

HIGHLIGHTS

- Fully-integrated embedded development environment
- Language-sensitive editor
- Highly optimizing C/C++/EC++ compilers
- MISRA C enhanced code checking
- CrossView Pro debugger
 - Instruction set simulator
 - ROM monitor
 - On-Chip debug
 - (OSEK) Kernel-aware
- Extensive RTOS, ICE and evaluation board support
- DAVe 2 compatibility
- Available for
 - PC/Windows
 - SUN/Solaris

THE TASKING CLASSIC C166 SOFTWARE DEVELOPMENT TOOLSET

The C166 16-bit microcontroller family from Infineon Technologies offers a high-performance CPU with high peripheral functionality and enhanced I/O capabilities.

As a result of the close partnership with Infineon in the ongoing development of the architecture, this TASKING toolset is able to offer a state-of-the-art software development environment for the C166 microcontroller family, supporting all variants within the C166 architecture including the XC16x family and the licensable core designs C166S v1 and C166S v2. The XC16x derivatives and the C166S v2 are Infineon's latest extensions to the C166 architecture, offering many enhancements such as single cycle instruction execution, a five-stage pipeline and a Multiply Accumulate Coprocessor unit (MAC).

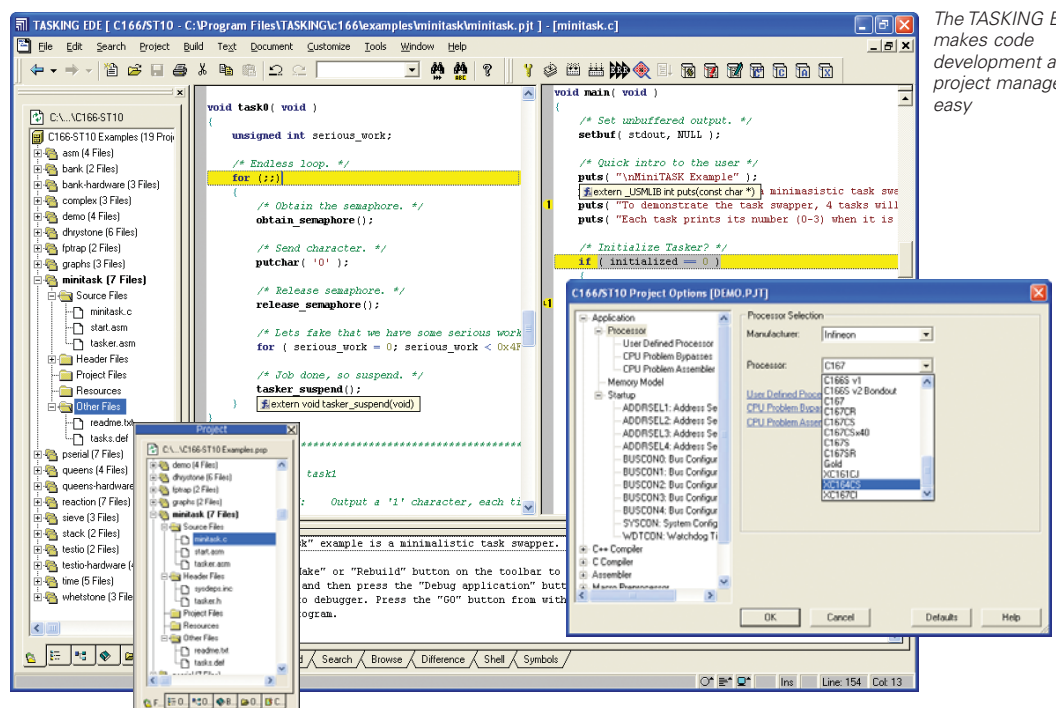
The toolset includes an ISO C++/EC++ compiler, ANSI C/MISRA C compiler, assembler with macro-preprocessor, linker/locator, libraries, and the CrossView Pro debugger – all integrated into the TASKING Embedded Development Environment (EDE) to provide a composite, user-friendly interface.

EMBEDDED DEVELOPMENT ENVIRONMENT

The Embedded Development Environment is an easy-to-use interface that integrates all the components of the classic C166 toolset, enabling you to edit, build and debug your embedded applications. The EDE is more than a language-sensitive editor; it is a complete environment that offers you direct access to the tools and features you need to maximize your productivity.

EDE provides:

- **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, 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 application's 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

C++/EC++ COMPILER

Fully aware of the undeniable trend towards higher level language programming, the TASKING C166 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 C166 family. The object-oriented benefits of C++ can be incorporated into an existing C application, one module at a time, providing a graceful migration from C to C++. Inheritance reduces the number of places where software behavior is defined and thereby speeds up development. The C++ compiler automatically includes a pre-link phase when templates are used. The C++ compiler comes with the Standard Template Library (STLport).

Scalable C++

Compatibility with the Embedded C++ (EC++) standard allows selective disabling of C++ features that may not be essential for your embedded application. By selecting full, or partial, compliance to the EC++ standard, code size overheads and run-time inefficiencies can be minimized.

C COMPILER

Based upon Altium's renowned expertise in the field of C compiler technology, the highly optimizing TASKING C166 compiler takes full advantage of the architecture and fully complies to the ANSI standard.

Features include:

- **Support for all members of the C166 architecture family**
- **Extensive optimizations for highly efficient code**
- **Five memory models to fit your application: tiny, small, medium, large, and huge**
- **Complete, ANSI-compliant C libraries**

Language extensions

In addition to full ANSI C compliance, the C compiler offers a wealth of specific language extensions for embedded C166 based applications.

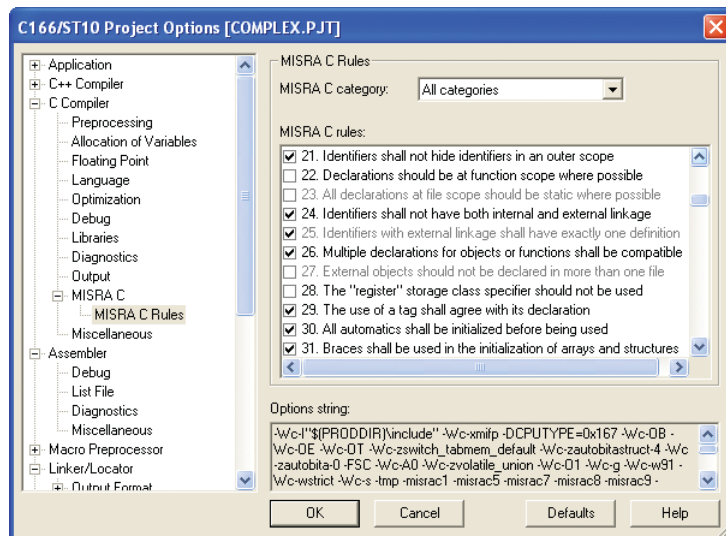
Language extensions include:

- **Additional data types `sfr`, `sfrbit`, `esfr`, `esfrbit`, `bit` and `bitword`**
- **The `_at()` function for easy allocation of variables at specific addresses**
- **A wealth of C166 specific intrinsic functions**
- **Easy C-level interrupt definition using `_interrupt`**
- **User-definable inline functions**

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 error checking, the use of error-prone C-constructs can be prevented.

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. The diagnostic messages from the compiler are configurable as either warning or error for both the MISRA C required rules as well as advisory rules. To ensure compliance with the MISRA C rules throughout



Fully configurable MISRA C code checking

the entire project, the C166 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 that were used to compile them.

Powerful optimizations

The C166 C compiler tools implement a wide variety of optimizations to allow reduction of code and data size as well as execution time. Optimizations can be applied on the complete project or specific files, or they can be switched on/off at function or source line level.

Optimizations include:

- **Various loop and jump optimizations to speed up execution and/or reduce code size**
- **Common sub-expression elimination detects and eliminates repeating (sub-) expressions**
- **Common tail merging for finding duplicate sequences of code and merging them together to reduce code size**
- **Dead assignment, dead storage and dead code elimination removes all kinds of unreachable code or invariant data**
- **Data flow analysis peephole optimizations replace instruction sequences with equivalent, but faster and/or shorter, sequences or delete obsolete instructions**
- **Global storage optimization pre-compiles the application to gather information about all static objects in order to determine the best memory configuration**

Floating-point libraries

Fast, reentrant floating-point libraries have been developed to offer state-of-the-art, high speed, high precision floating-point calculations. The floating-point libraries smoothly integrate within any of the supported real-time operating systems.

The ANSI/IEEE 754-1985 compliant software floating-point libraries support both single and double precision floating-point operations. To optimize for speed, both single and double precision libraries are available with and without run-time error trapping.

User inline assembly

The C166 compilers fully support inline assembly, allowing easy insertion of assembly routines into the C files. Altium's implementation of inline assembly leaves you in full control, providing access to variables defined at C level and identification of scratch registers – it's all possible.

ASSEMBLER

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

Features include:

- Full macro and conditional assembler
- Optimizing jmp/call instructions
- Extensive section directives
- Support of DPP (register) usage with extensive checking
- Error file with textual error reporting
- Full assembly source level debugging

Linker/Locator

The linker and locator are an essential part of the software building process that enable linking and locating of data and code into the target memory. The C166 locator will locate a linker file to absolute addresses. The ability to accurately describe the available memory and control the behavior of the locator is crucial for successful development of embedded applications.

Linker/Locator features:

- Automatic or user-defined allocation of code and data in memory
- Advanced overlaying features allowing efficient memory usage
- Incremental linking
- Smart linking to only include used functions
- IEEE695 object output format with HLL debugging information
- SREC / Intel HEX output format for (E)PROM programmers

FLASH MEMORY SUPPORT

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

CROSSVIEW PRO DEBUGGER

An easy-to-use interface with powerful and extensive debugging features helps you debug your applications faster. CrossView Pro provides multiple, resizable and independently controlled windows, which provide you with all the information required.

You choose the windows needed 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, adding functionality that reduces time spent on testing and debugging.

Functionality includes:

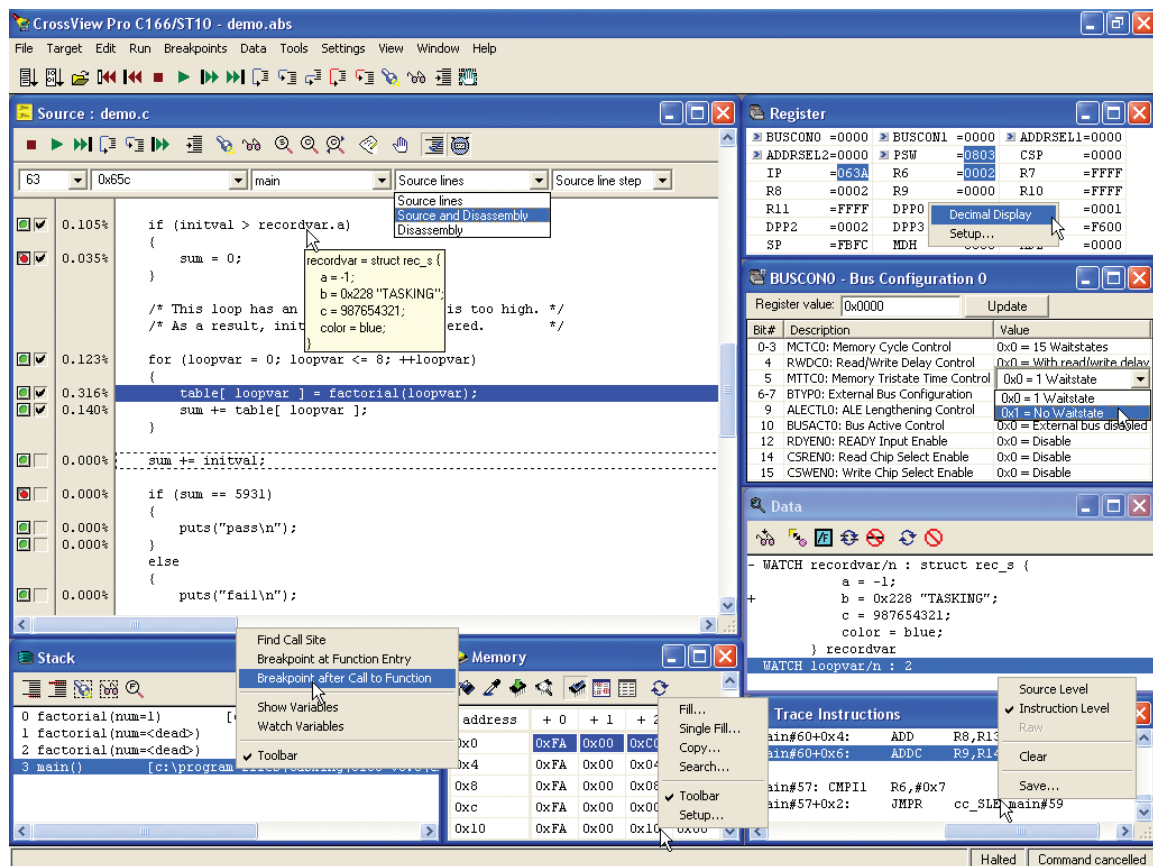
- Simple through to advanced debugging features
- Tracking scope and monitoring locals
- Intuitive navigation through the source window
- Double-click, right-click and tip-point functions
- Bubble-Spy™ for easy inspection of variables and functions

Source window

The source window is the main debugging window. It allows you to view source; step through your 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/C++ source, assembly or a mixed mode that allows a simultaneous view on your C/C++ source, intermixed with the corresponding assembly code. In order 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

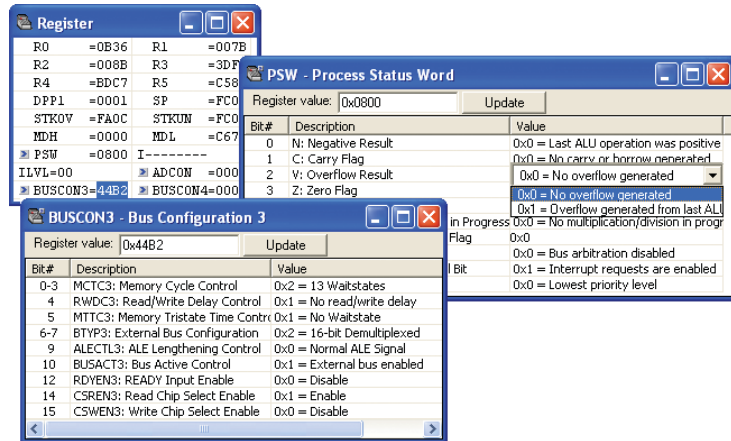
CrossView Pro offers a wealth of information windows allowing you to navigate through your application, monitor and modify data objects, CPU registers, the stack and memory locations.



Spent less time debugging with CrossView Pro

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. Symbolic Register Support shows and explains the bit fields that are present in a register, also allowing in-situ editing of the bit values.



Manipulation of bit field registers on symbolic level

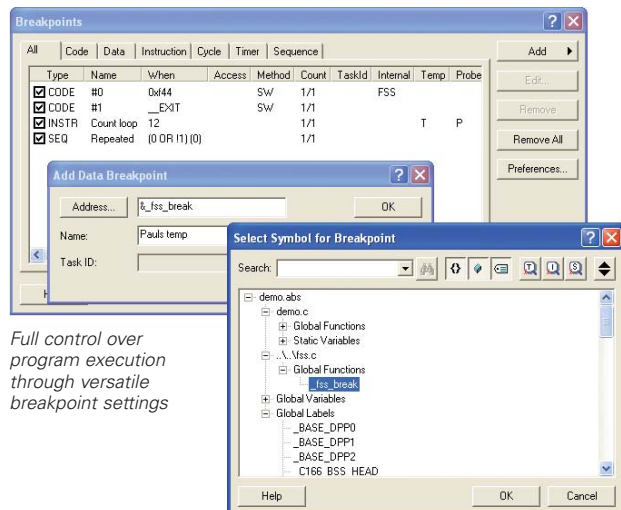
The **stack window** displays the contents of the function-call stack frame. You can easily configure stack-level breakpoints, navigate to the function-call's source and monitor local variables for selected functions.

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.



Full control over program execution through versatile breakpoint settings

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.

I/O simulation

File System Simulation (FSS) is one of CrossView Pro's I/O simulation techniques and allows the use of standard ANSI 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 FSS, you can read from and write to files on the host platform, or a CrossView Pro Virtual I/O window, directly. File System Simulation will work in any CrossView Pro target execution environment.

Multiple execution environments

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

■ C166 instruction set simulator debugging

The CrossView Pro C166 simulator debugger features instruction set simulation and simulation of a wide range of the microcontroller's on-chip peripherals, allowing you to extensively debug your application on the host platform, even before your target hardware is available.

The support peripherals include general purpose timers, interrupt system, peripheral event controller, parallel ports, SSC/ASC serial ports, phased locked loop, watchdog timer, real-time clock, A/D convertor and CAPCOM.

■ RAM/ROM monitor debugging

The CrossView Pro ROM monitor debugger can be used with any commercial off-the-shelf evaluation board or custom target board. CrossView Pro, running on the host computer system, debugs your application on the target board through the monitor application via a RS232 or CAN interface. All files required to build the monitor are shipped with the CrossView Pro ROM monitor, including documentation on how to retarget the monitor to your target board.

For PC/Windows a Remote Evaluation Board Access Server utility is available to allow debugging over a TCP/IP network on an evaluation board that is connected to a remote PC.

■ Target debugging through OCDS

Making the most of the On-Chip-Debug-Support (OCDS) facilities built into the latest Infineon Technologies C166 derivatives, CrossView Pro offers high quality in-circuit-emulation functionality at low cost. Via the host PC's parallel interface, CrossView Pro can communicate with the target device directly.

Easy debugging of 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 CrossView Pro capabilities to read, format and report kernel data structures for any commercial or proprietary RTOS. CrossView Pro RADMs are available for popular commercial real-time operating systems including OSEK-compliant RTOS's. Our generic RADM for OSEK kernels is based on the ORTI 2.0 and 2.1 language specification.

The RADM extends CrossView Pro 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**

Program performance analysis

CrossView Pro provides several 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 by providing timing information about a particular function, or set of functions. You can see how often a function is called and how much time is spent in each function. Profiling information can be shown in the left margin of the source window, but it is also presented in table format, providing you with the full overview.

■ Programmable data analysis

The programmable data analysis feature enables quick detection of gross errors in your signal processing routines by reducing large sets of data into meaningful visual

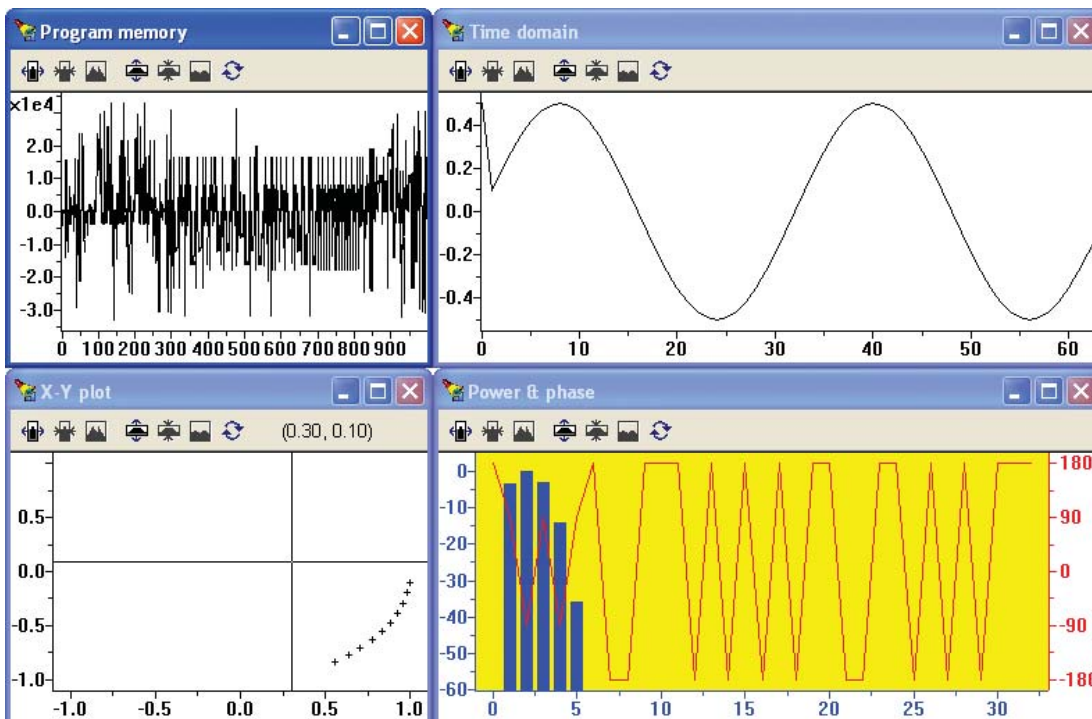
diagrams. The CrossView Pro debugger can analyze the data according to pre-defined or user-defined specifications, and display the data the way you need it. This eliminates the need for reviewing or post-processing large files of raw data. You can also view the same set of data in several ways at the same time (e.g. in the time and the frequency domains). Four different analysis types are available: x-t plotting, x-y plotting, FFT (Fast-Fourier-Transform) power spectrum, and Eye diagram. The C language script files for these pre-defined graphs can easily be used as the basis for custom data analysis windows.

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 Operating Systems, evaluation boards, CAN libraries, and CASE and UML tools for the C166 architecture family.

Compatible with Infineon's DAVe 2

Infineon Technologies' Digital virtual Application Engineer "DAvE 2" is the second generation expert system that can take away much of the hassle when starting a project based on the C166 architecture. DAVe can assist you in selecting the C166 derivative most suitable for your particular application. It also assists you with configuring and initializing the relevant peripherals within the selected device. Ultimately, DAVe generates a complete set of files that can be used as the basis of further developments. The files include configuration and initialization of the peripherals and function templates to make use of the selected peripherals. The TASKING C166 tool set is optimally tuned to inherit all settings and source files generated by DAVe. You can start your project with the assistance of DAVe, then import it into the TASKING C166 environment without redefining the same settings.



Analyze large sets of data through visualization in various representations

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, which entitles you to enhancements and improvements, as well as individual response to problems. Annual maintenance agreements are available to extend this initial support period.

AVAILABILITY

The Classic C166 software development toolset is available for PC/Windows and Sun/Solaris environments. For the latest information, contact your local sales office or reseller.

PRODUCT PACKAGING AND ORDERING CODES

Each TASKING product comes with full documentation in easy-to-use binders. This documentation is also available online in Windows Help format and PDF files.

Product code	Package contents
07-200-019-002	EDE, C compiler, assembler/linker, CrossView Pro simulator debugger
07-200-019-012	EDE, C/C++/EC++ compiler, assembler/linker, CrossView Pro simulator debugger
07-200-019-024	EDE, C/C++/EC++ compiler, assembler/linker, CrossView Pro OCDS, ROM monitor and simulator debugger, remote evaluation board access server

A trial version of the TASKING Classic C166 toolset can be downloaded from www.tasking.com/c166-classic

Developer's forum: www.yahogroups.com/group/TASKINGforum

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: +49 721 8244 300
Fax: +49 721 8244 320
sales.de@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
Ph: +81 3 6672 6155
Fax: +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