Using the Keil 8051 Evaluation Tools with the SMC97C100 USB Evaluation Board

February 7, 1998
Revision 57

Abstract: This application note describes how to use the Keil evaluation tools for the 8051 with SMC's 97C100 USB Evaluation Board.

What You Need

There are only a few things you need to get started with the SMC97C100 Evaluation Board.

- SMC97C100 Evaluation Board,
- Keil Software 8051 Evaluation Kit,
- Keil Monitor HEX File,
- Keil Software Example Programs.

You must obtain the SMC Evaluation Board from SMC. The Evaluation Kit, Monitor, and Example Programs from Keil are all available on the world wide web.

Keil Software 8051 Evaluation Kit

You may download the Keil 8051 Evaluation Kit from the following URL:

http://www.keil.com/demo/ek51.htm

You may also contact Keil Software directly at 800-348-8051 and request a copy of the Keil Development Tools CD-ROM.

Keil Monitor HEX File

The SMC97C100 Evaluation Board requires that the Keil monitor is programmed into the FLASH memory. The Monitor is a program that communicates with your PC via the serial port. It starts running when the SMC97C100 starts. Your evaluation board comes with a FLASH device that has this monitor pre-programmed. However, you may download it from the following URL:

http://www.keil.com/smc/usb/smc_moni.zip

Keil Software Example Programs

There are several example programs you may use to quickly get started with the SMC97C100. They all may be found at the following URL:

http://www.keil.com/smc/usb

This application note discusses the COUNT, PRNTTEST, and SEVENSEG example programs which are found there.

Getting Started

You build and test the example programs discussed in the following sections in exactly the same fashion. After connecting and applying power to the SMC97C100 Evaluation Board, use the Keil µVision IDE to compile and link one of the example programs. Then, use the Keil dScope debugger to download the program to the evaluation board and step through the program code.

Connecting the Evaluation Board

You must properly connect and configure the SMC97C100 Evaluation Board first. This is pretty easy to do and is outlined below.

Serial Cable

Connect the serial cable between COM2 (P5) on the SMC97C100 Evaluation Board and a COM port on your PC. Make sure that your PC's COM port is not shared with an internal modem and also that no other software makes assumptions about another device that may be connected to this port. The cable must be a null-modem or "cross cable". Pins 2 and 3 must be cross-connected. 5 is ground.

JP7 and JP8 Jumpers

The JP& and JP8 jumpers must be set so that the 8051 UART is used.

JP22 GPIO Resource Jumpers

All GPIO resource jumpers must be set. Especially High, Low, T1, T0, TXD, and RXD.

Power Supply

Finally, apply power to the board. At Keil, we found that a standard PC power supply works just fine. We are using a 120 Watt switching supply because of the low current requirements of the board.

Using the µVision Integrated Development Environment

µVision is an integrated software development platform that includes a full-function editor, project manager, make facility, and environment control for the Keil 8051 development tools.

Start µVision from the Keil PK51-Eval Program Group (Windows 3.1*x*) or from the Start Menu (Windows 95 and Windows NT).

Using the dScope Simulator/Debugger

dScope is a powerful simulator and target debugger. It provides you with a way to test and verify your programs before you have target hardware (using the simulator capability). It also provides you with a way to run your application programs on target hardware (using the target debugging capability).

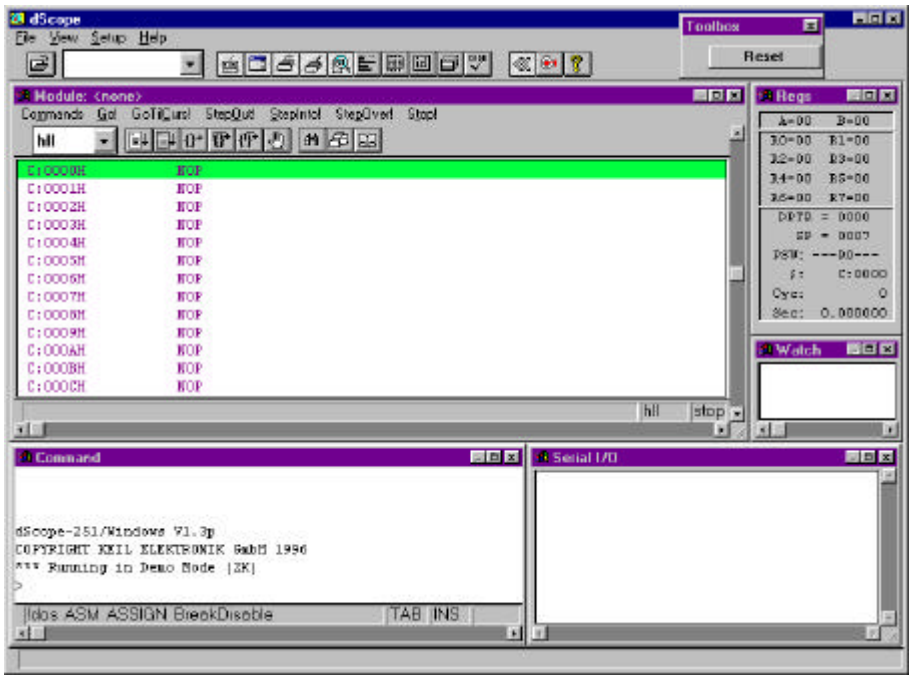
dScope's simulation capability lets you completely simulate all aspects of a target CPU. This includes the instruction set as well as most on-chip peripherals. Dialog boxes let you observe and dynamically change the on-chip peripherals.

dScope's target debugging capability interfaces to your target hardware using the serial port on your PC and a serial port on the target. The Keil MON51 monitor program runs on your target and communicates with dScope to let you download programs, read memory, set breakpoints, and so on. The primary requirement from your target hardware is that it support some type of von Neuman memory architecture. The SMC97C100 Evaluation Board does that starting at address 0xC000 (XDATA and CODE are mapped to the same SRAM device). Note that when target debugging is used, dScope's program name in the Title Bar changes to tScope to signify target debugging is used.

You select whether you are simulating or target debugging by the CPU driver DLL you select after dScope starts. See below for more information about which driver to use.

There are a number of ways you can start the dScope Debugger. In µVision, you may click on the Debug Button on the Toolbar. You may select the dScope program from the Keil PK51 Eval program group. You may even have µVision start dScope automatically after a successful project build.

When dScope starts, a screen similar to the following displays.



The buttons on the Toolbar display and hide the different dScope windows. The primary windows you will use are the Debug Window, the Command Window, the Registers Window, the Watch Window, and the Serial Window.

Loading the CPU Driver DLL

dScope uses CPU driver DLLs to determine the on-chip peripheral set to simulate (for simulation) and to determine the communication protocol to use (for target debugging).

You must select the MON51.DLL driver to communicate with the SMC97C100 Evaluation Board. You may do this from the File menu or simply by selecting the driver from the drop-down combination box on the Tool Bar.

When you select the MON51.DLL CPU driver, dScope immediately attempts to establish communications with the monitor running on your target hardware. So, before you start the CPU driver, make sure power is applied to the evaluation board.

It is always good idea to reset the evaluation board using the Reset button immediately before you select the MON51 CPU driver. This ensures that the monitor program is the only application that is running and that a previously loaded program does not cause any side effects.

If a communication problem occurs and the target cannot be found, dScope displays a dialog box informing you.

You may change the default COM port and baud rate. The correct baud rate is 9600 for MON51 on the SMC97C100 Evaluation Board. If you continue to have communication problems, point your web browser to http://www.keil.com/support and search the Support Solutions Database for help with dScope and Serial Port Problems.

Once the MON51 CPU driver loads and successfully communicates with the target board, the Command Window displays the monitor driver's version number.

When this happens, dScope successfully communicated with the monitor on the evaluation board and everything is ready to start debugging.

Configuring the CPU Driver DLL

After you have successfully loaded the MON51 CPU driver, you may need to make configuration changes. Select the Configuration command from the Peripherals menu to open the Configuration dialog box.

From the Configuration Dialog Box, you may change the Port Settings, Baud rate, Cache Settings, and Target Configuration.

The only configuration option you may need to change is the Target Configuration option, "User serial interrupt". This option selects whether the serial port used by the monitor on the target board is interrupt-driven or polled. This is a very important option that may affect how some of the example programs work.

If "Use serial interrupt" is selected, you may halt program execution at any time using the Stop! command in the Debug Window. However, you may not use the **printf** function or other routines like **puts** and **putchar** to output text through the serial port to the Serial Window.

If "Use serial interrupt" is not selected, you may not halt program execution using the Stop! command. The only way you program stops is if it hits a breakpoint or if you reset the target board using the reset button. When this option is not selected, you may use **printf**, **puts**, and **putchar** to output text through the serial port to the dScope Serial Window.

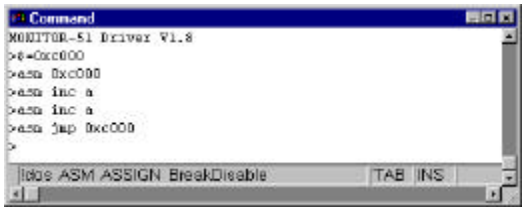
Testing the Monitor Interface

It is probably a good idea to test the interface between dScope and your target using a simple program before you launch into a large-scale program. Fortunately, this is easy to do from within dScope.

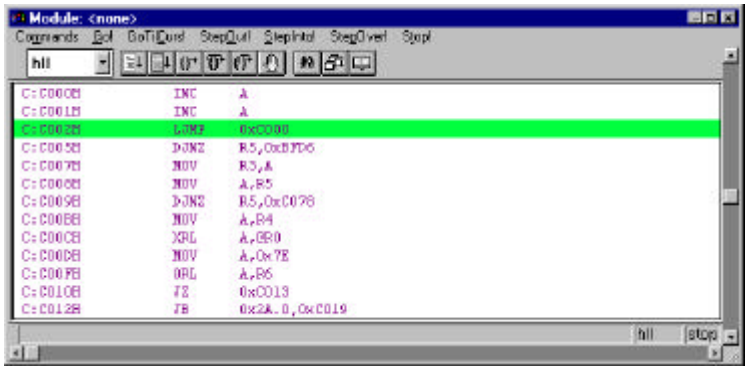
Use the mouse to select the Command Window and enter the following lines (without the comments).

```
$=0xC000       /* Set the program counter to 0xC000 (start of von Neuman memory) */
asm 0xC000     /* Set dScope's assembler to start at address 0xC000 */
asm inc a      /* Assemble an Increment Accumulator instruction */
asm inc a      /* Another one */
asm jmp 0xC000 /* Assemble a Jump instruction */
```

These instructions appear in the Command Window as follows:



You may single-step through your program using the StepOver! command in the Debug Window. As you step through the program, the next line to be executed is highlighted as shown below.



Take a look at the Register Window and note that the value of the accumulator (A) changes each time you step over the INC A instruction. The Parity flag in the PSW changes every other time the accumulator increments.

If you can make this example program work, you have successfully connected the evaluation board to your PC and you have correctly configured dScope to communicate with it.

Using the Example Programs

Three simple example programs provided on the Keil Software web site (**http://www.keil.com/smc/usb**) can help you get started with the SMC97C100 Evaluation Board. Store them on your hard disk in a directory of your choosing.

- COUNT is a simple program that counts from 0 to 99 repeatedly.
- PRNTTEST is a program that prints a message out the serial port.
- SEVENSEG writes values from 0x00 to 0xFF on the 2-digit display on the evaluation board.

These programs are described in the following pages.

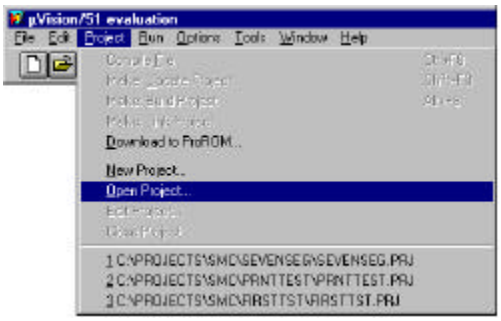
1) COUNT

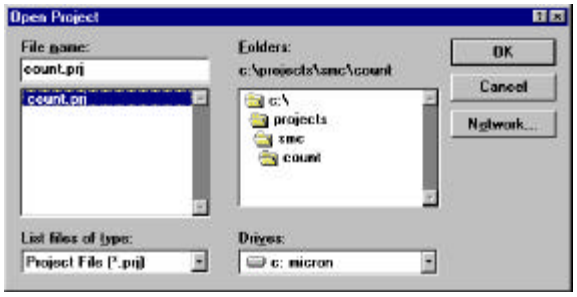
The COUNT program is a very simple loop that increments a variable. The purpose of this example is to demonstrate how to single-step and watch variables as they change. It is not a sophisticated program!

The purpose of such a simple program is to validate the steps of opening a project, building the project, starting dScope, loading the MON51 driver, downloading the program to the evaluation board, and stepping through the code.

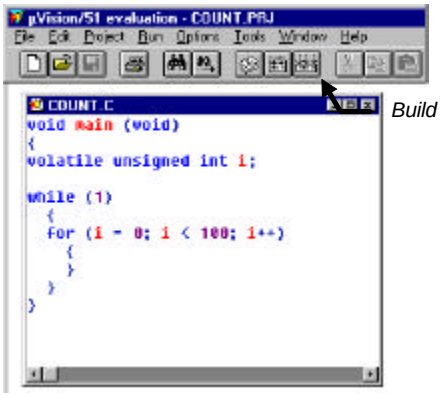
Opening the Project

To load and build the COUNT program, start the µVision-51 IDE from the Windows Program Group or from the Start Menu and select Open Project from the Project Menu. When the Open Project dialog box displays, select the directory where you installed the COUNT program and then select the COUNT.PRJ file.





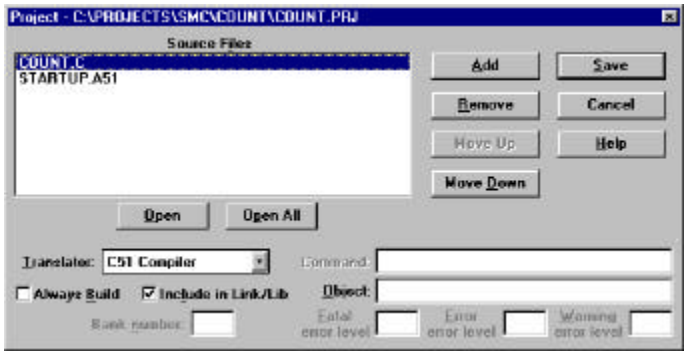
When you open the COUNT project, µVision displays the COUNT.C file. The count program really does nothing more than count from 0 to 99 over and over and over.



Build All Button

Editing the Project

A project includes all the source files and assembler, compiler, and linker options necessary to create a program. To display and edit the list of files in the project select the Edit Project command from the Project Menu. This displays the Project Manager dialog box.



The only source files in the COUNT program are COUNT.C (the C program that counts from 0 to 99) and STARTUP.A51 (the startup code modified with a starting address of 0xC000). Refer to "Creating Your Own Programs" later in this application note for more information about the startup code.

When you are finished viewing the project file list, select Cancel to exit without saving any changes.

Editing the Source Code

Add these two lines in the source code: **`unsigned int a;`** & **`a=a+2;`**

```
void main (void)
{
volatile unsigned int i;
unsigned int a;

while (1)
  {
  for (i = 0; i < 100; i++)
    {
    a=a+2;
    }
  }
}
```

Building the Project

To build the COUNT project, click on the Build All button on the Tool Bar. This compiles and assembles all files in the project. While µVision compiles and assembles the COUNT program, the status of the build is displayed.

When the COUNT program has been successfully compiled and linked, µVision informs you that the Make was successful and waits for you to select OK. Once the program is compiled and linked, you are ready to start debugging using dScope and the SMC97C100 Evaluation Board.

Starting dScope and Loading the MON51 Driver

To start dScope from µVision, click on the Debug Button on the Toolbar. When dScope starts, you must load the MON51 CPU driver using either the File menu or the Drop-Down Combination Box on the Tool Bar. If you have trouble - try pressing the reset (S1) and trying again.

If "No Target Found" dialog box appears, check the COM port, baud rate (9600) and the cable. You may need to turn off the COM port FIFO buffer in Windows 95.

At the Command window prompt: type d and the contents of some memory area will be returned. This indicates that communication is successful.

Downloading COUNT to the Evaluation Board

From the File menu, select Load object file. The filename is count (with no extension) and its location will already been passed to dScope by µVision. Select count and press OK.

At the Command prompt: enter $=0xc000. The program source will be visible in the debug window.

Double-click on the line a=a+2 to set a breakpoint. [BR0] indicates breakpoint zero.

Open the Memory window from the View menu and if necessary, enter d d:0x0 to show the data memory. The variable I is at 08-09 and variable a resides at 0xa-0xb. Each is two bytes long.

Stepping Through the Code

Press Go! And confirm that these variables increment by the expected 1 and 2 respectively. Each time you press Go!, this small program is run and halts at the breakpoint [BR0] and the memory window is updated.

Close dScope before going on to the prnttest example.

2) PRNTTEST

The PRNTTEST program prints a string to the 8051's serial port using the **printf** library routine. This program demonstrates how the monitor and your target can share the 8051's serial port. Since communications between the monitor and dScope follows a communications protocol, output from your application program can easily be separated out and displayed in the Serial Window.

The instructions that follow the same format as the previous one (COUNT).

Opening the Project

Make sure dScope was closed. There is a prntest.ini file with some preloaded Commands that will be activated the next time dScope is started. Open µVision The project name you want to select is prnttest.prj. Open the Project menu and select Open Project. Select the file prntest.prj in the usual manner.

Building the Project

The source file ptest.c should open. If not, open the file menu and select it under Open. Everything is preconfigured in the .prj file. Click on the Build All icon. Press OK when the project has been linked.

Starting dScope and Loading the MON51 Driver

Start dScope in the fashion appropriate for your version of Windows. Note that some Commands were automatically executed in the Command window. Use the up arrow key to select the history feature to view these commands.

The use Serial interrupt option must be disabled. Open the Peripherals menu and select Configuration. Make sure this option is not checked.

Downloading PRNTTEST to the Evaluation Board

The file prntest.ini loaded the proper file into memory and set the program counter to 0xc000.

We want to set a breakpoint at 0xc40d which is the printf statement of the source example. Scroll down to this memory location and double-click on the printf statement to set breakpoint [0].

Stepping Through the Code

Make sure the Serial window is open and visible.

Press Go! Repeatedly and every time the printf statement is executed, a string will be printed to the Serial I/O window. Remember the Serial interrupt option must be disabled for this to work properly.

Enable the Serial interrupt option and close dScope.

3) SEVENSEG

The SEVENSEG program counts from 0x01 to 0xFF and displays the count on the 2-digit display on the SMC97C100 Evaluation Board. This program shows you how to use one of the ISA bus features of this evaluation board.

Opening the Project

The project name is sevenseg.prj. Select this file as in the other examples. It is preconfigured for the various compiler options. It and the source file sevenseg.c need no editing.

Building the Project

Click on the Build All icon to compile the files. Press OK when successful linking has occurred. Start dScope. Note that once again, Commands are automatically loaded.

Stepping Through the Code

Scroll down to line 144 - display (I) and double-click on it to set breakpoint [0]. Note the LED on the SMC board is set to 00. Press Go! Repeatedly and note the LED increments by one each time.

You can set the value of the LEDs by pressing Change 7-seg on the Toolbox window. Note this button was created by a single line in the Command window.

Creating Your Own Programs

When you create your own programs for the SMC97C100 Evaluation Board, remember that your program must be relocated to offset 0xC000 when you debug using the Keil monitor and dScope. When you program a FLASH device, your program must be located at offset 0x0000.

Therefore, you must perform the following operations when you create a program.

1. Set the starting address for the program in the STARTUP.A51 startup code.
2. Set the starting address for CODE memory in the Linker Options.
3. Set the interrupt vector offset within the CODE segment in the Compiler Options.
4. Use the **$=** command in dScope to set the program counter to the beginning of your application.

Setting the Starting Address in STARTUP.A51

The startup code found in STARTUP.A51 includes code that begins at the 8051's reset vector. This address is normally fixed at 0x0000. However, when you debug using a monitor, this vector must be relocated.

The easiest way to "modify" this file is to copy it from the directory of one of the example programs (since they have already been modified).

If you want to do this yourself, copy the STARTUP.A51 file from the \C51\LIB directory into your project directory. Then, change the line that reads:

```
                CSEG    AT      0000h
```

to read

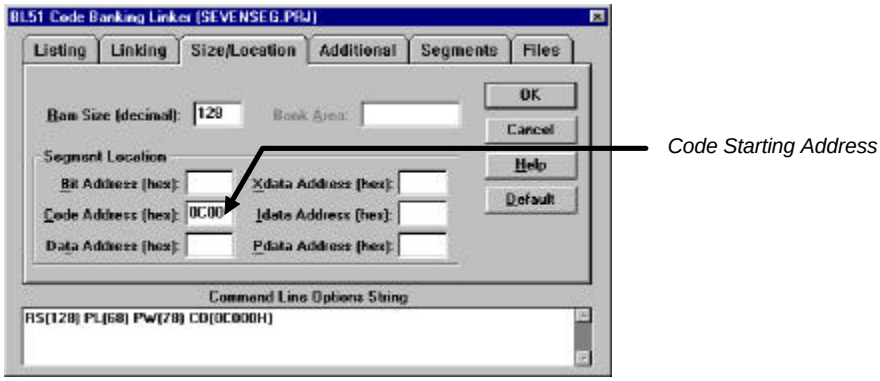
```
                CSEG    AT      0C000h
```

A partial listing of the STARTUP.A51 file is shown below.

```
.
.
.
                EXTRN CODE (?C_START)
                PUBLIC  ?C_STARTUP

                CSEG    AT      0C000h          ; Setup for SMC97C100 Eval Board
?C_STARTUP:     LJMP    STARTUP1

                RSEG    ?C_C51STARTUP

STARTUP1:
.
.
.
```

Setting the CODE Starting Address in the Linker Options

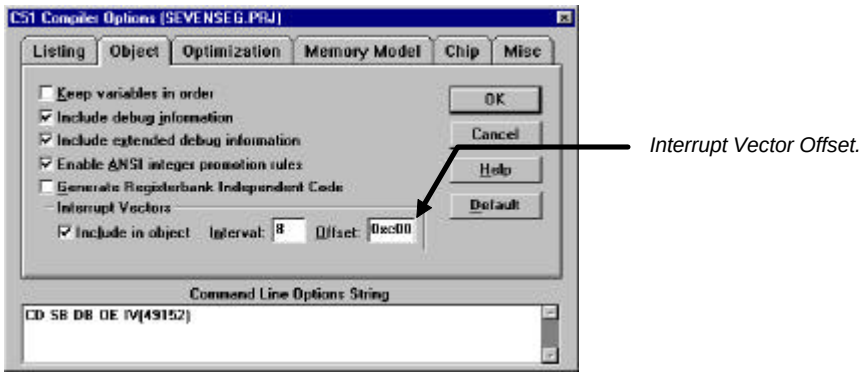
Open the Linker Options Dialog Box by selecting BL51 Code Banking Linker from the Options Menu. When the dialog box displays, select the Size/Location Tab.



In the Code Address input line, enter "0C000" for the starting address for code space. This tells the linker to start locating all relocatable CODE at address 0xC000 and up.

Setting the Interrupt Vector Offset in the Compiler Options

Open the Compiler Options Dialog Box by selecting C51 Compiler from the Options Menu. When the dialog box displays, select the Object Tab.



In the Offset input line, enter "0xC000" for the interrupt vector offset. This tells the compiler to locate interrupt vectors starting at 0xC000 for any interrupt functions you use.

Setting the Program Counter in dScope

When you load a program in dScope, the Program Counter is set by default to 0x0000. If this is where your program starts (which is normal for 8051 programs), then you must do nothing to begin debugging.

If you create a program to download to the 97C100 for debugging with the monitor, your program must start at 0xC000. In the Command Window, enter the following:

```
$=0xC000
```

This command sets the Program Counter to 0xC000 and displays the startup code for your program in the Debug Window. You are now ready to begin debugging your program.

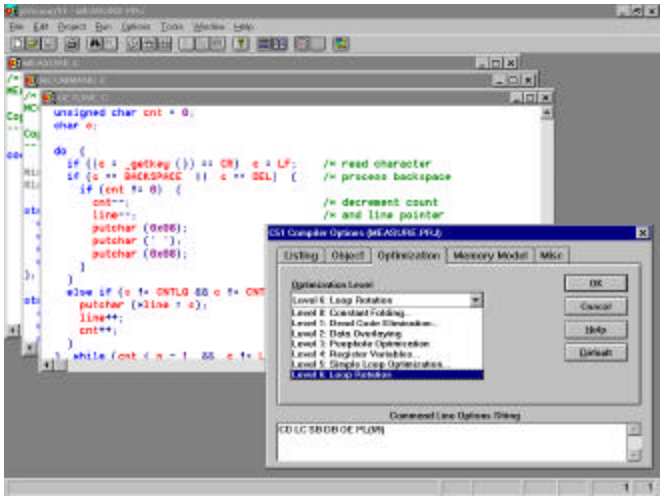
Conclusion

Look at the source code provided with each example to see how easy it is to program with the Keil Software tool chain. Keil Software provides quality C Compilers for all 8051 derivatives. See our Web site www.keil.com and www.keil.com/~market for more information.

Keil Software

C51 C Compiler for the 8051 *Version 5.20*

- The C51 is the leading 8051 C Compiler. It produces very fast and small code using advanced optimizing algorithms.
- True ANSI C Compilers provide support for *all USB 8051 derivatives.*
- Superior code optimization and global register coloring generate high-performance applications
- Arguments and return values are passed in registers
- Generic 3-byte pointers and code-efficient memory-specific pointers are supported
- Support provided for reentrant functions
- Interrupt service routines may be written in C
- Compiler package includes the BL51 code banking linker/locator
- Debugger can simulate external hardware events
- The SMC 97C100 Evaluation Board is fully supported



Product Information

C51: The Keil C51 compiler is a dedicated ANSI C Compiler designed explicitly for the 8051 family of microcontrollers. The C51 compiler generates code that is extremely fast and very compact. C language extensions give you full access to the resources of the 8051. With over 15,000 installations world-wide, the Keil C51 is the most widely used compiler for the 8051 family. The C51 supports 8051 based USB parts such as the SMC USB97C100.

The professional developer's kit is a complete software development environment that users can expand with third party emulators, fuzzy-logic tools, and any Intel-compatible utility. The dScope source-level debugger provides fast and reliable program debugging. dScope lets you simulate an entire 8051 system including on-chip peripherals, and external hardware and signals.

Constant improvements have been made to the C51 compiler since its market introduction. Version 5.20 improves long and float arithmetic, comes with a DOS extender, and supports three warning levels and extensive syntax checking. Version 5.20 also provides support for all 8051 derivatives with multiple data pointers.

Technical Support

Keil provides free technical support for one full year via a toll free number from Dallas, Texas. Support is also available on the Web site (www.keil.com/support) and via Email (support@keil.com).

Keil Software Product Overview

Keil Software develops, manufactures, and distributes embedded software development tools for the 8051 microcontroller family. Tools include C Compilers, Assemblers, Real-time Executives, Debuggers and Simulators, Integrated Environments, and Evaluation Boards. Keil has a Technical Training facility in Dallas, Texas for 8051 products. Training can be provided at your site. Keil provides distribution, product development, and technical support from its office in Dallas, Texas. Keil provides contemporary and relevant application notes and sample C code.

Keil's web site [www.keil.com] provides the latest information about our development tools, demo programs, software updates, application notes, example programs, and links to other sources of information of value to embedded developers. See also www.keil.com/~market for additional information.

Keil development tools come in many configurations to suit your needs. The Product Matrix below is a brief description of these kits. *xx*51 refers to the entire 8051 Family. The Kit components listed in this chart are described on the other side of this page. For additional pricing call Keil Software at (800) 348-8051.

Technical Support is included via an 800 number for the first year and is based in Dallas, Texas. The operating manuals are written in Texas as is the Windows IDE µVision.

The RTX51,is a Multitasking Real-time Operating System. These powerful tools let you manage multiple tasks on a single CPU. There are two versions of each - FULL and Tiny. The TINY is a subset of the Full version. CAN libraries are included with FULL version. No royalty charges are generated.

Product matrix

Component / **Product**	Macro Assembler	C Compiler	µVision IDE	dScope Simulator	RTOS Tiny
A51	A51		✔		
CA51	A51	C51	✔		
DK51	A51	C51	✔	✔	
PK51	A51	C51	✔	✔	✔

Updates & Upgrades

Free updates for Keil compilers are available from the Web to registered users of Keil products. Special upgrade prices are available for selected competitive compilers. Special prices are available to qualified college students. Call Keil Software on how to save money using these programs.

C51 C Compiler for the entire 8051 Family

The C51 is a full ANSI C compiler with assembler. µVision is the Windows User Interface that drives the optimizing C51 compiler. The Compiler, Assembler and Linker options are point and click. µVision and the manuals are written in the USA and extensive on-line help is included. Keil compilers produce the smallest and most efficient code in the industry. Free technical support from Dallas via a 1-800 line is included for one year. The C51, as well as all other Keil compilers, assemblers and linker/locators, can also be operated in DOS mode with your own make file if you prefer.

A51: Macro Assemblers for all Keil supported microcontrollers

This Macro Assembler is included with the Compiler package or is available separately. All utilities needed to complete your project are included. This Assembler is DOS based or can be run from µVision which is included with every Assembler and Compiler package from Keil.

µVision Integrated Development Environment - Windows based User Interface

µVision is a USA developed Windows-based front end for all Keil Compilers and Assemblers. It includes an editor, project manager, and make facility. Compiler, assembler, and linker options are set by pointing and clicking on prompted selections. The Application Manager conveniently accesses your files, and 3rd party executables and also calls the Keil Simulator dScope easing the transition from application to application. Works with Windows 3.11, 95 and NT.

dScope-Debugger and Simulator for the 8051. Windows or DOS based.

dScope is a source-level debugger that lets you debug programs created by Keil compilers. dScope simulates your program either in stand-alone mode or in your target using the monitor. External hardware, signals, and interrupts can be simulated. Viewable windows include Program Source, I/O space, Trace, Stack, Watch and CPU registers plus more.

RTX51 Full-Function RTOS for the entire 8051 family

The RTX51 is a multitasking Real-time Operating System for the entire 8051 family. This powerful tool lets you manage multiple tasks on a single CPU. The RTX51 Full includes CAN libraries. The RTX51 Tiny is a subset of the RTX51 Full. Functions include interrupt and memory management, clock, and semaphores. Royalty-free. A Tiny version is available.

Contact Information

United States
Keil Software, Inc.
16990 Dallas Pkwy., Ste. 120
Dallas, Texas 75248-1903
USA

Phone: 800-348-8051
Phone: 972-735-8052
Fax: 972-735-8055

Email: sales_us@keil.com

Website: www.keil.com

www.keil.com/~market

Europe
Keil Elektronik GmbH
Bretonischer Ring 15
D-85630 Grasbrunn
Germany

Phone: ++49 89 456040 0
Fax: ++49 89 468162

Email: sales_intl@keil.com