

## HIGHLIGHTS

- **World's best-selling TriCore compiler toolset and selected by AUTOSAR alliance**
- **Full TriCore EABI and ISO C'99 compliance**
- **State-of-the-art C/C++/EC++ compiler**
  - Fast and compact
- **MISRA C code checking**
- **TriCore 1 and 2 support:**
  - All derivatives
  - Auto Cstart code
  - FPU / MMU / PCP1-2
- **CrossView Pro debugger**
  - TriCore I/S simulator
  - On Chip Debug
  - OSEK kernel aware
  - Supports Infineon's DAS
- **Complete integrated development environment**
- **Includes free to use TCP/IP reference design**
- **Available for:**
  - PC/Windows
  - SUN/Solaris

## THE TASKING TRICORE VX-TOOLSET – THE VIPER EXPERIENCE

The TriCore™ architecture from Infineon Technologies is a revolutionary single-core 32-bit microcontroller/DSP design that is already in its second generation. The architecture offers embedded developers the best of three worlds: the real-time capabilities and on-chip peripherals of a versatile microcontroller, the computational power of a DSP, and an excellent price/performance ratio for a RISC load-store architecture.

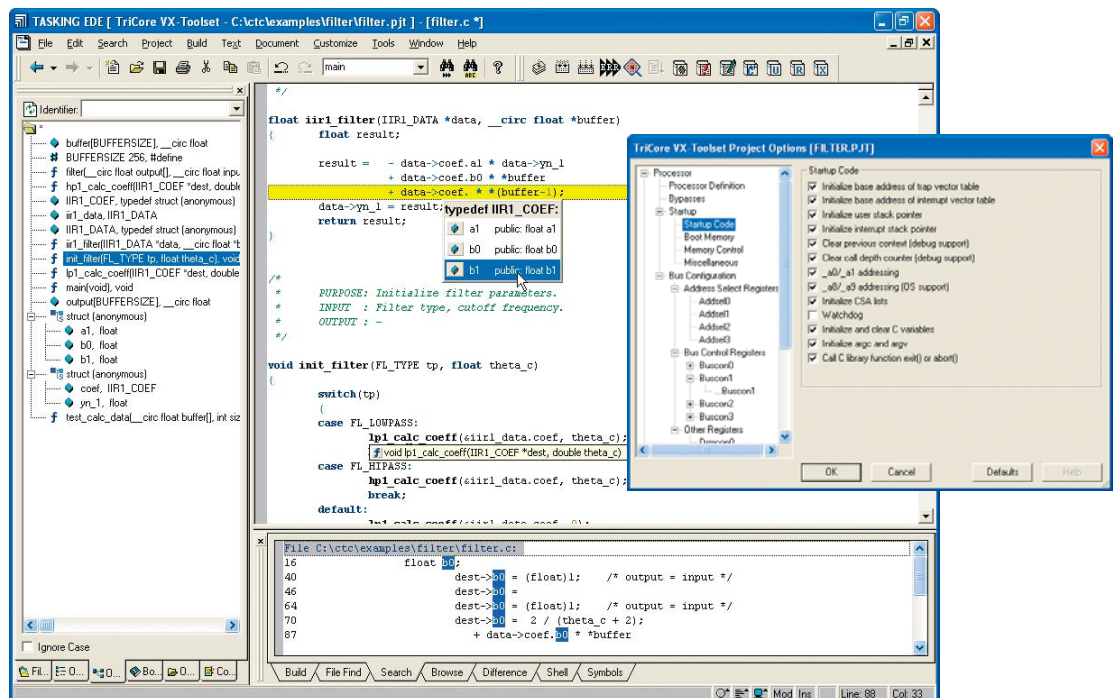
The TASKING TriCore VX-toolset brings to developers the power of Altium's latest in-house, next-generation compiler technology framework, allowing them to take full advantage of this high-performance architecture. Codenamed Viper, it generates code with the level of execution speed and code density needed for tomorrow's automotive, industrial and telematics applications. In addition, the TASKING TriCore VX-toolset offers an excellent migration path for C166 programmers.

## EMBEDDED DEVELOPMENT ENVIRONMENT

With the TASKING TriCore VX-toolset 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 EDE of the TriCore VX-toolset offers many productive features for application and code development, such as:

- **Project Spaces** that enable you to group multiple projects in one view, thus offering improved project management for more complex developments
- **CodeSense** advanced coding assistance that offers rich type-ahead features, assisting 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
- **Tags Browsing** which offers you a graphical overview of the applications' cross-references, and allows easy navigation through the available variables and functions
- **CodeFolio** that allows you to easily insert 'snippets' of template code, thus adding to coding efficiency, making possible macro expansion and prompted input as you insert the code
- **HTML View Window** to allow you to browse through your project or code documentation or even surf the net without leaving the EDE
- **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
- **Right-Mouse-Button** clicks to expedite a variety of tasks within the EDE (e.g. creating new files, adding files to a project, etc.)



The TASKING EDE makes code development and project management easy

## FULL TC1 AND TC2 ARCHITECTURE SUPPORT

The TASKING TriCore VX-toolset is compliant to Infineon's TriCore EABI standard and fully prepared for the complete range of Infineon TriCore architecture revision steps, including TC1v1.2, TC1v1.3 and TC2. The TriCore VX-toolset compiler automatically deploys the specific features of the selected TriCore architecture step or derivative.

Additionally, specific support for the more advanced TriCore extensions is included in the TriCore- and PCP Assembler:

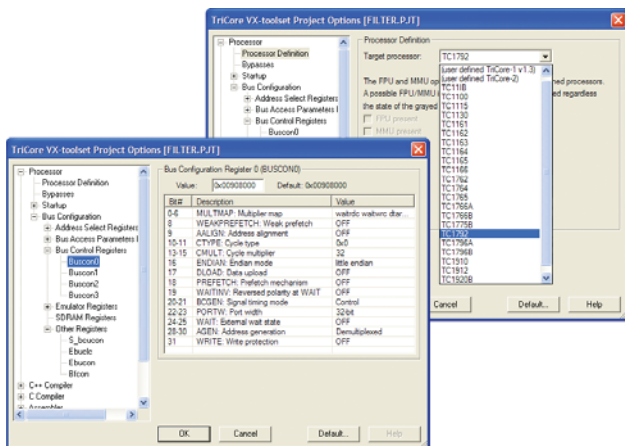
- **FPU (Floating Point Unit)**
- **MMU (Memory Management Unit)**
- **PCP2 (Peripheral Control Processor 2)**

## TriCore derivatives support

As an integral part of our renowned architectural support, the TASKING TriCore VX-toolset provides full support for all available TriCore derivatives. This includes pre-prepared definitions and SFR header files, as well as automated Cstart code adaptation.

Supported TriCore devices include:

- **TC111B**    ■ **TC1161**    ■ **TC1165**    ■ **TC1775**
  - **TC1100**    ■ **TC1162**    ■ **TC1166**    ■ **TC1796**
  - **TC1115**    ■ **TC1163**    ■ **TC1765**
  - **TC1130**    ■ **TC1164**    ■ **TC1766**
- And more...



## CPU functional problem support

Infineon Technologies regularly publishes microcontroller errata sheets reporting deviations from the electrical and timing specifications. As an integral part of best practice architecture support, the TASKING TriCore VX-toolset provides bypasses and checks for identified silicon defects.

CPU functional problem support is provided throughout the complete tool chain:

- **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 TriCore CPU functional problems**

## C++/EC++ COMPILER

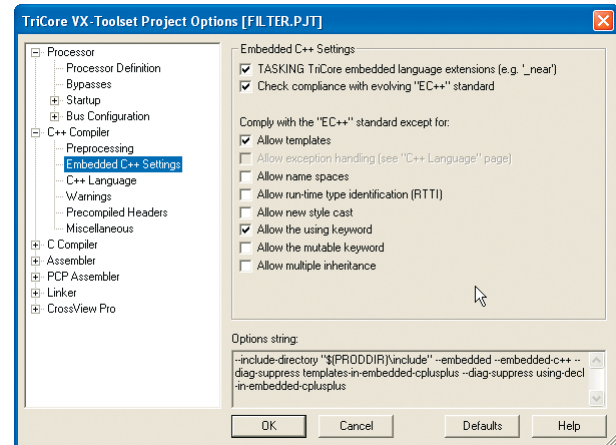
Fully aware of the undeniable trend towards higher level language programming, the TASKING TriCore VX-toolset offers the full range of C++, C and assembly programming languages.

Its ISO C++ compliant compiler allows developers to utilize the power of object-oriented design and coding techniques for the TriCore family. The object-oriented benefits of C++ can

be incorporated into your TriCore 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 TriCore VX-toolset 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.



Minimize overhead by using the scalability of EC++

## C COMPILER

Based upon Altium's latest DSP-C compiler technologies, the TASKING TriCore VX-toolset C is reliable, compliant, competitive, complete, easy to use and generates the most optimal code possible to allow you to take full advantage of the TriCore architecture.

The TASKING TriCore VX-compiler suite is tested for ISO C'99 and ISO C++ conformity against authoritative validation suites, such as Perennial. In addition, the optimization techniques of the compilers are tested with various large real-world applications (e.g. audio/GSM codec suites), as well as industry benchmark standards such as Nullstone.

## Fast and compact

Altium understands that you expect your TriCore compiler to produce the most optimal code possible with no fuss. With its Viper compiler technology, the TASKING TriCore VX-toolset, in its default configuration, generates code with the smallest footprint and fastest execution possible. Depending on the specific requirements of your TriCore 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
- **Instruction Pipeline Scheduler** rearranges instructions to take advantage of parallel capabilities of the TriCore 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
- **Single Instruction, Multiple Data (SIMD)** optimizations to execute in one cycle

## TriCore architectural support

In addition to the data types outlined in the ISO C '99 standard and TriCore EABI, the TASKING TriCore VX-toolset compiler provides a number of DSP specific data types, such as *fract*, *sfract* and *accum*. The compiler supports these as true data types instead of typedef / intrinsic constructs and as such they offer many benefits in 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 TriCore's memory reach.

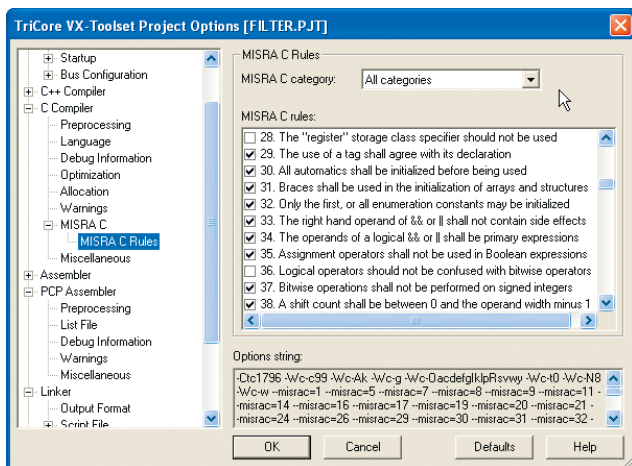
The TASKING TriCore VX-toolset offers a wealth of built-in intrinsic functions. Intrinsic functions appear as normal C functions, but the code generator interprets them and, if possible, generates more efficient code. Several pre-declared functions are available to generate inline assembly code at the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

## MISRA C

C is by far the most widely used higher level programming language to be found in embedded systems. Unfortunately, drawbacks in consistency and intrinsic quality make it debatable as to whether or not C is an appropriate language for safety-related applications.

Fully aware of this contradiction, the Motor Industry Software Reliability Association (MISRA®) published the "Guidelines for the use of the C language in vehicle based software" (known as 'MISRA C'), which describes programming rules for developers when using the C language in safety-related applications. The TASKING TriCore VX-toolset includes support for MISRA C through a configurable system of enhanced code checking, thus guiding programmers toward writing more robust, consistent C code.

The VX-toolset supports the original MISRA-C:1998 standard as well as the new MISRA-C:2004 rules set.



Fully-configurable MISRA C code checking

## 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 bits of TriCore assembly programming from within the C environment. The 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 scratch registers, which can be

optimized by the compiler. Enhanced debug information in the TASKING TriCore VX-toolset's object files enables source level debugging of inline functions and inline assembly.

## Industry standard libraries

The TASKING TriCore VX-toolset compiler 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 a number of highly practical variants.

Floating-point variants include:

- **Double and single precision** variants offering the choice between IEEE 754 compliant precision and increased speed
- **Trapping and non-trapping** variants offering the choice between run-time error checking and increased speed
- **FPU-optimized** variants for TriCore devices equipped with the Floating Point Unit

## TRICORE AND PCP MACRO-ASSEMBLERS

Although programming in C or C++ has the preference, hardware dependent I/O layers may still need to be coded in assembly in order to fully deploy the TriCore's rich feature set. For this purpose the TASKING TriCore VX-toolset incorporates powerful macro-assemblers for the TriCore main core and the PCP peripheral control processor.

TriCore/PCP Assembler features include:

- **TriCore 1/2, respectively PCP 1/2 compliant**
- **Full macro and conditional assemblers**
- **Branch/call instruction optimizations**
- **Extensive section directives**
- **Error file with textual error reporting**
- **TriCore EABI compliant ELF/Dwarf object output format with HLL debugging extensions**

## 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 TriCore VX-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.

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**
- **SREC and Intel HEX ROM image output formats**

## CROSSVIEW PRO DEBUGGER

In the world of embedded software development, there's no such thing as 'first time right'. Particularly with today's complex, high-performance applications, a fair share of development time will be spent in the debugger, validating your code.

The TASKING TriCore VX-toolset's CrossView Pro debugger is a perfect partner in 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. You choose the windows you need to view the different 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 time spent on testing and debugging.

Functionality includes:

- Simple through to advanced debugging features
- Intuitive source window
- Tracking scope and monitoring local variables
- Bubble-Spy™ for easy inspection of variables and functions

### 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 on your C++/C source, intermixed with the corresponding assembly code. To allow immediate access to your source files, you can jump directly from the CrossView Pro source window into the EDE editor at the exact source line.

### Multiple information windows

CrossView Pro offers a wealth of information windows allowing you to navigate through your application, 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.

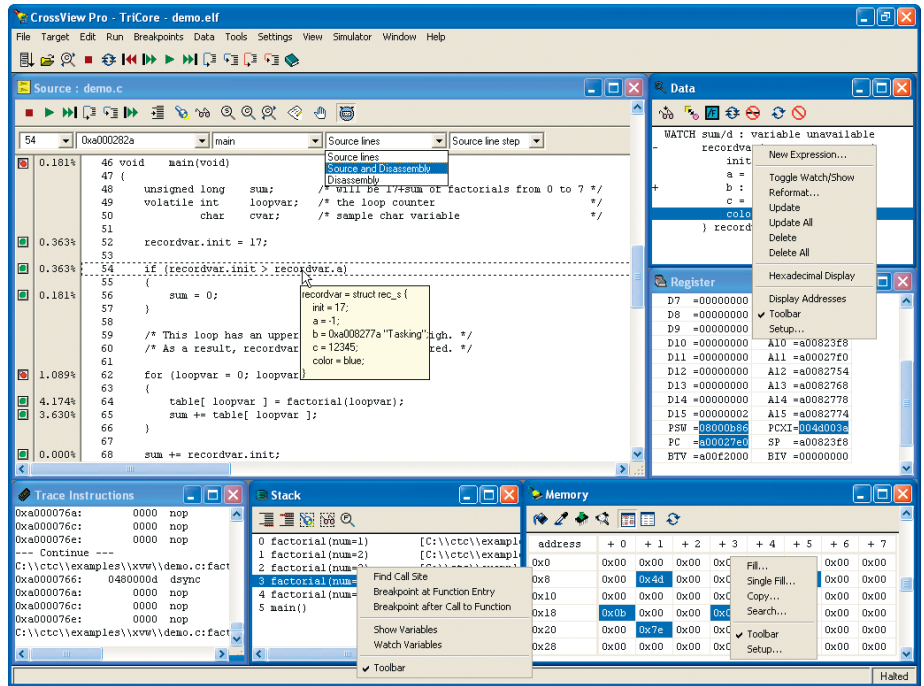
**Register windows** allow display and modification of CPU register values. Register windows are fully configurable to display any set of CPU registers. By defining multiple register windows you can easily organize your focus.

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

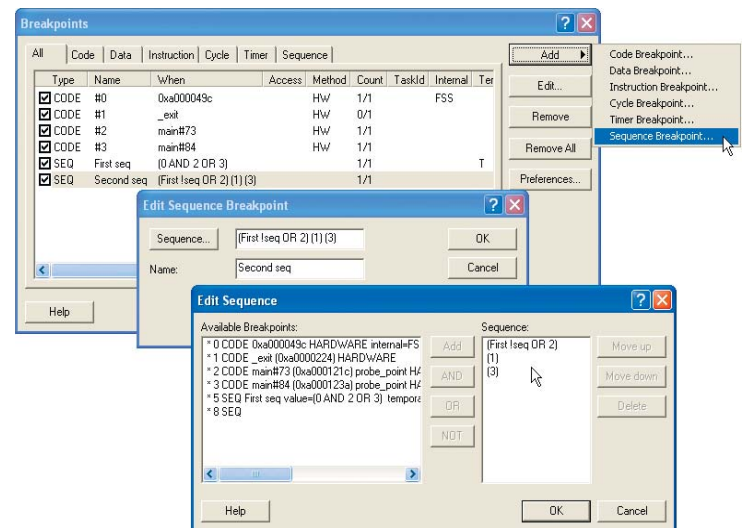
Breakpoints halt program execution and return control to the user. In addition to industry standard code and data breakpoints, you can configure your application to halt based upon instruction counts, cycle counts, or timer counts. All types of breakpoints can be defined as 'stop-and-go' probe points.



Spent less time debugging with CrossView Pro

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 'breakpoint-sequences'. This allows easy specification of the most complex conditions that need examining.



Full control over program execution through versatile breakpoint settings

### 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.

## Execution environments

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

### ■ TriCore Instruction Set Simulator

With CrossView Pro and the bundled TriCore 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 TriCore-1 and -2 instruction sets.

### ■ Target debugging through OCDS

Taking advantage of the TriCore On-Chip-Debug-Support (OCDS) facilities, CrossView Pro offers high quality in-circuit-emulation functionality at a low cost. Via the host PC's parallel or USB interface, CrossView Pro can communicate with your TriCore device directly.

The TriCore VX-toolset has been tested and qualified with Infineon's Debug Access Server (DAS) solution. The DAS environment is the universal emulation access software architecture for the XC8xx, XC166 and TriCore microcontrollers, and it excels in stability and reliability. Extensive support for DAS is guaranteed by Infineon and, as a result, Altium has adopted this new debug standard. Through DAS the TASKING CrossView Pro debugger is compatible with Infineon's USB-JTAG Wiggler box, enabling a very cost-effective debug solution for in-hardware testing (on-chip debugging).

## 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

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.

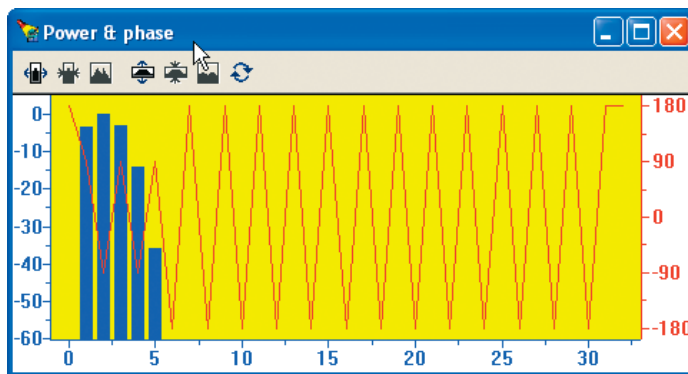
### ■ Profiling

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, but can also be presented in a report, 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 typically used in DSP applications. 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).

The four different analysis types included are 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 with CrossView Pro's Graphical Data Analysis

## Easy debugging RTOS-based applications

Altium'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. The KDI specification describes the open interface that can be used to add kernel-awareness to CrossView Pro for any commercial or proprietary RTOS. CrossView Pro RADMs are available for many popular commercial real-time operating systems including OSEK ORTI compliant RTOSes.

Object	Alarm Counter	Alarm Action	Time left	Alarm Task	Alarm Event
alarm1	system_timer	SETEVENT	0x00000017	bounce1	eventwrite
alarm2	system_timer	SETEVENT	0x00000000	bounce2	eventdelay
alarm3	system_timer	SETEVENT	0x00000000	bounce3	eventbounce
alarmrsi	counter_rsi	ACTIVATETASK	0x00000064	rsi	n.a.
alarmcb	system_timer	ALARMCALLBACK	0x00000000	n.a.	n.a.

Object	Event Mask
eventread	0x00000001
eventwrite	0x00000002
eventdelay	0x00000004
eventbounce	0x00000008

Object	Vector	Interrupt Number	Interrupt Category
HandlerISR0		0x00000004	CATEGORY_2
HandlerISR1		0x00000000	CATEGORY_2

Inspect the internals of your kernel in CrossView Pro

The RADM extends CrossView Pro with impressive kernel-aware debugging capabilities, such as:

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

## NEW REFERENCE DESIGN: TCP/IP STACK

More and more embedded devices are remotely managed – commanded, diagnosed, or upgraded – over the Internet or other networks. The included TCP/IP stack reference design enables you to add these networking functions to your end product. The stack is specifically designed for use in embedded systems and as a result it is fast, compact, efficient and reliable.

The TCP/IP stack is ready to run on Infineon's TC1130 TriBoard evaluation kit, using the on-board Ethernet connector and provided low-level drivers. It can easily be ported to any other TriCore-based application and allows a smooth incorporation into your embedded software.

The TCP/IP stack reference design comes with Altium's most extensive TriCore VX-toolset bundle and may be used royalty-free in your TriCore-based end product developed with this TASKING toolset.

### COOPERATION WITH THIRD PARTIES

Our extensive third party cooperation 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 and evaluation boards for the TriCore.

### 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 prolong this initial support period.

### PRODUCT PACKAGING & ORDERING CODES

Each TASKING product comes with full documentation in easy-to-use binders. This documentation is also available on-line as PDF files.

Product Code	Package contents
07-200-160-012	EDE, C/C++ Compiler and CrossView Pro Simulator debugger
07-200-160-024	EDE, C/C++ Compiler, CrossView Pro Simulator and OCDS debugger, and TCP/IP stack reference design
07-200-160-025	EDE, C/C++ Compiler, PCP C compiler, CrossView Pro Simulator and OCDS debugger, and TCP/IP stack reference design

A trial version of the TASKING TriCore VX-toolset is downloadable from our website at: [www.tasking.com/TriCore](http://www.tasking.com/TriCore)

## ALTIUM SALES OFFICES

### North America - Altium Inc

3207 Grey Hawk Court, Suite 100  
Carlsbad, CA 92010  
Ph: +1 760-231-0760  
Fax: +1 760-231-0761  
Email: [sales.na@altium.com](mailto:sales.na@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  
Email: [sales.cn@altium.com](mailto:sales.cn@altium.com)

### Australia – Altium Limited

3 Minna Close, Belrose  
NSW 2085  
Ph: +61 2 8622 8100  
Fax: +61 2 8622 8140  
Email: [sales.au@altium.com](mailto:sales.au@altium.com)

### Germany – Altium Europe GmbH

Albert-Nestler-Straße 7  
76131 Karlsruhe  
Ph: +49 (0) 721 8244 300  
Fax: +49 (0) 721 8244 320  
Email: [sales.de@altium.com](mailto:sales.de@altium.com)

### France – Protel AG

(A subsidiary of Altium Limited)  
121 rue d'Aguesseau  
92100 Boulogne-Billancourt  
Ph: 0800 88 05 06  
Fax: 0800 82 85 92  
Email: [info.fr@altium.com](mailto:info.fr@altium.com)