

IDE for Microcontrollers



Quick Start



Project Manager Editor & Debugger

µVision2, the *new* IDE from Keil Software, combines Project Management, Source Code Editing, and Program Debugging in one powerful environment. This Quick Start guide gives you the information necessary to use µVision2 for your own projects. It provides a step-by-step introduction of the most commonly used µVision2 features including:

- Project Setup for the Make and Build Process
- Editor facilities for Modifying and Correcting Source Code
- Program Debugging and Additional Test Utilities

The screenshot shows the Keil µVision2 IDE interface with several windows and components labeled:

- Project Window:** Located on the left, showing a list of registers (r0-r15) and their values.
- Peripheral Dialog:** A dialog box for configuring a Timer/Counter 1, showing settings like Mode, Input Prescaler, Status, and various registers (TI, TOICON, TITRE).
- Output Window:** Located at the bottom left, showing the execution log with source code lines and their addresses.
- Memory Window:** Located at the bottom right, showing memory addresses and their corresponding values.
- Watch Window and Call Stack:** Located in the middle right, showing the current state of variables (e.g., current, current.port, current.ar) and the call stack.
- Disassembly:** A window showing the assembly code corresponding to the source code.
- Serial #1:** A window showing the serial output, including commands and data.

Project Management

µVision2 now includes several robust features to make project management easier than ever.

File Groups let you group associated files together in a project. This is useful for grouping files into functional blocks or for identifying engineers in your software team.

Project Targets let you create several programs from a single project. You may need one target for testing and another target for a release version of your application. Each target allows individual tool settings within the same project file.

The **Project menu** provides access to all dialogs for project management including...

New Project... which creates a new project.

Targets, Groups, Files... which add components to a project. Local menus in the Project window let you add files to the project.

Open Project... which opens an existing project. Note that the Quick Start examples are based on the measure project files located in **EXAMPLES\MEASURE**.

Device Database

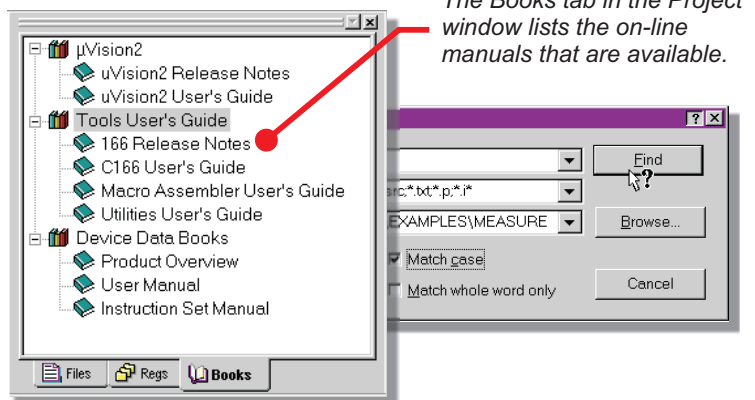
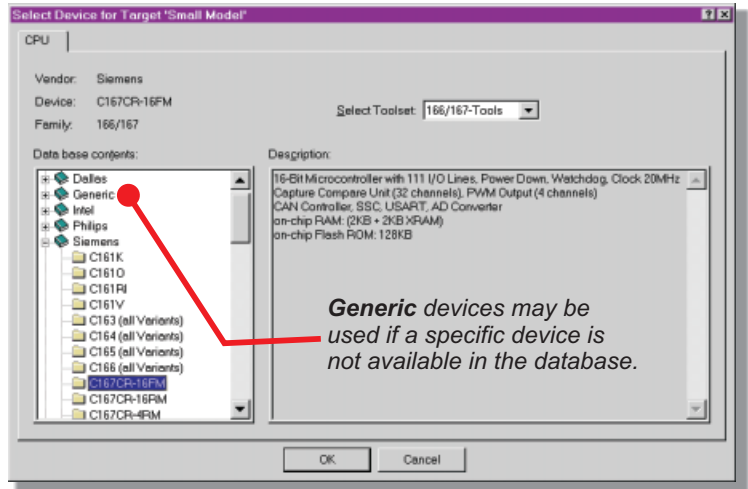
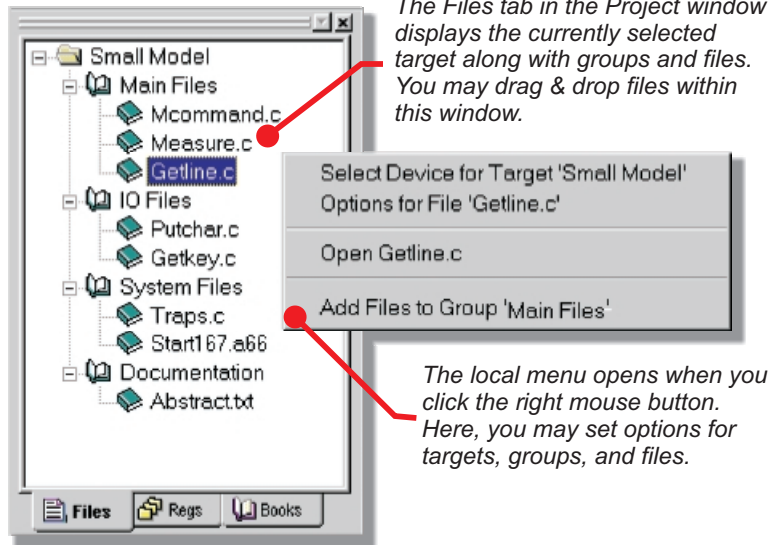
The Device Database makes it easy to start writing programs for a particular CPU. Just select the microcontroller you use and µVision2 sets the necessary options automatically. Since µVision2 already includes numerous devices in its database, tool setup is fast and easy. You may add new devices to the database as the need arises.

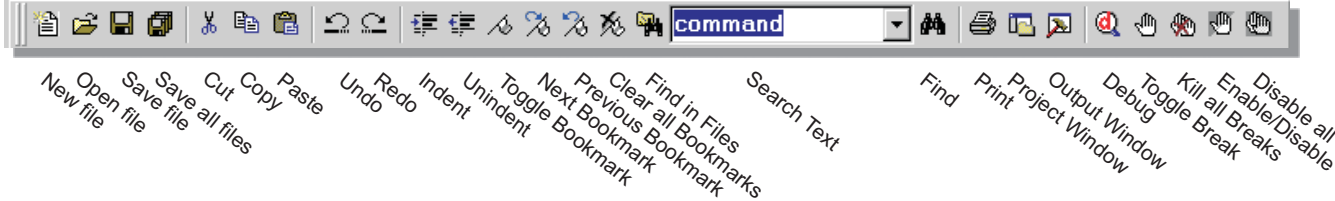
To choose a device for a project, point to the target name in the Project window and open the local menu. Then, use **Select Device**.

Getting Help ?

µVision2 provides a Books tab in the Project window where extensive on-line manuals for the toolchain and selected CPU are found. You may double-click on a book title to open the on-line manual.

Most dialogs have a *What's this* help button which provides detailed information about the dialog controls. To get help on menu items, select the item and press F1.

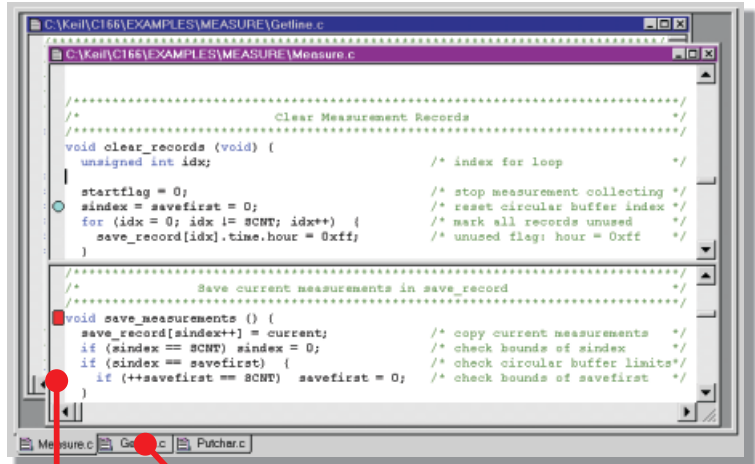




Source Code Editor

The μ Vision2 editor includes all the standard features you are accustomed to. Color syntax highlighting and text indentation are optimized for editing C source code. Most editor functions may be quickly accessed from the toolbar. The editor is available while debugging your program. This gives you a natural debugging environment that lets you quickly correct your source code.

Double-click in the Files page of the Project Window to open a selected source file. Then, use the Workbook navigation bar to quickly select editor windows. Enable **Workbook Mode** from the **View** menu.

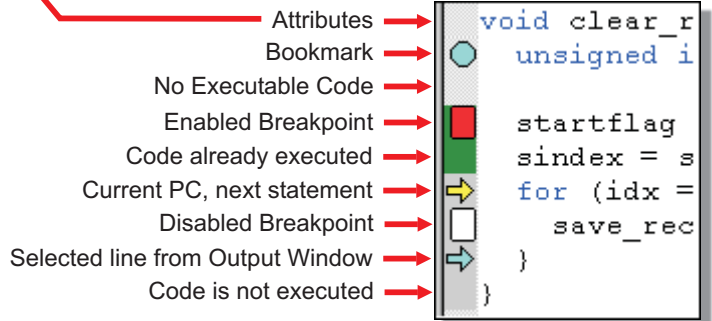


Workbook bar for quick document selection

Breakpoints

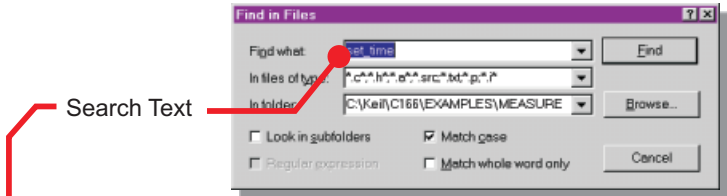
μ Vision2 lets you set program breakpoints while writing your source text. Simply use the buttons on the **Editor Toolbar** to mark breakpoints on source lines. After making your program, you can start the Debugger with the **Debug** toolbar button. Breakpoints you have set while editing are activated in your debugging session.

μ Vision2 marks the status of the source lines in the **Attributes** column of the editor window. This provides a quick overview of the current breakpoint settings.

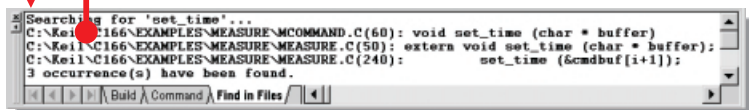


Utilities

In addition to its standard editor features, μ Vision2 contains many powerful functions that help you complete your projects on time. For example, the **Find in Files** dialog performs a text search in all specified files. The search results are displayed in the Find in Files page of the Output window. You can use this feature to locate all uses of a function or variable.

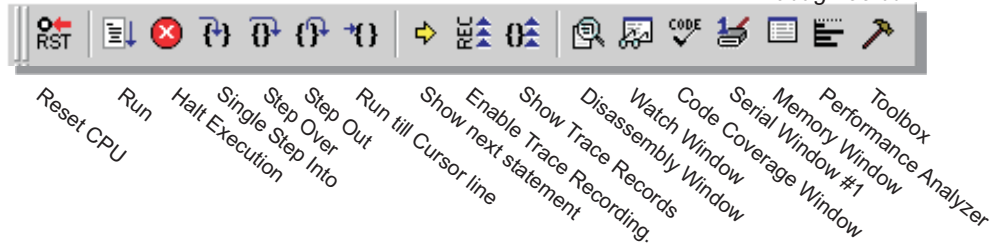


Double-click to position the cursor on the line with the matching string.



DEBUGGER

Debug Toolbar



Code Execution

You may use buttons on the toolbar to step through your application program. The **Run** button executes code until a breakpoint is reached. When **Trace Recording** is enabled, the **Show Trace Records** button lists the last 1024 instructions that were executed. Trace recording lets you analyze the program flow prior to a breakpoint.

Watch Window

The Locals page of the Watch Window displays the values of automatic variables in the current function. The value of a variable can be changed as follows:

- Select the value with a mouse click.
- Wait a second.
- Another mouse click enters edit mode.

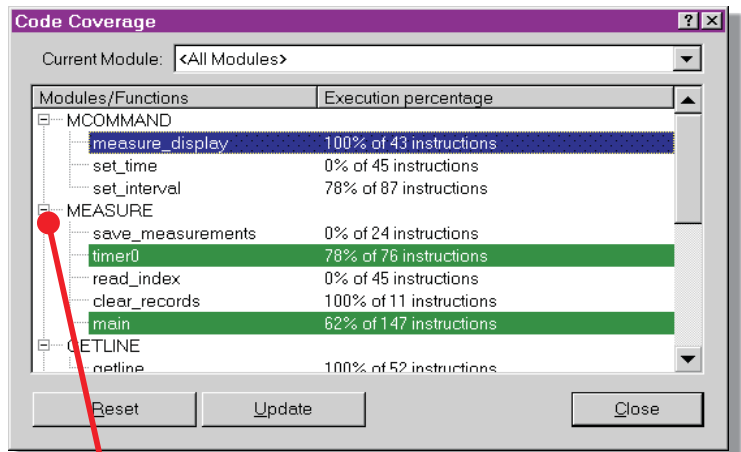
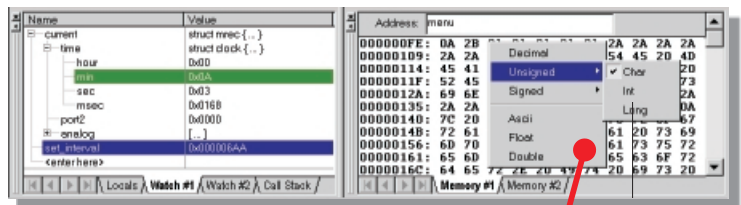
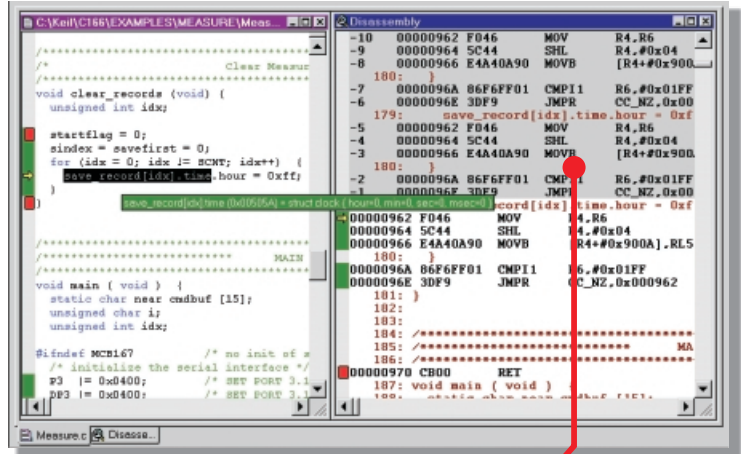
You can use this technique to edit variables in other Watch pages. The Call Stack page lists the program call tree; a double click shows the function invocation in the editor window.

Memory Window

The Memory Window displays two different memory areas. Double click to open an edit box to change the memory contents.

Code Coverage

The Code Coverage Window shows you the percentage of instructions in your program that have been executed. This aids in debugging and testing. It lets you easily distinguish the parts of your program that have been executed from the parts that have not. You may then change your testing strategy to ensure that all program paths are traveled and tested.



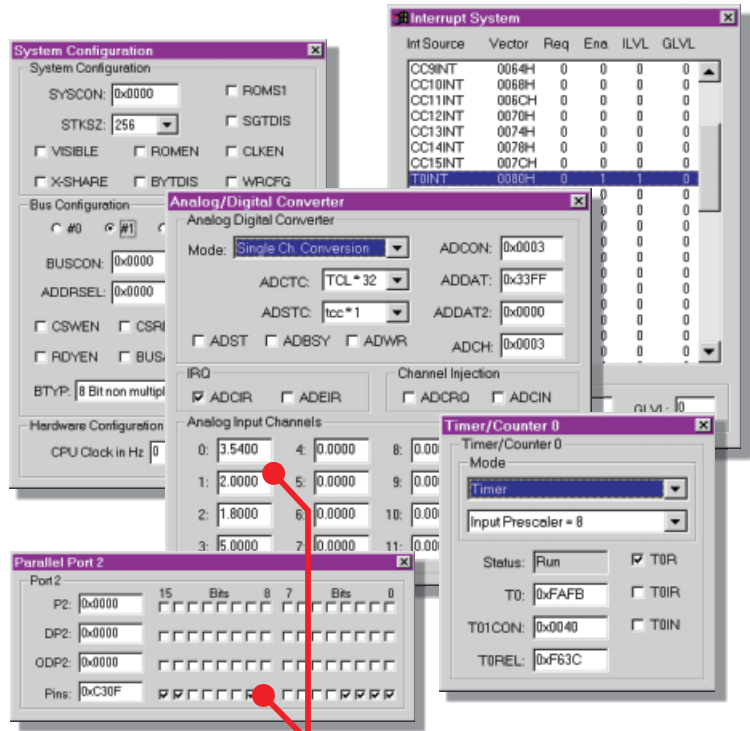
Code coverage displays statistical information and helps you locate un-executed program sections.

Peripheral Simulation

μ Vision2 simulates the on-chip peripherals of numerous microcontrollers. When you select a CPU from the device database to configure your project, μ Vision2 automatically configures the peripheral simulator for you. With its logical and timing simulation, it is possible to test an application before the target hardware is even available. The simulator makes it easy to test hardware defects and critical situations which are difficult to debug with real hardware.

*Peripheral dialog boxes, available with commands in the **Debug** menu, let you review and modify the status of on-chip components including system and BUS configuration. All interrupts, including timers, are simulated.*

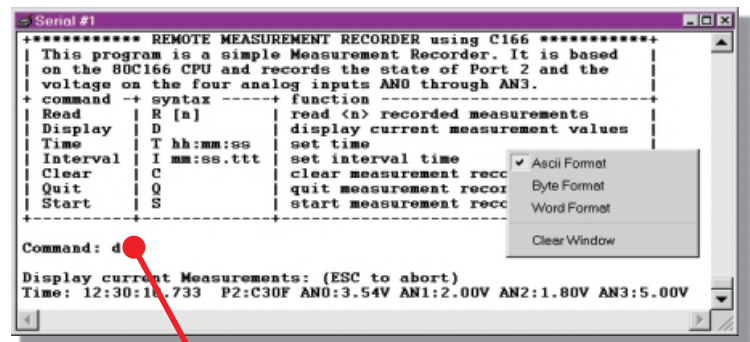
*You may enter I/O port values in the **Parallel Port** dialog pages. If an I/O pin is configured as an interrupt input, the interrupt service routine is serviced. In the **Analog/Digital Converter** dialog, you may enter the input voltage for the analog inputs of the CPU. μ Vision2 simulates the conversion time and the interrupt of the on-chip A/D converter.*



You may supply I/O pin values in the peripheral dialog boxes.

Serial Window

The Serial Window provides a terminal simulator for the on-chip UART. HEX and ASCII modes may be selected via the local menu. For devices with two serial interfaces a second Serial Window is available. The Serial Windows are convenient to use for printf test outputs in your application programs.

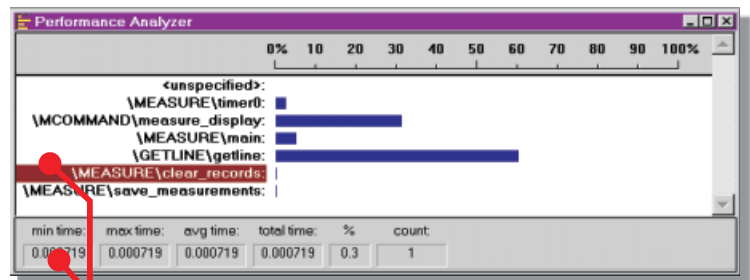


Select the Serial Window and enter ASCII characters for UART input.

Performance Analyzer

The built-in performance analyzer records and displays execution times for functions and program blocks you select. A bar graph displays the amount of CPU time spent in each part of your program.

You can use the information gathered by the performance analyzer to determine where your program spends most of its time. Then, you can concentrate your efforts on making that section faster.



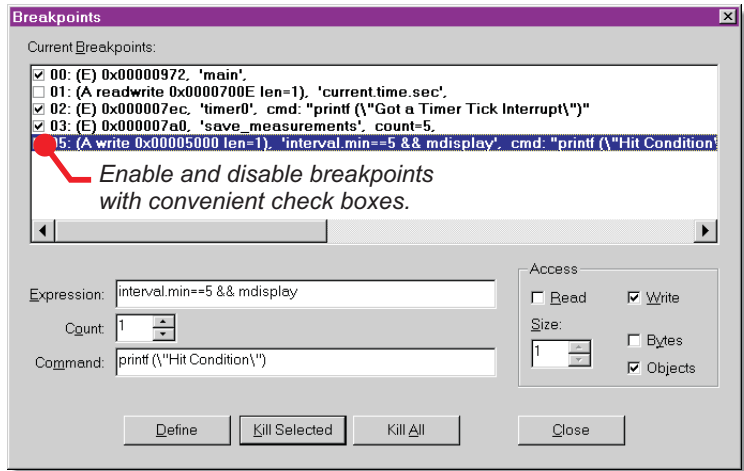
Select a function or block for detailed statistics. Use the **Debug - Performance Analyzer** menu for configuration.

Complex Breakpoints

µVision2 allows breakpoints on C statements, assembly instructions, conditional expressions, and even on different types of memory accesses. Breakpoints may include a count which is decreased until the breakpoint is triggered. A command may be executed when the breakpoint is triggered. If no command is specified, program execution stops.

You may use the **Breakpoint** command from the **Debug** menu to set breakpoint definitions.

The command string may invoke debug functions when a breakpoint is triggered.



Function Language

µVision2 incorporates a C-like function language that lets you generate and use:

- Built-in functions like printf, memset, rand, and other useful routines.
- Signal functions to simulate analog and digital inputs to the CPU.
- User functions to extend the command scope and combine repetitive actions.

You may use the function language to describe the behavior of external hardware components or to supply input values to the controller I/O pins. Signal functions run in the background while µVision2 simulates your target program. Therefore, signal functions must call the **twatch** function to let your target program run.

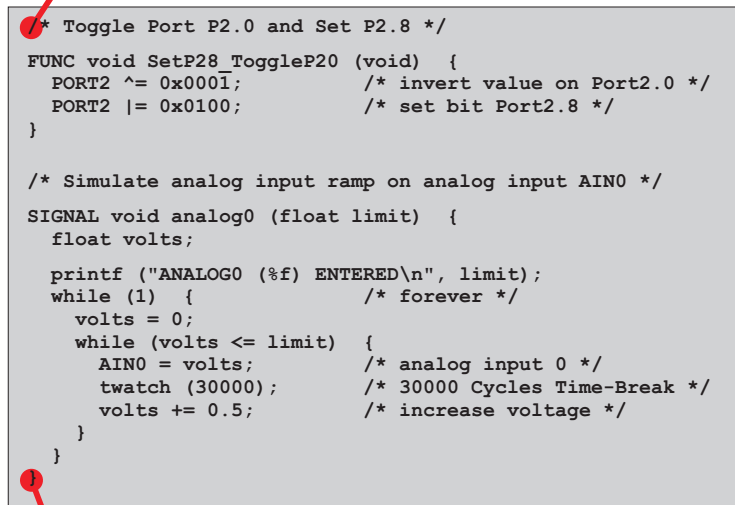
Toolbox

The **Toolbox** dialog contains user-configurable buttons. Click on a Toolbox button to execute the associated command. Toolbox buttons may be executed at any time, even while running the test program.

Target Monitor

The µVision2 debugger communicates with your target hardware and the Keil Monitor program to help you easily debug your applications in real-time. You can single-step through program code, check variables, set breakpoints, examine memory contents, and more! The Keil Monitor comes pre-installed on many evaluation boards.

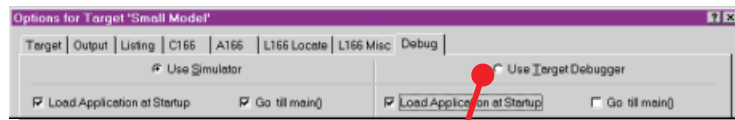
Create Debug Functions with the **Debug - Function Editor** dialog.



Using toolbox buttons, functions can be called at any time, even during program execution.



Toolbox buttons are defined in the Command page of the output window.



Select **Use Target Debugger** to connect to a Monitor board.

SHORTCUTS

The following table lists important shortcut keys for μ Vision2:

Key	Function
Ctrl+N	Create new file
Ctrl+O	Open an existing file
Ctrl+P	Print current document
Ctrl+S	Save current document
F5	Go (execute) until the next active breakpoint
Ctrl+F5	Start/stop debug session
F7	Build project
Ctrl+F7	Compile current file
Alt+F7	Open project options dialog box
F10	Single-step over a function
Ctrl+F10	Go (execute) to the current cursor line
F11	Single-step into a function
Ctrl+F11	Step out of the current function
EDITOR KEYS	
Ctrl+A	Select all text
Ctrl+C	Copy selected text to the clipboard
Ctrl+F	Find text
Ctrl+H	Replace text
Ctrl+V	Paste text from the clipboard
Ctrl+X	Cut selected text to the clipboard
Ctrl+Y	Cut line to the clipboard
Ctrl+Z	Undo last operation
Ctrl+[Find matching brace
Shift+Ctrl+Z	Redo last Undo command
Ctrl+Home	Move the cursor to beginning of file
Ctrl+End	Move the cursor to end of file
Ctrl+←	Move cursor one word left
Ctrl+→	Move cursor one word right
Ctrl+F2	Set bookmark on current line
F2	Move cursor to next bookmark
Shift+F2	Move cursor to previous bookmark
F3	Repeat search text forward
Shift+F3	Repeat search text backward
Ctrl+F3	Search for word under cursor

Example Programs

Example programs are found in the **EXAMPLES** folder. Each program is stored in a separate folder along with a project file that will help you quickly build and evaluate the example.

Many of the example project files contain several targets with different configurations. Refer to the document **ABSTRACT.TXT** for a detailed description.

Example	Description
BADCODE	Program with syntax errors and warnings. You may use the editor to correct these.
CSAMPLE	Simple calculator which shows how to build a multi-module project with μ Vision2.
DHRY	Dhrystone benchmark. Calculates the dhrystones factor for the target CPU.
HELLO	Hello world program. Try this first when you begin using μ Vision2. It prints <i>Hello World</i> on the serial interface and helps you confirm that the development tools work correctly.
MEASURE	Data acquisition system for analog and digital signals.
RTX_EX1	RTX51 example. Demonstrates round-robin multitasking with the RTX real-time kernel.
RTX_EX2	RTX51 application that uses signals for task communication.
SIEVE	Benchmark that calculates prime numbers.
TRAFFIC	Traffic light controller using the RTX real-

Contact your local distributor for more information about our other embedded systems development tools and products.

Keil Elektronik GmbH and Keil Software, Inc., have representatives in the following countries:

Australia, Austria, Belgium, Brazil, Canada, Czech, China, Denmark, Finland, France, Germany, Hong Kong, India, Ireland, Israel, Italy, Japan, Korea, Mexico, Netherlands, Norway, Poland, Russia, Spain, Singapore, South Africa, Sweden, Switzerland, Taiwan, United Kingdom, and United States.

Contact Keil Elektronik GmbH or Keil Software, Inc., for the distributor nearest you.

Evaluation software and detailed technical information are available from your distributor and our World Wide Web site (www.keil.com).

Europe:

Keil Elektronik GmbH
Bretonischer Ring 15
85630 Grasbrunn
Germany

Phone ++49 89 / 45 60 40 - 0
Support ++49 89 / 45 60 40 - 24
FAX ++49 89 / 46 81 62
Email sales.intl@keil.com
support.intl@keil.com

United States:

Keil Software, Inc.
1501 10th Street, Suite 110
Plano, TX 75074
USA

Phone 800-348-8051
972-312-1107
FAX 972-312-1159
Email sales.us@keil.com
support.us@keil.com



<http://www.keil.com>