C Compiler • Real-Time OS • Simulator • Training • Evaluation Boards

Using the Keil Monitor-51 / µVision2 Debugger with the Analog Devices ADuC812 Evaluation Board

Application Note 151

May 17, 2000, Munich, Germany

by Keil Support, Keil Elektronik GmbH support.intl@keil.com ++49 89 456040-0

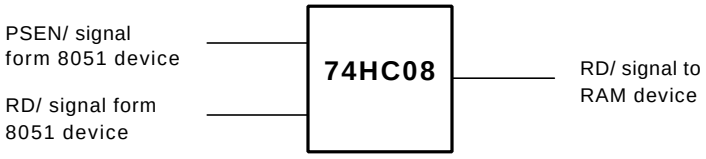
This Application Note describes the steps that are required to install and use the Keil Monitor-51 together with the Analog Devices ADuC812 Evaluation Board, Rev. B01

Hardware Modification – Von-Neumann wired code/xdata Memory

A requirement for the Keil Monitor-51 is von-Neumann wired **code** and **xdata** memory. The Analog Devices evaluation board comes without this hardware feature. Therefore you need to modify the hardware to match this requirement.

Von-Neumann means that you can read a physically the same memory bytes from code and xdata space. This is necessary to download software into **code** space since the 8051 does not provide CPU instructions to write into code memory. Typically a AND gate is used to combine the RD/ and PSEN/ signals of the CPU and generate a RD/ signal for the RAM device.

For the ADuC812 board this means that you have to add an 74HC08 (or similar device) on the prototype area and cut the RD/ signal to Pin 1 of the RAM (U8) device. The signals to the RAM device need to be wired as shown in the figure on the right.



The figure below shows you the signals on the board that you need to cut and wire to the AND gate.

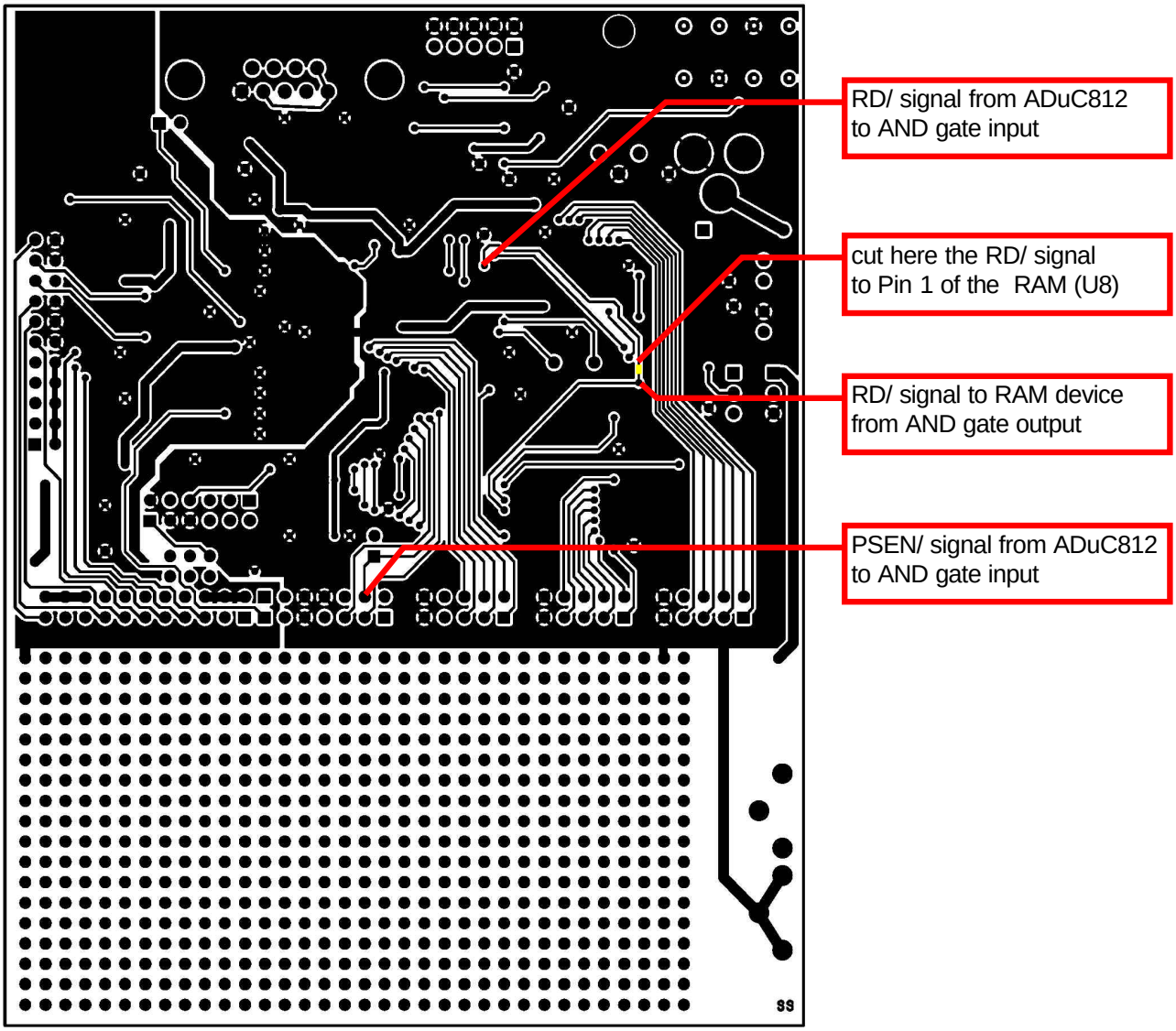


Figure: Modifications and Signals on the Analog Devices ADuC812 Circuit Board

Download the Keil Monitor-51 to the on-chip code Flash ROM

This application note comes with a pre-configured Monitor that can be directly downloaded to the ADuC812 Evaluation Board using the AD Download utility. This download utility is located on the Analog Devices QuickStart Development System CD ROM in the folder DOWNLOAD.

The pre-configured Monitor is located in **\ADUC812\MONITOR\MON51.HEX**. This monitor was generated with the batch file INSTALL.BAT with the following parameters:

INSTALL 8 7F

This creates a Monitor program that uses the Timer 2 as baudrate generator with auto-adjust baudrate. The Monitor code area starts at default address 0 and the monitor data area starts at xdata address 0x7F00.

NOTE
Due to a chip problem in the ADuC812 silicon it is required to use the monitor version that comes with this application note. The standard Keil Monitor-51 included with the Keil 8051 toolchain will not work on the ADuC812 device.

The following steps are required to download the Keil Monitor-51 on the ADuC812 board:

- close jumper LK3 on the board
- insert jumper LK4 to position A (pin 1)
- insert jumper LK6 to position A (pin 1)
- connect the board via a 9-PIN serial cable to the COM1 interface of your PC
- connect power supply to the board (LED PWR is on)
- press the RESET button on the board
- open a **MSDOS command prompt window** on your PC
- enter **G:\DOWNLOAD\DOWNLOAD \ADUC812\MONITOR\MON51.HEX**
  G: is the path to your CD ROM drive
  \ADUC812 is the path to the application note folder

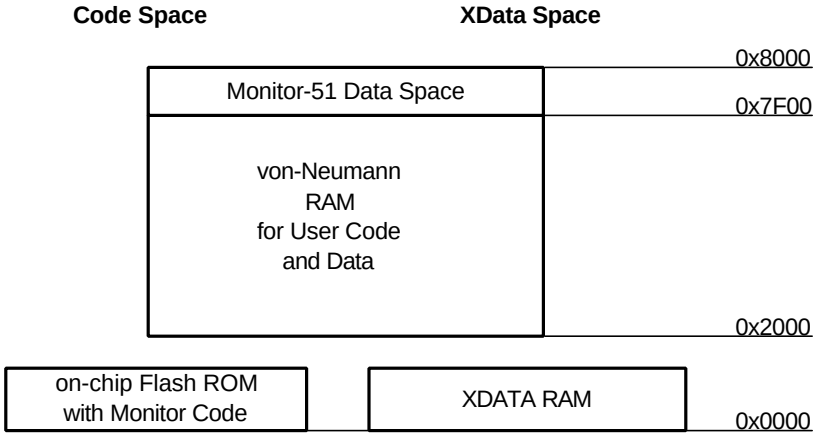
This should give the following output:

```
C:\aduc812\monitor>G:\DOWNLOAD\DOWNLOAD \ADUC812\MONITOR\MON51.HEX
DOWNLOAD.EXE: ADuC program downloader ver  2.05
6th Sept, 1999  C.King Accutron Ltd.
Initialising Com1 at 9600 baud:.....OK!
Resetting the target device: ADuC812 reset OK!
Target firmware revision 2.01
Erasing code and data memory . . OK
Downloading C:\ADUC812\MONITOR\MON51.HEX: ..............
.....................................................
.....Downloaded OK!
```

Now remove jumper LK3 and press the RESET button to start the Keil Monitor-51.

Memory Mapping on the ADuC812 Board

After this modifications your Analog Devices ADuC812 board has the following memory layout.

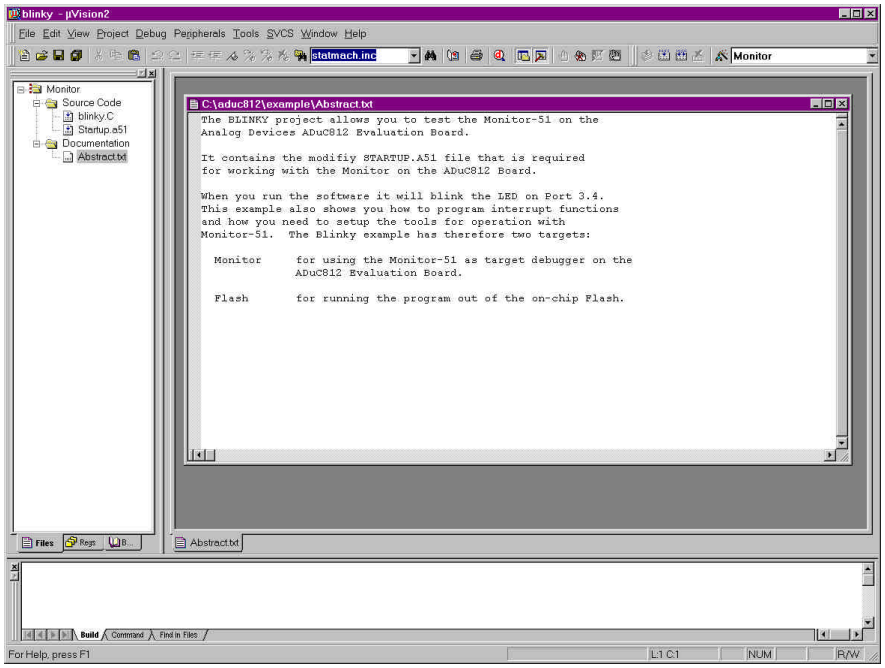


This gives you for your application program 8KB xdata RAM from 0x0000 - 0x1FFF and about 24KB program code space from 0x2000 - 0x7EFF. However you may use portions of the von-Neumann RAM also as data space.

Using µVision2

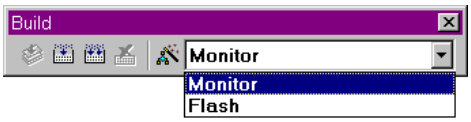
To provide a quick start with the µVision2 IDE this application note comes with a small example program that is located in the **`EXAMPLE`** folder. This example is called **`BLINKY`** and flashes the P3.4 LED on the board. It covers also an interrupt service routine and contains a configured version of the **`STARTUP.A51`** file.

After installation of the µVision2 tool chain you can start the IDE and load the project file with Project – Open Project: **`\ADUC812\EXAMPLES\BLINKY.UV2`**. This gives you the following screen output.



µVision2 Screen after loading the BLINKY project

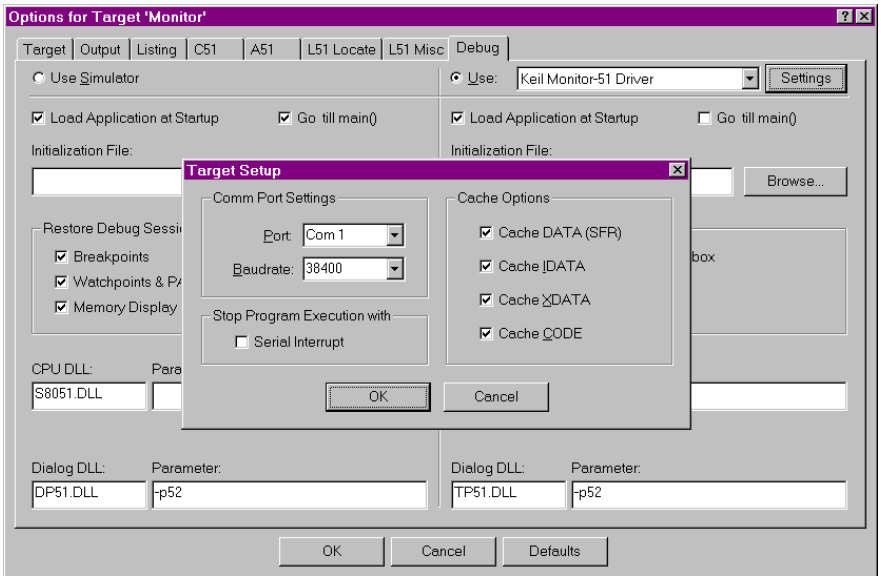
Then you can start using the Keil 8051 tool chain. The following shows you some of the possibilities.



To create and test the application with Monitor-51, you need to select the **Monitor** as target. Our example program is correctly configured for the ADuC812 board. For detailed information about the tool setup for the ADuC812 board refer to "Create Own µVision2 Projects" on page 7. The Flash target is a configuration for loading the application program into the Flash ROM of the ADuC812 device.

Compiling and Linking: When you are ready to compile and link the application, use the **Build Target** command from the Project menu or the toolbar. µVision2 begins to compile and link the source files in BLINKY and displays a message in the **Output Window – Build** when the build is finished.

Setup Monitor Communication: Open the **Project – Options for Target - Debug** dialog and select **Use: Keil Monitor-51**. Then use the **Settings** button to configure the serial communication with the ADuC812 board. µVision2 presents you with the following dialog:



Since the Monitor-51 is configured for automatic baudrate detection you may use any baudrates up to 57600 bps. A good choice is a Baudrate of 38400.

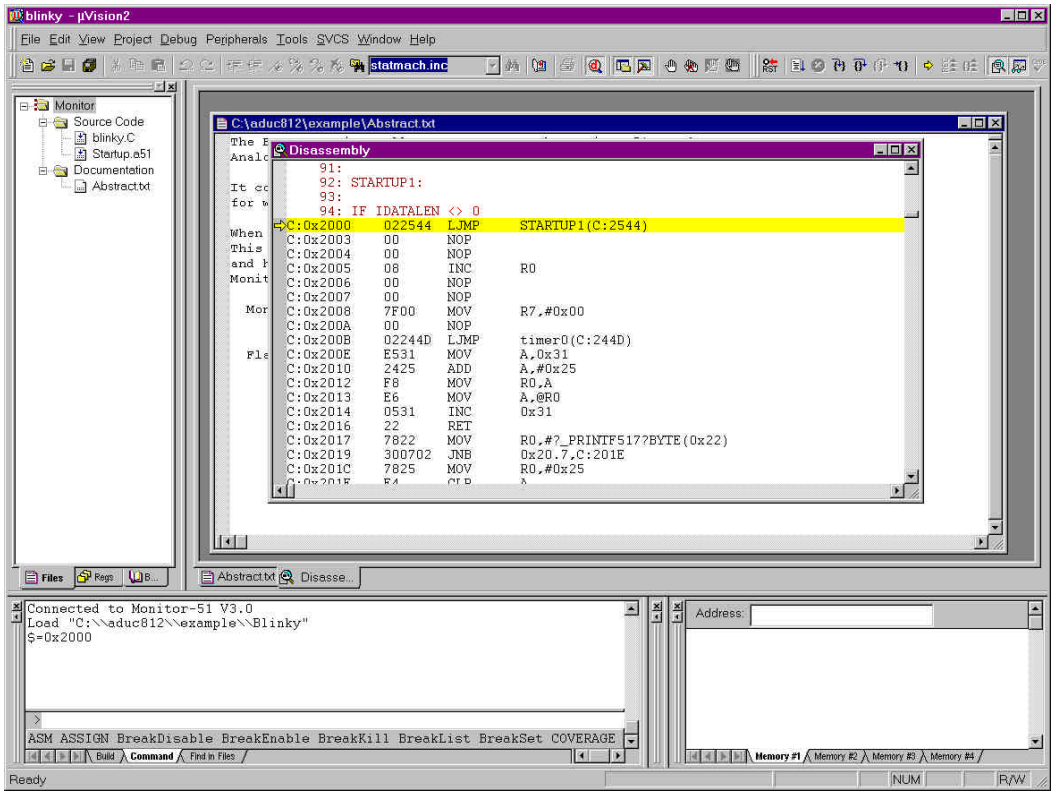
NOTES

If you have problems starting the debug session with the Monitor check the following:

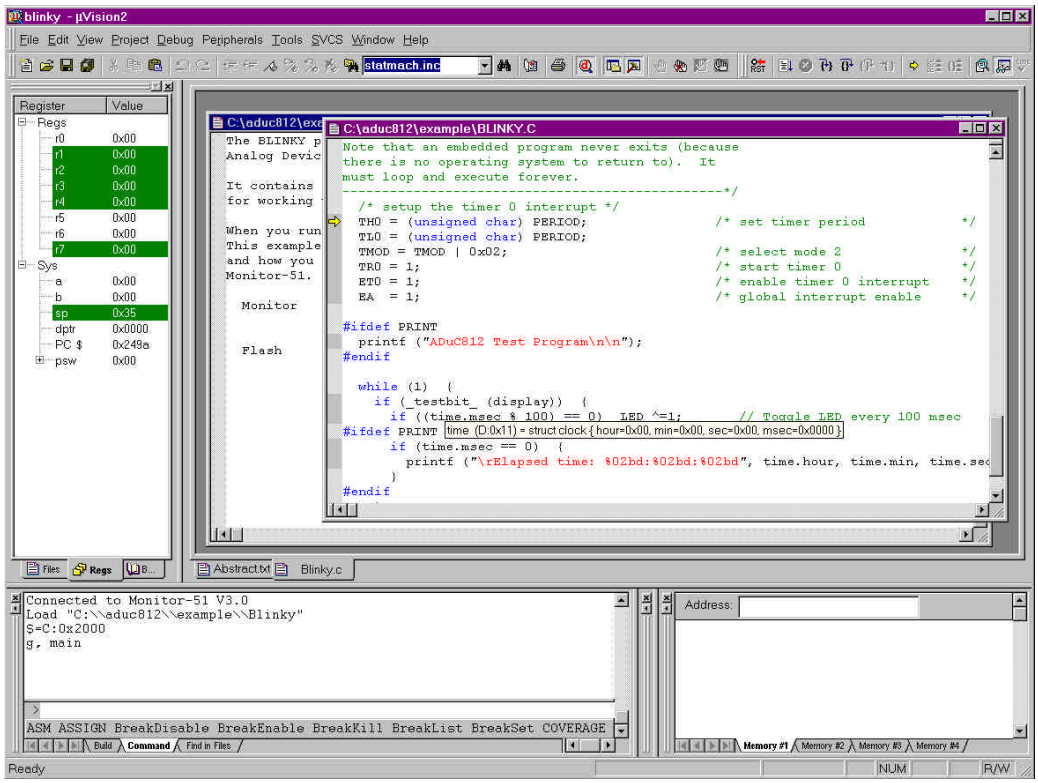
- *Jumper **LK3** on the ADuC812 board must be open.*
- *The board must be connected via a serial link to the PC and the power supply must be ON.*
- *Reset the board again and use the **Try Again** button in the µVision2 debugger.*

Then you may start the debug session. The µVision2 debugger connects to the ADuC812 board, downloads the application program. In case of communication problems a dialog box opens that displays further options and instructions.

Review the program code with **View – Disassembly Window**. If the disassembly does not start at address C:0x2000 enter **$=C:0x2000** in the **Output – Command** window of µVision2. The µVision2 screen should look like shown below:you may start the debug session. This should give the following screen output:



You may now enter **g, main** in the **Output – Command** window to execute the C compiler startup code. Then close the **Disassembly Window** and open the source file **BLINKY.C**. This can be done with the µVision2 command **Debug – Show next statement**. µVision2 presents the following screen:



The Monitor-51 supports common debugging features. You may single step through code, set breakpoints and run your application. Variables can be viewed by placing the mouse cursor on the variable or within the watch window. For detailed information about using the µVision2 debugger refer to the "µVision2 Getting Started with the 8051" User's Guide "Chapter 5. Testing Programs" (Keil\C51\HLP\GS51.PDF) that is included in the Keil 8051 tool chain.

The Keil Monitor-51 allows you to share the serial interface that is used for Monitor communication with user I/O. The **Serial Window #1** shows the *printf* and *putchar* output. Disable the option **Monitor Driver Settings – Serial Interrupt** to enter characters that are sent to the user program.

NOTES

*It is important that you skip the initialization of the serial interface in the user application, since the Monitor performs the UART setup. You may use conditional compilation as shown in our program examples. Also it is impossible to single step through **putchar** or **getkey** I/O functions.*

If the option **Monitor Driver Settings – Serial Interrupt** is enabled, you may stop program execution with **Halt** command from the **Debug** menu or the toolbar or type **ESC** in **Output Window – Command page.** The Monitor uses the serial interrupt to halt the user program. If the Monitor cannot stop your program (because the user application has disabled interrupts or so) a dialog box opens that displays further options and instructions.

The µVision2 **Reset** command sets the program counter to 0. However it should be noted that peripherals and SFRs of the ADuC812 device are not set into reset state. Therefore this command it is not identical with a hardware reset of the CPU.

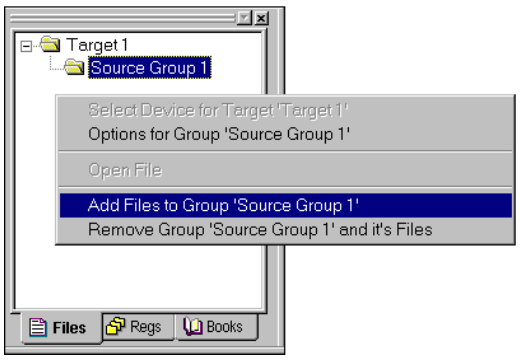
Create Own µVision2 Projects

To create a new project file select from the µVision2 menu **Project – New Project…**. This opens a standard Windows dialog that asks you for the new project file name.

We suggest that you use a separate folder for each project. You can simply use the icon **Create New Folder** in this dialog to get a new empty folder. Then select this folder and enter the file name for the new project, i.e. **Project1**. µVision2 creates a new project file with the name **PROJECT1.UV2** which contains a default target and file group name. You can see these names in the **Project Window – Files**.

Now use from the menu **Project – Select Device for Target** and select the Analog Devices ADuC812 CPU for your project. The **Select Device** dialog box shows the µVision2 device database. This selection sets necessary tool options for the ADuC812 device and simplifies in this way the tool configuration.

You may create a new source file with the menu option **File – New**. This opens an empty editor window where you can enter your source code. µVision2 enables the C color syntax highlighting when you save your file with the dialog **File – Save As…** under a filename with the extension ***.C**.



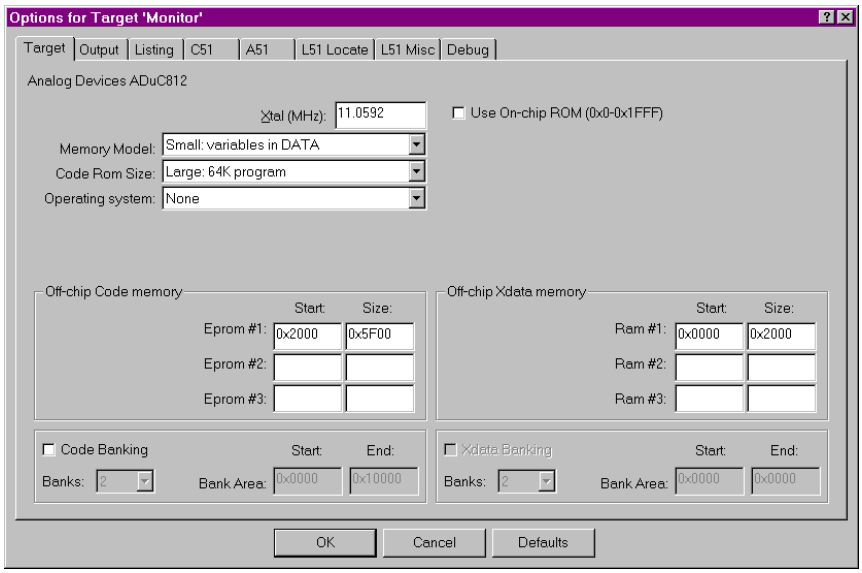
Once you have created your source file you can add this file to your project. µVision2 offers several ways to add source files to a project. For example, you can select the file group in the **Project Window – Files** page and click with the right mouse key to open a local menu. The option **Add Files** opens the standard files dialog. Select the file **MAIN.C** you have just created.

When you are using the Monitor-51 on the ADuC812 board you need a modified version of the C startup code. We

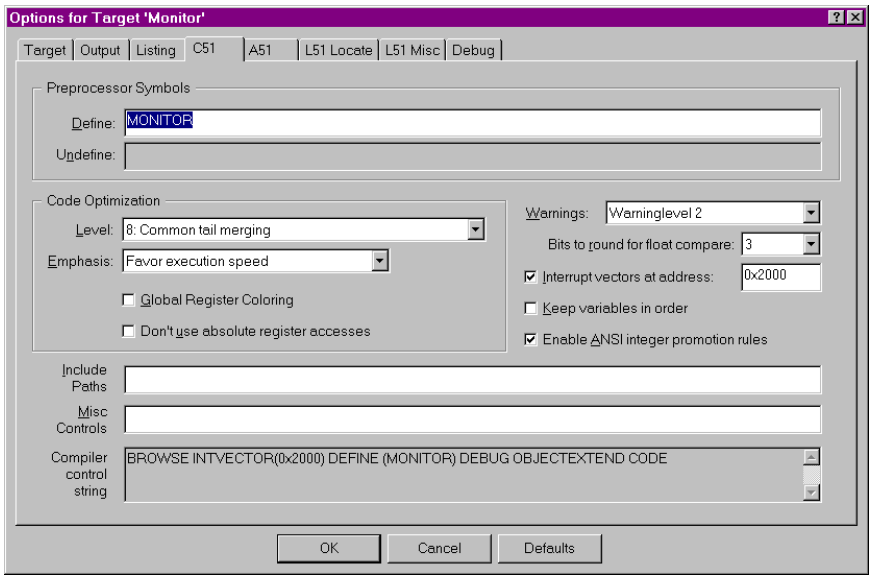
recommend that you simply copy the file **\ADUC812\EXAMPLE\STARTUP.A51** from the BLINKY example and insert this file into your project. Important is that this file has the statement **CSEG AT 2000H** before the **LJMP STARTUP1** instruction.

Set Tool Options for Using the Monitor-51

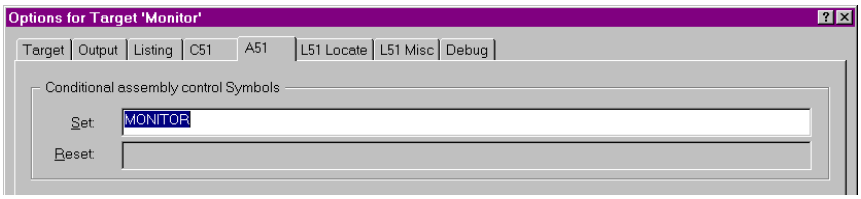
µVision2 lets you set options for your target hardware. The dialog **Options for Target** opens via the toolbar icon. In the **Target** tab you specify the CPU Xtal frequency, and the address ranges for the code memory and the xdata memory. Refer to "Memory Mapping on the ADuC812 Board" on page 4 for a discussion.



In our program example we have used conditional translation to adapt the program code to the Monitor-51. Therefore enter in the **C51** tab under **Define**: **MONITOR**. Since the interrupt vectors on the ADuC812 board are redirected you need to enter also **Interrupt vectors at address: 0x2000** in this dialog page. Now you may use the #ifdef MONITOR and #ifndef MONITOR in your C sources to modify the program code for using with the Monitor-51 or using on a standard ADuC812 device.

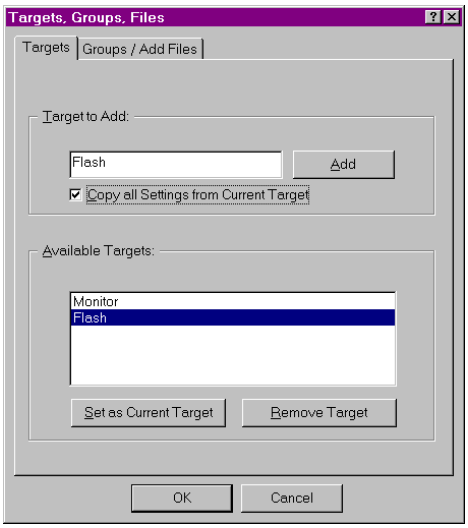


We have used the same MONITOR symbol also in the STARTUP.A51 file to redirect the reset jump. Therefore you must enter in the A51 tab under **Set: MONITOR**.



To configure the µVision2 debugger, you select in the **Debug** tab **Use: Keil Monitor-51**. Then use the **Settings** button as described under "Using µVision2" on page 4 to configure the serial communication with the ADuC812 board. This completes the tool setup for the ADuC812 board.

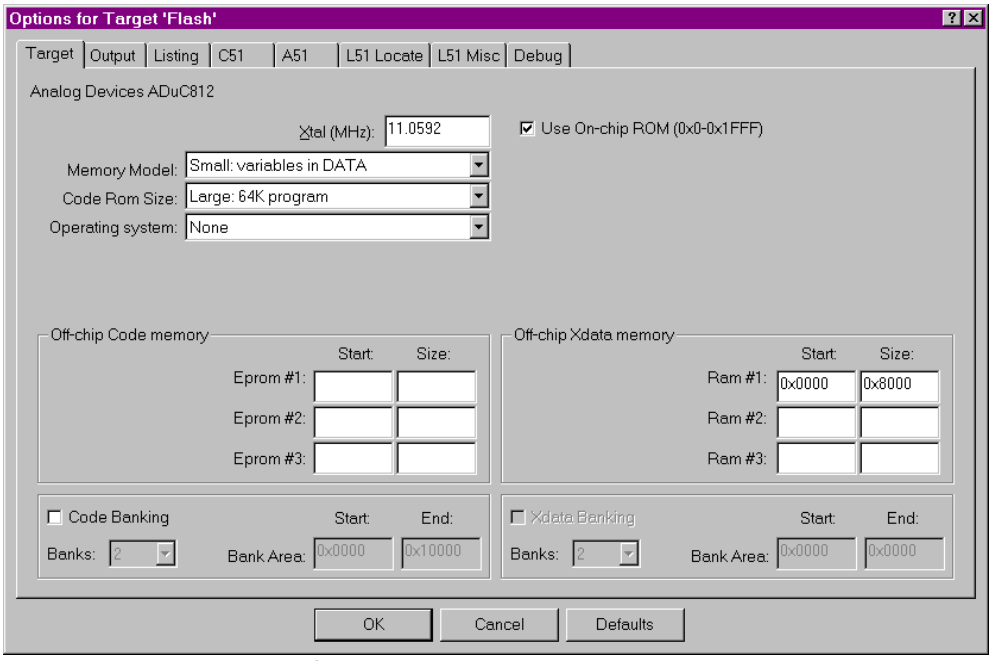
Set Tool Options to Download the Application into Flash ROM



You may create with µVision2 a second target that allows you to run the program directly from the Flash ROM. µVision2 offers **Targets** that allow you different tool settings for the same project. You create a target with the **Project – Target, Groups, Files…** dialog.

Then you use the Project – Options for Target dialog to change the tool settings:

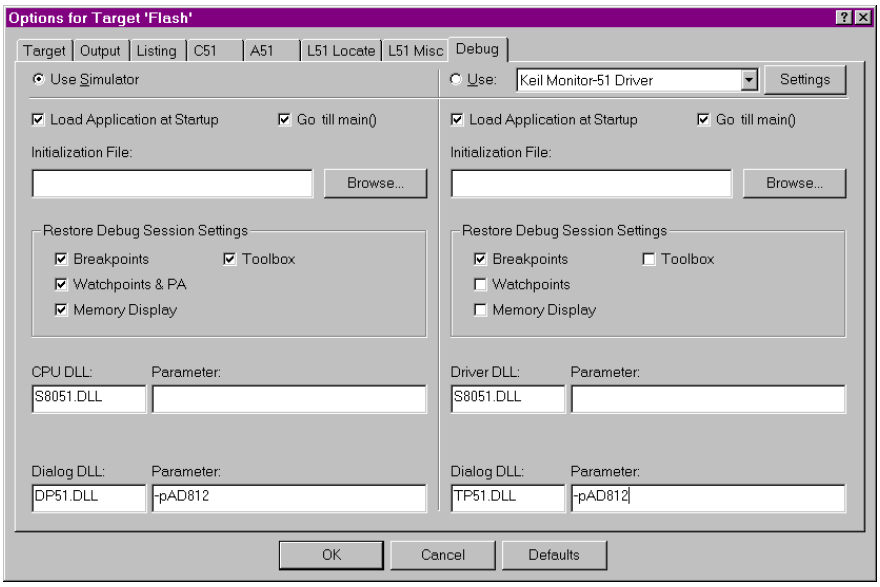
- in the **Target** tab enter the memory layout as shown below.
- the Flash **DOWNLOAD.EXE** utility form Analog Devices needs a Intel HEX-file. Therefore enable in the **Output** tab **Create HEX file**.
- in the C51 tab remove the **Define: MONITOR** are enter **Interrupt vectors at address setting: 0**
- in the A51 tab remove **Set: MONITOR**.



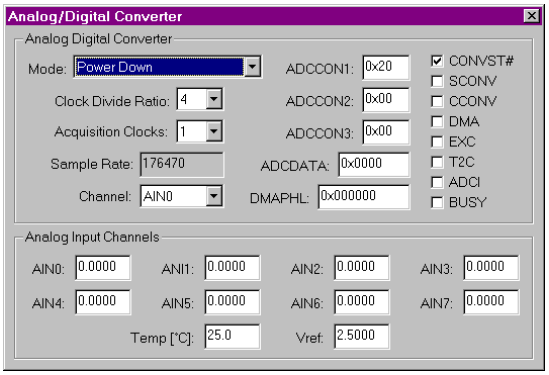
Settings in the Target Tab for using the on-chip Flash ROM.

Using the µVision2 Simulator

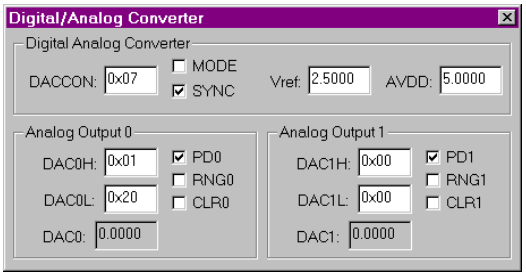
In the C51 Version 6.02 package we provide also simulation support for the ADuC812 device. You may configure the µVision2 simulator under **Options for Target – Debug**. Select **Use Simulator** to debug your application with the µVision2 simulator.



The µVision2 debugger simulates the ADuC812 specify peripherals and provides in the peripherals menu of the µVision2 debugger dialogs for the ADuC812 specific peripherals.



Analog/Digital Converter Dialog



Digital/Analog Converter Dialog

The simulator also defines several VTREG's that can be used with the µVision2 debug functions. This allows you to write signal functions or to use automated tests. For detailed information about using VTREG's refer to the "µVision2 Getting Started with the 8051" User's Guide "Chapter 6. µVision2 Debug Functions" (Keil\C51\HLP\GS51.PDF) that is included in the Keil 8051 tool chain.

Conclusion

Keil C51 Version 6 is the most efficient and flexible 8051 development tool platform available. Keil C51 Version 6.02 or higher contains full support for the ADuC812 software development. Once you understand the concept and handling of the µVision2 IDE you will find the tool chain easy to use. This allows you to create ADuC812 applications faster than ever before.

On http://www.keil.com we provide you with current product information and a multitude of application notes. We also publish on the Internet our "support solutions" database, containing over 1000 current technical questions and answers.