which replaces calls to small functions with inlined copies of the function code**
- **Constant propagation, where a variable with a known constant value is replaced by that constant**
- **Code compaction, which identifies identical code sequences, creates a new function from the sequence, and replaces the sequences with calls to the new function**



Many optimizations are possible for both code size and execution time

MISRA C

Based on the 'Guidelines for the use of the C language in vehicle based software' published by the Motor Industry Software Reliability Association (MISRA®), Altium is the first to implement the MISRA C concept in a software development environment. MISRA C guides programmers in writing more robust C-code by defining selectable C usage restriction rules. Through a system of strict code checking, the use of error-prone C constructs can be prevented.



Fully configurable MISRA C code checking

A predefined configuration for compliance with the MISRA guidelines is available with a single click. It is also possible, using pull-down menus, to enable a custom set of MISRA C rules to suit specific company requirements.

To ensure compliance with the MISRA C rules throughout the entire project, the R8C/Tiny linker/locator can generate a MISRA C Quality Assurance report.

This report lists the different modules in the project with the respective MISRA C configurations which were used to compile them.

Under the guidance of MISRA C in the toolset, programmers can now write better, more maintainable code that contains less error-prone C constructs, which in turn will lead to more robust and safer embedded systems.

R8C/TINY ARCHITECTURE SUPPORT

As an integral part of our renowned architectural support, the TASKING R8C/Tiny toolset provides full support for all available R8C/Tiny derivatives. Definitions and SFR header files are supplied, along with automated Cstart code adaptation.

The supported R8C/Tiny derivatives are:

- R8C/10
- R8C/11
- R8C/12
- R8C/13
- R8C/14
- R8C/15
- R8C/16
- R8C/17

The TASKING R8C/Tiny toolset offers a wealth of built-in intrinsic functions. Intrinsic functions appear as normal C functions, but the code generator interprets them and, if possible, generates more efficient code. Several pre-declared functions are available to generate inline assembly code at the location of the intrinsic function call, ensuring faster execution by avoiding the standard function calling and parameter-passing overhead.

INDUSTRY STANDARD LIBRARIES

The TASKING R8C/Tiny compiler contains all the necessary ISO C libraries, run-time libraries, and floating-point libraries. The floating-point libraries are supplied in a number of highly useful variants which include single precision and trapping and non-trapping.

Source code for most of the library routines allows you to tailor the libraries to your specific application.

ASSEMBLER

The TASKING assembler is supplied complete with linker/locator, librarian, and object format utilities.

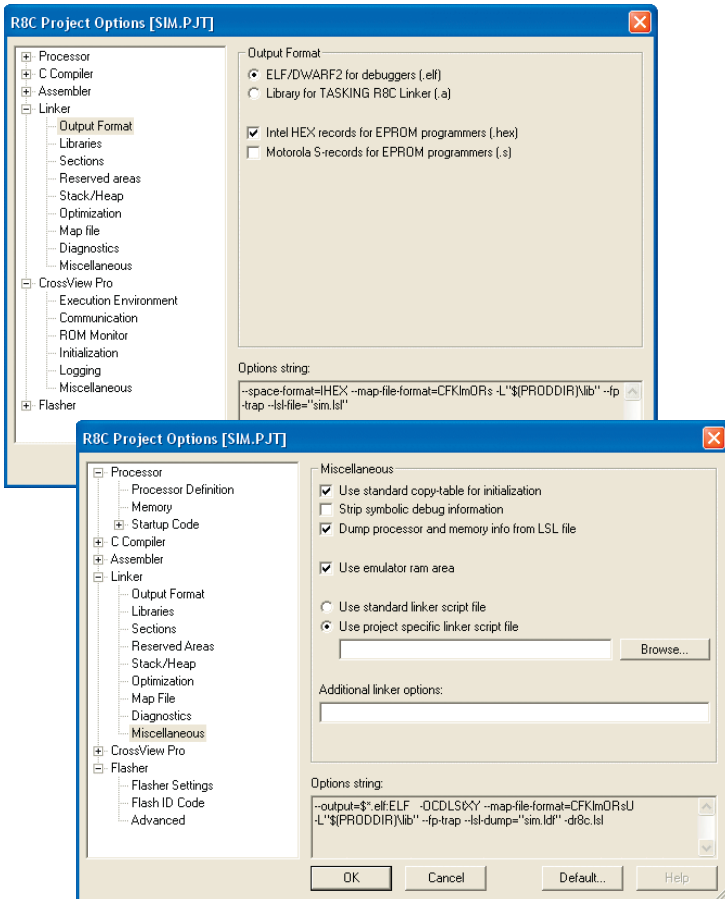
Assembler features include:

- Full macro and conditional assembler
- Optimizing jump/call instructions
- Extensive section directives
- Full assembly source level debugging

LINKER/LOCATOR

The linker/locator plays a pivotal role in the software building process by combining the compiler- and assembler-generated code and data sections with possible library functions and allocating the result into available target memory.

The new target-independent linker in the TASKING R8C/Tiny toolset allows you to accurately describe available target memory and fully control the behavior of the locating process, so that all pieces of code and data fall into their intended places.



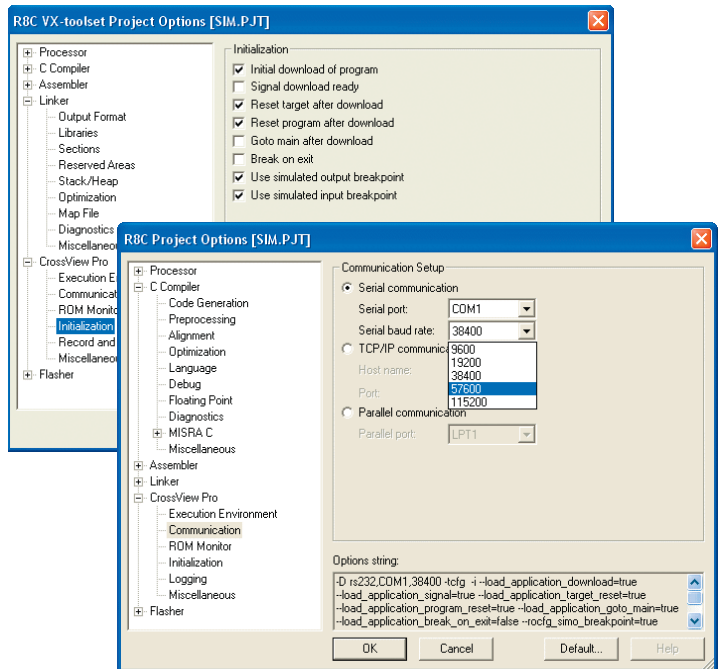
Select the object output format desired

Linker/Locator features include:

- Automatic and user-specified allocation in target memory
- Powerful, intuitive linker script language
- Data/code section initialization
- Powerful data/code overlaying facilities
- Smart linking removing unreferenced code/data
- Industry Standard ELF/Dwarf object output format
- SREC and Intel HEX ROM image output formats

CROSSVIEW PRO DEBUGGER

The TASKING R8C/Tiny CrossView Pro debugger is a perfect partner for checking, verifying and debugging your application. With its easy-to-use interface and powerful, extensive debugging features, CrossView Pro helps you debug your applications faster. CrossView Pro provides multiple, resizable and independently controlled windows.



Select the execution environment desired

Crossview Pro allows you to choose the windows you need to view the relevant aspects of your code during debugging. It combines the flexibility of the C language with the control of code execution found in assembly language, bringing functionality that reduces the amount of time spent on testing and debugging.

Functionality enhancements include:

- Basic through to advanced debugging features
- Tracking scope and monitoring locals
- Intuitive navigation through the source window
- Double click, right mouse button, and tip-point functions
- Clipboard copy and paste
- Bubble-Spy™ technology for easy and quick inspection of variable contents
- Code/data coverage and profiling (performance analysis) in CrossView Pro Simulator

Source Window

The main window is the Source window. It allows you to view source; step through your code; set and clear breakpoints, assertions and code coverage markers; watch and show variables; search for strings, functions, lines and addresses; and evaluate expressions. The Source window can display code in C source, assembly, or mixed.

While moving the mouse pointer over your source, our Bubble-Spy™ technology provides quick checks of variable values and function prototypes. Double clicking on a function call automatically navigates to the corresponding source.

From the cursor in the debugger source window, you can jump directly into the EDE editor, allowing immediate access to the source line of the problem that needs correcting.

Multiple Information Windows

The CrossView Pro debugger offers a wealth of information windows allowing you to monitor and modify data objects, CPU registers, memory locations, and the stack.

The Data window enables you to watch and modify data objects. Data structures can be shown collapsed as well as expanded. Objects can be displayed in any format on an element-by-element basis.

Register windows can be configured to display any set of CPU registers and their values. Defining multiple Register windows helps you organize your focus.

The Stack window displays the contents of the function-call stack frame. You can easily configure stack-level breakpoints, navigate to the function call's source, and monitor local variables for selected functions.

The Memory window enables you to monitor and modify any memory location, with complete control over size and format of the data, as well as view coverage of the memory range.

All information windows are automatically updated, and changed values are highlighted for easy identification. In-situ editing allows you to modify values on the spot.

Advanced Breakpoints

Breakpoints halt program execution and return control to the user. In addition to industry standard code and data breakpoints, you can configure your application to halt based upon instruction counts, cycle counts, or timer counts. All types of breakpoints can be defined as 'stop-and-go' probe points.

Probe points briefly halt and immediately resume execution of the application. During the brief period that the application is halted, only user-specified actions will be performed. Through this mechanism, probe points allow least-intrusive debugging of time critical applications.

Finally, any number and type of breakpoints can be combined into so-called breakpoint-sequences. This allows easy specification of the most complex conditions that need examining.

Multiple Execution Environments

The CrossView Pro debugger supports multiple execution environments with a standard interface.

■ R8C/Tiny Instruction Set Simulator debugger

The simulator environment allows you to test, debug, and monitor the performance of code in a known and repeatable environment independent of target hardware. It uses the same description file as the linker/locator when locating your application and therefore knows exactly where and how memory is mapped.

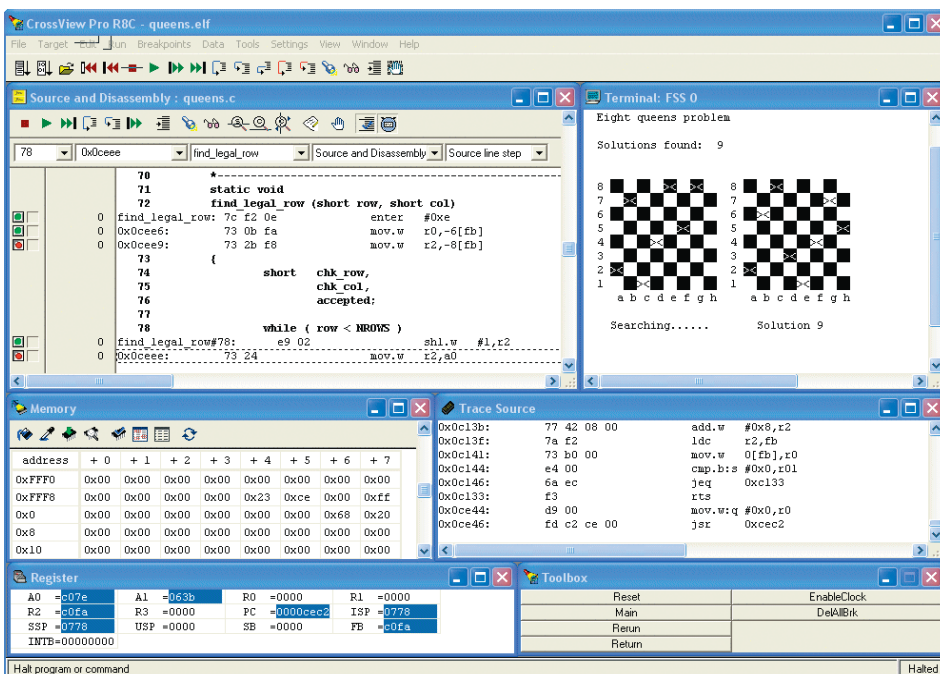
All CrossView Pro features, including C level trace, Code/Data Coverage, performance analysis (profiling), and unlimited amount of code and data breakpoints are available to you, allowing you to test code before target hardware is available.

■ The TASKING ROM monitor debugger

The ROM monitor debugger is a powerful debug instrument that can be used to debug a target without requiring an expensive emulator. The key benefit of the TASKING ROM monitor is that it does not need any regular polling by the host platform and thus offers optimal run-time performance. The 3D starterkit is supported by the TASKING ROM monitor. As the ROM monitor is provided in C source, it can be ported to new or custom hardware easily.

■ Renesas E7 Emulator

CrossView Pro also includes support for the Renesas E7 emulator. The E7 offers substantial debugging functionality, such as Break and Trace functionality. Connection between the E7 and the host PC is via a USB interface.



Spend less time debugging with CrossView Pro

TASKING ROM Synchronization Tool

Integrated in the EDE is the TASKING ROM synchronization tool. The 'Sync' dialog, which is used with the ROM monitor debugger, will synchronize project settings like processor and memory settings. Once the TASKING ROM monitor has been flashed onto the target board using this tool, it will simplify setting the target specifics of your project. By clicking the Sync button, your project settings can be updated with the target specifics found by the TASKING ROM monitor.

File System Simulation

CrossView Pro I/O Simulation (IOS) allows the use of standard ISO C system calls such as open(), read(), printf() and scanf() within your embedded application in order to interface with the host platform file I/O services.

Using IOS, you can read from and write to files on the host platform or a CrossView Pro Virtual I/O window directly. I/O Simulation will work in any CrossView Pro target execution environment.

Program Performance Analysis

CrossView Pro provides a number of performance analysis capabilities to help you further optimize your application as well as shorten your debug session.

■ Code Coverage

Code coverage enables you to check whether specific parts of your application code actually have been executed. Based on the code coverage reports, you can build a complete test suite for your product and improve the quality of your application.

■ Profiling

Profiling allows you to perform timing analysis on the complete application or specific parts of it. Profiling information can be shown in the left margin of the source window, and can also be presented in a report, providing you with a full overview. Based upon this profiling information, you can easily decide which functions should be optimized for speed.

■ Graphical Data Analysis

CrossView Pro's outstanding programmable Graphical Data Analysis simplifies quick detection of gross errors in signal processing routines. By displaying large sets of data in meaningful visual diagrams, CrossView Pro allows you to analyze the data without the need of reviewing or post-processing large files of raw data. You can also view the same set of data in several ways at the same time (e.g. in time and frequency domain).

Four different analysis types are provided:

- x-t plotting
- FFT (Fast Fourier Transformation)
- Power spectrum
- Eye diagram

The C-language scripts for these pre-defined graphs can be easily used as the basis for custom data analysis windows.

Easy Debugging of RTOS-based Applications

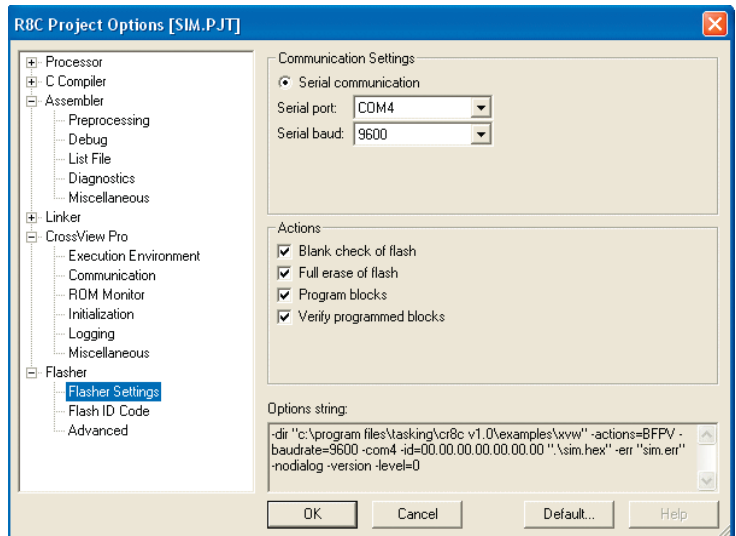
TASKING's Kernel-Aware Debugging Interface (KDI) defines an open standard interface between CrossView Pro and an RTOS-Aware Debug Module (RADM). The RADM can be used to add kernel-awareness to CrossView Pro for any commercial or proprietary RTOS.

CrossView Pro's kernel-aware debugging capabilities enables:

- 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

TASKING R8C/TINY FLASHER

Integrated within the TASKING EDE is an easy to use flash tool called the TASKING R8C/Tiny Flasher. This flasher interfaces to the standard Renesas flash tool available on chip. The TASKING R8C/Tiny Flasher allows you to directly flash a Motorola S-Rec or Intel Hex file into the chip. Flashing a file can be done via a USB connection (USB to Serial).



Unique TASKING Flasher integrated in EDE

CO-OPERATION WITH THIRD PARTIES

Our extensive third party co-operation ensures that you have access to the tools you need to be most productive. Altium works closely together with manufacturers of In-Circuit-Emulators, Real Time Kernels, TCP/IP connectivity, CAN solutions and Evaluation boards. For more detailed information of R8C/Tiny partners, please visit www.altium.com/tasking/R8C

CUSTOMER SUPPORT

When you purchase a TASKING product, it is the beginning of a long-term relationship. Altium is dedicated to providing quality products and support worldwide. This support includes program quality control, a product update service, and support personnel ready to answer your questions by telephone, fax or email.

A maintenance period is included with the purchase of TASKING products and entitles you to enhancements and improvements as well as individual response to problems. Annual maintenance agreements are available to extend the initial support period.

PRODUCT PACKAGING AND ORDERING CODES

Each TASKING product comes with full printed documentation. The documentation is also available on-line and provides full-text search capabilities for quick and easy lookup of topics. The TASKING R8C/Tiny Software Development Toolset is available for PC/Windows.

Product Code	Package Contents
07-200-199-024	<ul style="list-style-type: none">- EDE/Editor- ISO C'99 Compiler- Assembler- Linker/Locator- CrossView Pro Simulator- ROM monitor debugger- Renesas E7 Emulator interface- TASKING R8C/Tiny Flasher

Demonstration versions of the R8C/Tiny tools are available on CD-ROM or downloadable from our website at www.altium.com/tasking/R8C

INTERNET

Website: www.altium.com/tasking/R8C
Developer's forum: www.yahogroups.com/group/TASKINGforum

DISTRIBUTOR

ALTIUM SALES OFFICES

North America

Altium Inc

17140 Bernardo Center Drive, Suite 100
San Diego, CA 92128
Toll Free: 877-TASKING
Tel: 858 521 4280
Fax: 858 485 4610
E-mail: tasking.sales.na@altium.com

Asia - Pacific

Japan - Altium Japan K.K.

Resona Gotanda Building 7F
1-23-9, Nishi-Gotanda
Shinagawa-ku Tokyo 141-0031
Tel: 03 5436 2501
Fax: 03 5436 2505
E-mail: tasking.sales.jp@altium.com

Australia - Altium Limited

Level 14, 39 Murray Street
Hobart TAS 7000
Free Call: 1800 030 949
Fax: 03 6231 4167
E-mail: sales.au@altium.com

Europe

Germany - Altium Europe GmbH

Albert-Nestler-Straße 7
76131 Karlsruhe
Free Call: 0800 0 258486 (0800-0-ALTIUM)
Fax: (0)721 82 44 320
E-mail: tasking.sales.de@altium.com

Switzerland - Protel AG

(A subsidiary of Altium Limited)
Unterdorfstrasse 1
CH-4334 Sisseln
Tel: (0)62 866 41 11
Fax: (0)62 866 41 10
E-mail: tasking.sales.ch@altium.com

From Austria

Free Call: 00800 776 776 77
Free Fax: 00800 776 776 00

From France

Free Call: 0800 88 05 06
Free Fax: 0800 82 85 92

European Free Call Numbers

German speaking: 00800 776 776 77
French speaking: 00800 776 776 55
English speaking: 00800 776 776 44
Free Fax: 00800 776 776 00