StarCore DSP® Family

SOFTWARE DEVELOPMENT TOOLSET

TASKING
Embedded software development from Altium

HIGHLIGHTS

- Highly efficient StarCore compiler toolset
- ISO C'99 and StarCore ABI compliance
- Complete integrated development environment
- Highly optimizing C/C++/EC++ compiler:
  - Fast AND Compact
- MISRA C enhanced code checking
- SC 110/140 support:
  - Derivatives support
  - Auto Cstart code
- CrossView Pro debugger:
  - StarCore I/S simulator
  - On Chip Emulation
- Available for:
  - PC/Windows
  - SUN/Solaris

VX
VIPER COMPILER
TECHNOLOGY BY ALTIUM

StarCore
Brighter DSP Technology

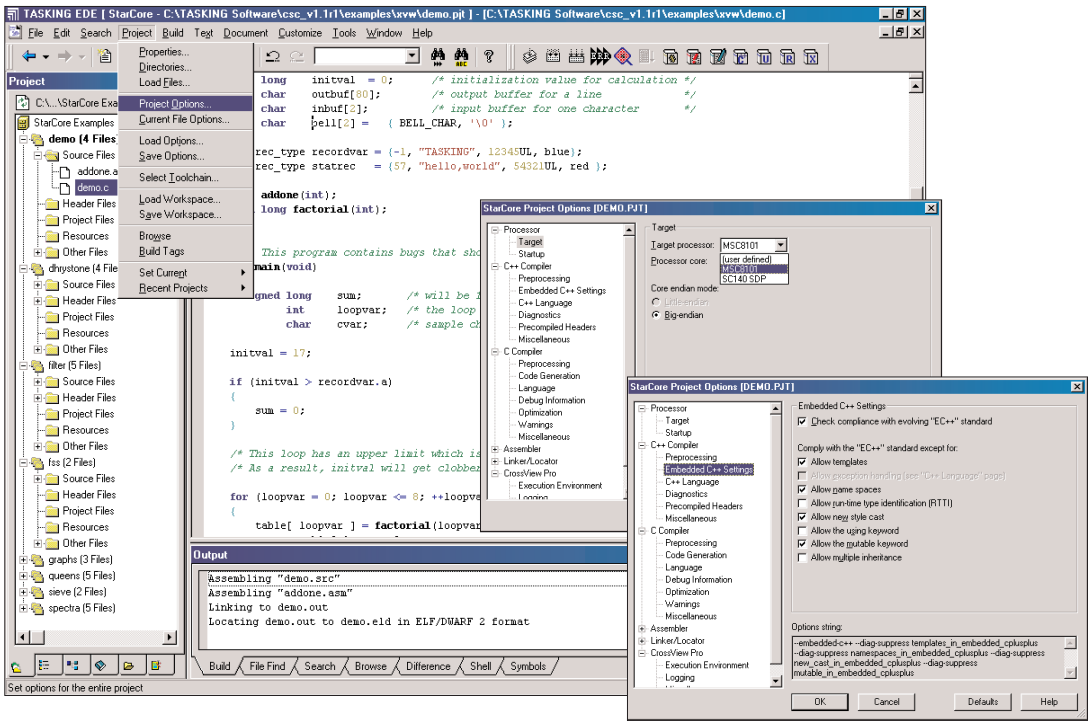
THE TASKING STARCORE TOOLSET

The StarCore DSP is a development of StarCore LLC, and is designed as an IP component for system-on-chip design environments. The core is especially suited to high-performance DSP applications such as wireless communications, 3G mobile phones, networking infrastructure, DSL/cable modems, and automotive.
Built upon Altium's in-house DSP compiler and debugger expertise, the TASKING StarCore Software Development Toolset provides the level of execution speed and code density needed for today's high-end DSP applications. In addition to supplying high performance development tools, the TASKING StarCore toolset offers an excellent migration path for TASKING DSP56k programmers.

EMBEDDED DEVELOPMENT ENVIRONMENT

With the TASKING StarCore Embedded Development Environment (EDE), you can create and maintain projects the easy way. 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 StarCore EDE offers many productive features for application and code development:

- **Project Spaces allow grouping of multiple projects in one view, thus offering improved project management for more complex developments.**
- **Right-Mouse-Button clicks expedite a variety of tasks within the EDE (e.g. creating new files, adding files to a project, etc.)**
- **CodeSense advanced coding assistance offers rich type-ahead features, which help 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.**



The TASKING EDE makes code management easy.

- **Tags Browsing offers a graphical overview of the applications' cross-references and allows easy navigation through the available variables and functions.**
- **CodeFolio allows easy insertion of 'snippets' of template code, thus adding to coding efficiency. It also allows macro expansion and prompted input as you insert the code.**
- **HTML View Window allows browsing through the product manuals, project or code documentation or even surfing the net.**
- **XML Collapsible Grid Viewer displays the hierarchy of elements and element attributes in XML documents.**
- **Split Windows provide full control over source code by allowing you to split your file horizontally or vertically to up to four edit windows.**

FULL STARCORE SC100 ARCHITECTURE SUPPORT

The TASKING StarCore tools are fully prepared for the complete range of StarCore SC100 architecture design steps, including SC110 and SC140. In particular, the SC140's four Arithmetic Logic Units (ALU) are fully exploited by the StarCore compiler's generated code.

StarCore DSP derivatives support

TASKING software tools are renowned for their excellent architectural support. In adherence to this (internal) standard, the TASKING StarCore tools provide full support for all publicly available StarCore derivatives, right from the beginning. Supported StarCore devices include:

- **MSC8101**
- **MCS8102**
- **StarPro 2000**
- **and more …**

C++/EC++ COMPILER

More and more DSP applications that previously have been written in assembly language now require the application of higher-level languages such as C and C++ to meet todays requirements. Fully aware of this undeniable trend, the TASKING StarCore toolset offers the full range of C++, C and assembly programming languages. Its ISO C++-compliant compiler delivers the power of object-oriented design and coding techniques to the StarCore family.
The object-oriented benefits of C++ can be incorporated into your StarCore DSP application one module at a time, providing appropriate use of Assembly, C and C++.

Scalable C++

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

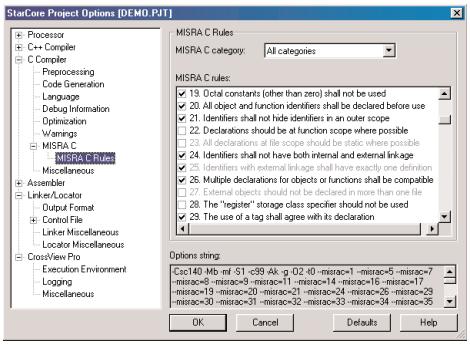
C COMPILER

Based upon TASKING's renowned DSP C compiler expertise, the StarCore C compiler is reliable, compliant, competitive, easy to use, and allows you to take full advantage of the StarCore DSP architecture by generating the most optimal code possible.
The TASKING StarCore compiler suite is tested for ISO and ANSI C conformance 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.
The StarCore C compiler features include:

- **ISO C'99 and StarCore ABI compliance**
- **Most efficient code output**
- **Complete C++/C and runtime libraries**
- **Easy to learn / easy to use**

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 only vendor offering advanced code checking in standard software development tools with their TASKING products. Through a configurable system of strict code checking, programmers are guided in writing more robust, consistent C-code.

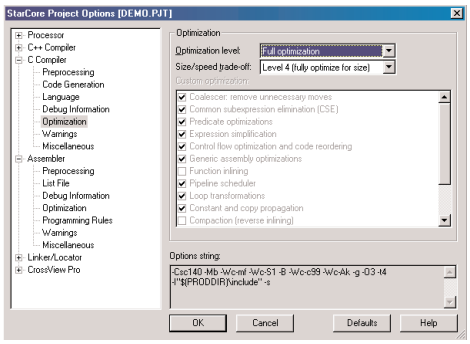


Write safer code with MISRA C

A predefined configuration for compliance with the required rules in MISRA C guidelines is ready prepared, but it is also possible to enable a custom set of MISRA C rules to suit company requirements.
To ensure compliance with the MISRA C rules throughout the entire project, the StarCore 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, and can then be filed under the company's Quality Assurance system.

Fast AND compact code

Altium understands that you expect your StarCore compiler to produce the most optimal code possible with no fuss. Thanks to its extremely powerful optimization techniques, the TASKING StarCore compiler already generates code that delivers an even balance between high execution speed and minimum size in its default configuration. Depending on the specific requirements of your StarCore DSP application, the optimizations can be further tweaked towards smaller code size or higher execution speed, respectively.



You choose which matters most: code size or execution speed.

Compiler optimizations include:

- **Common Sub Expression Elimination 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 as to minimize jumps.**
- **Function inlining replaces calls to small functions with inlined copies of the function code.**
- **The Instruction Pipeline scheduler rearranges instructions to take advantage of the parallel capabilities of the StarCore architecture.**
- **Software pipelining increases parallelism in loops by executing iterations in an overlapped fashion.**
- **Peephole optimizations replace instruction sequences with equivalent but faster and/or shorter sequences, or remove obsolete instructions.**

StarCore architectural support

In addition to the data types mentioned in the StarCore ABI and the ISO C '99 standard, the TASKING StarCore compiler provides a number of DSP-specific data types, such as *ssfract, sfract, fract, lfract, llfract* and *accum.* The compiler supports these as true data types instead of typedef / intrinsic constructs, and as such they offer many benefits in the form of program consistency and enhanced legibility.
Near and Far function call modes allow specification of 32-bit absolute or 20-bit PC-relative calling conventions, thus offering the ability to access code throughout StarCore's memory reach.
Data memory models Tiny, Small and Big can be used to specify the compiler's default mode of accessing data objects. This allows you to select the most efficient data access method applicable for your particular application.
The TASKING StarCore toolset offers a wealth of built-in intrinsic functions. Intrinsic functions appear as normal C functions, but the code generator interprets them in order to generate more efficient code. Several pre-declared functions are available to generate inline assembly code at the location of the intrinsic function call, ensuring fastest execution by avoiding the standard function calling and parameter passing overhead.

User inline C functions and inline assembly

The *inline* keyword enables the definition of user inline C-functions. To deliver the fastest possible implementation, inline function calls are stripped from their 'calling and parameter passing' overhead and copied into the code. If you prefer, you can do your assembly programming from within the C environment. The TASKING StarCore industry standard *asm()* inline assembly smoothly blends in the higher level C environment through passing of arguments and return values, as well as the use of compiler optimizable scratch registers.

Industry standard libraries

The StarCore compiler toolset contains all the necessary ISO C++ / ISO C libraries, run time libraries, and floating-point libraries. In particular, the floating-point libraries are supplied in four highly practical variants:

- **Double- and single-precision variants offer the choice between IEEE 754-1985 compliant precision and increased speed.**
- **Trapping and non-trapping variants offer the choice between run-time error checking and increased speed.**

STARCORE MACRO-ASSEMBLER

Originally, the main parts of DSP applications were written in assembly. In spite of the increasing size and complexity of today's DSP applications and the availability of advanced DSP C compiler technologies, you may still want to take full control and write that bit of time-critical code in assembly. For this purpose the TASKING StarCore toolset incorporates a powerful macro-assembler.

StarCore assembler features include:

- **Full compliance with the StarCore ABI**
- **Full macro and conditional assembler**
- **Branch/call instruction optimizations**
- **Extensive section directives**
- **Error file with textual error reporting**
- **Versatile list file generation**
- **ELF/Dwarf object output format with HLL debugging extensions**

Extensive assembly restriction checking

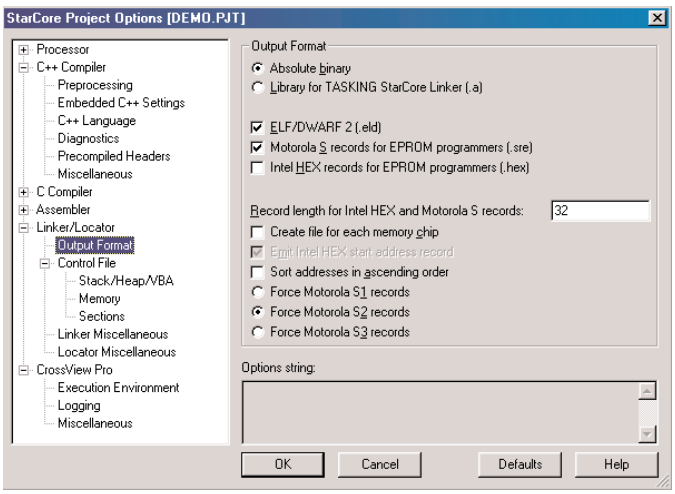
The StarCore instruction set offers a wealth of powerful instructions. However, as with other modern DSP architectures, not all combinations of instructions are actually supported by the architecture. The TASKING StarCore assembler is fully equipped with an integrated model of the core's instruction set database. This allows you to test your assembly code against restricted combinations of parallel, or consecutive, instructions, and warns you of potential errors by means of clear error messages.

LINKER/LOCATOR

The linker/locator is an essential part of the software building process that combines the compiler- and assembler-generated code and data sections with possible library functions and allocates the result into available target memory.
The TASKING StarCore toolset allows you to accurately describe available target memory and fully control the behavior of the location process, so that all pieces of code and data fall into their intended places.

Linker/locator features include:

- **Automatic or user defined allocation of code and data in memory**
- **Data/Code section initialization**
- **Powerful overlaying facilities**
- **Incremental linking**
- **Industry standard ELF/Dwarf 2 object output format including HLL debug-information**
- **Complete map file**
- **SREC and Intel HEX ROM image output formats**



Take full control with the TASKING linker/locator

CROSSVIEW PRO DEBUGGER

In the world of embedded DSP applications, there's no such thing as 'right first time'. When developing today's complex, high-performance DSP applications, a fair share of development time will be spent in the debugger validating your code.
The TASKING StarCore CrossView Pro debugger is a perfect partner in checking, verifying and debugging your DSP 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. You 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 includes:

- **Simple as well as advanced debugging features**
- **Intuitive source window**
- **Tracking scope and monitoring local variables**
- **Bubbly-Spy™ for easy inspection of variables and functions**
- **Double click and right mouse button functions**
- **Clipboard copy and paste**

Source window

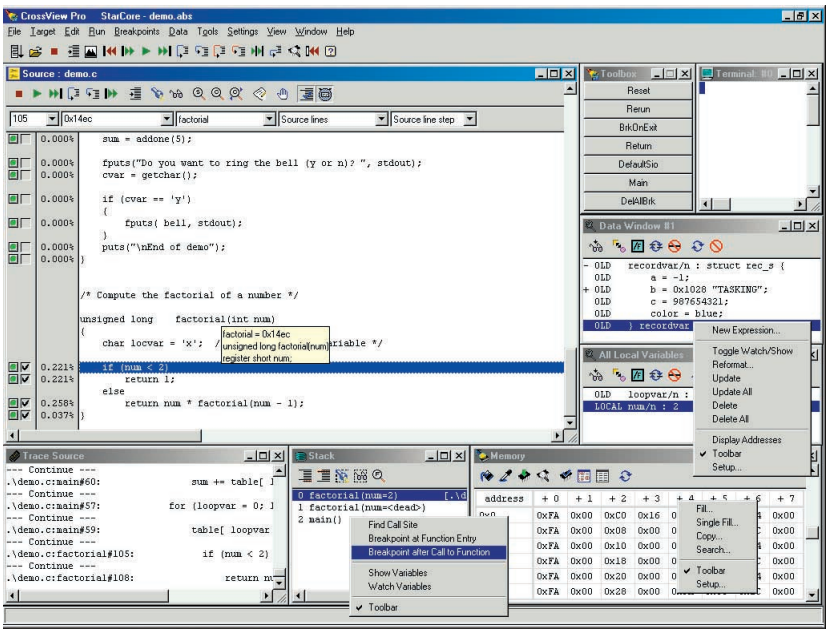
The source window is the main debugging window. It allows you to view source; step through your application; set and clear breakpoints and assertions; watch and show variables; search for strings, functions, lines and addresses; call functions; evaluate expressions; and view performance analysis data.
The source window can display code in C++/C source, assembly, or a mixed mode that allows a simultaneous view of your C++/C source intermixed with the corresponding assembly code.

Multiple information windows

CrossView Pro offers a wealth of information windows that allow you to navigate through your application and monitor and modify Data objects, CPU registers, memory locations and the function-call stack.

The Data window enables you to watch or show data, browse for locals or globals, double-click to modify values or to expand and contract complex data structures. Within this window, you can reformat (change display of radix and type) on an element-by-element basis. You can show or watch locals from any stack level, automatically track and display locals, and easily copy any variable as show or watch.

Register windows allow display and modification of CPU register values. Register windows are fully configurable to display any set of CPU registers. By defining multiple Register windows you can easily organize your focus.



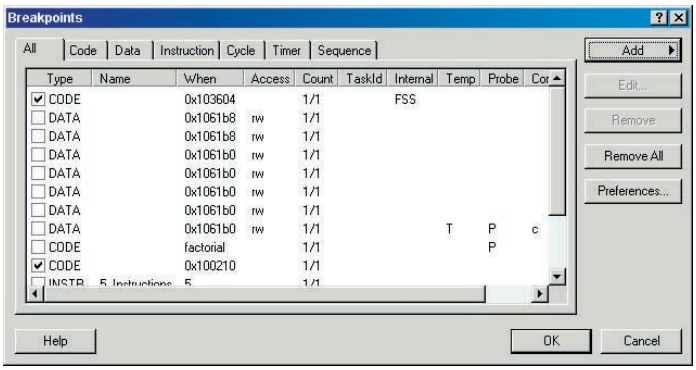
Spend less time debugging with CrossView Pro

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.

Advanced breakpoints

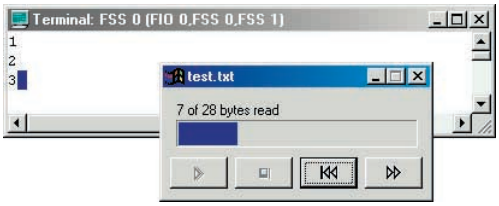
Breakpoints halt program execution and return control to the user. In addition to industry standard code and data breakpoints, you can configure your application to halt based upon Instruction counts, Cycle counts, or Timer counts. All types of breakpoints can be defined as 'stop-and-go' probe-points.
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 time critical applications to be debugged as unintrusively as possible. Finally, any number and type of breakpoints can be combined into 'breakpoint sequences'. This allows easy specification of the most complex conditions that need examining.



Advanced breakpoints put you in control

I/O 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 which interface with the host PC file I/O services. Using IOS, you can read from and write to files on the host PC or a CrossView Pro Virtual I/O window directly. I/O Simulation will work in any CrossView Pro target execution environment: I/S Simulator, EOnCE and supported In-Circuit-Emulators.



Multiple execution environments

CrossView Pro supports multiple execution environments with the same standard interface.

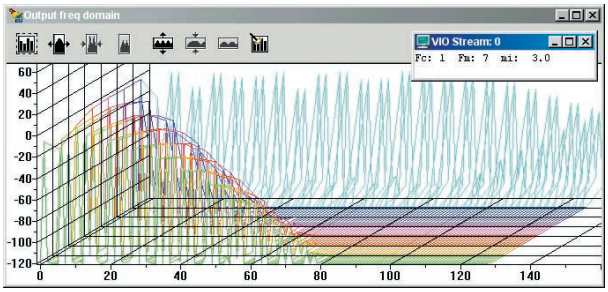
- **StarCore Instruction Set Simulator.** With CrossView Pro and the bundled StarCore Instruction Set Simulator, you can debug your application on the host platform even before your target hardware is available. The Simulator supports all instructions of the StarCore Instruction Set.
- **Target debugging through EOnCE.** Taking advantage of the StarCore On-Chip-Emulation (OnCE) facilities, CrossView Pro offers high quality in-circuit-emulation functionality at a low cost. Via the host PC's parallel interface, CrossView Pro can communicate with the StarCore device directly.

Program performance analysis

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

- **Code Coverage and Profiling.** Code coverage enables you to check whether specific parts of your application code have actually 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 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, but is also presented in table format, providing you with a full overview. Based upon the 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 such as those typically found in DSP applications. By displaying large sets of data in meaningful visual diagrams, CrossView Pro allows you to analyze the data without the need for 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).

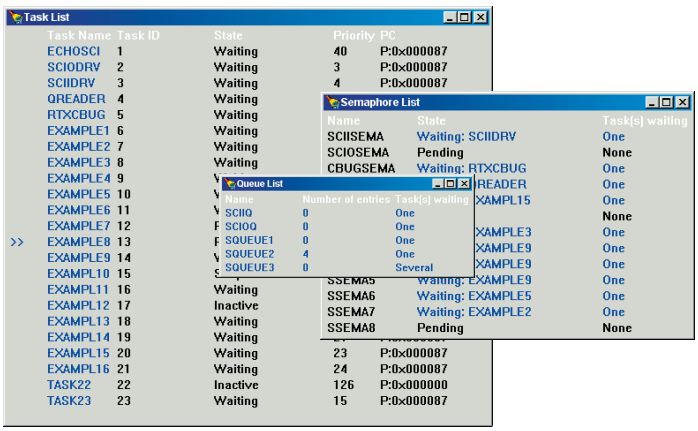
Four different analysis types are available: x-t plotting, x-y plotting, FFT (Fast Fourier Transformation) power spectrum, and Eye diagram. The C language scripts for these pre-defined graphs can be easily used as the basis for custom Data Analysis windows.



View large sets of data in a meaningful form

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 adds the capability to CrossView Pro to read, format, and report kernel data structures.



Add kernel-awareness to CrossView Pro with a RADM

The KDI specification describes the open interface that can be used to add kernel-awareness to CrossView Pro for any commercial or proprietary RTOS.

The RADM extends CrossView Pro with impressive Kernel-Aware Debugging capabilities, such as:

- **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**

TASKING's RTOS support via a RADM enables you to debug your RTOS-based application more quickly and smoothly.

TASKING

Embedded software development from Altium

COOPERATION WITH THIRD PARTIES

Our extensive third party cooperation ensures that you have access to the tools you need to be your most productive. Altium works closely together with manufacturers of In-Circuit-Emulators, Real Time Kernels and Evaluation boards for the StarCore.

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, 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 this initial support period.

PRODUCT PACKAGING & ORDERING CODES

Each TASKING product comes with full printed documentation. This documentation is also available on-line in the form of a Windows Help systems, HTML and PDF and provides full-text search capabilities for quick and easy access to topics.

Product Code	Package contents
TK100-024	EDE, C/C++/EC++ and MISRA C compiler, assembler, linker/locator, CrossView Pro EOnCE/simulator debugger

Demonstration versions of the StarCore Software Development toolset are available on CD-ROM. Contact your local Altium sales office or distribution to order your copy.

INTERNET

Web site: www.tasking.com
Developers forum: www.yahoogroups.com/group/TASKINGforum

DISTRIBUTOR

STARCORE-ds-1-1-641

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 KK
ASAHI-GIN Gotanda Building 7F
23-9, Nishi-Gotanda 1-chome
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: 1 800 030 949
Fax: 03 6231 4167
E-mail: sales.au@altium.com

Europe

Germany - Altium Germany GmbH
Albert-Nestler-Straße 7
D-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 4110
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
Fax Free Call: 00800 776 776 00

641-SPGEETG-1

Altium
Making Electronics Design Easier™

www.tasking.com