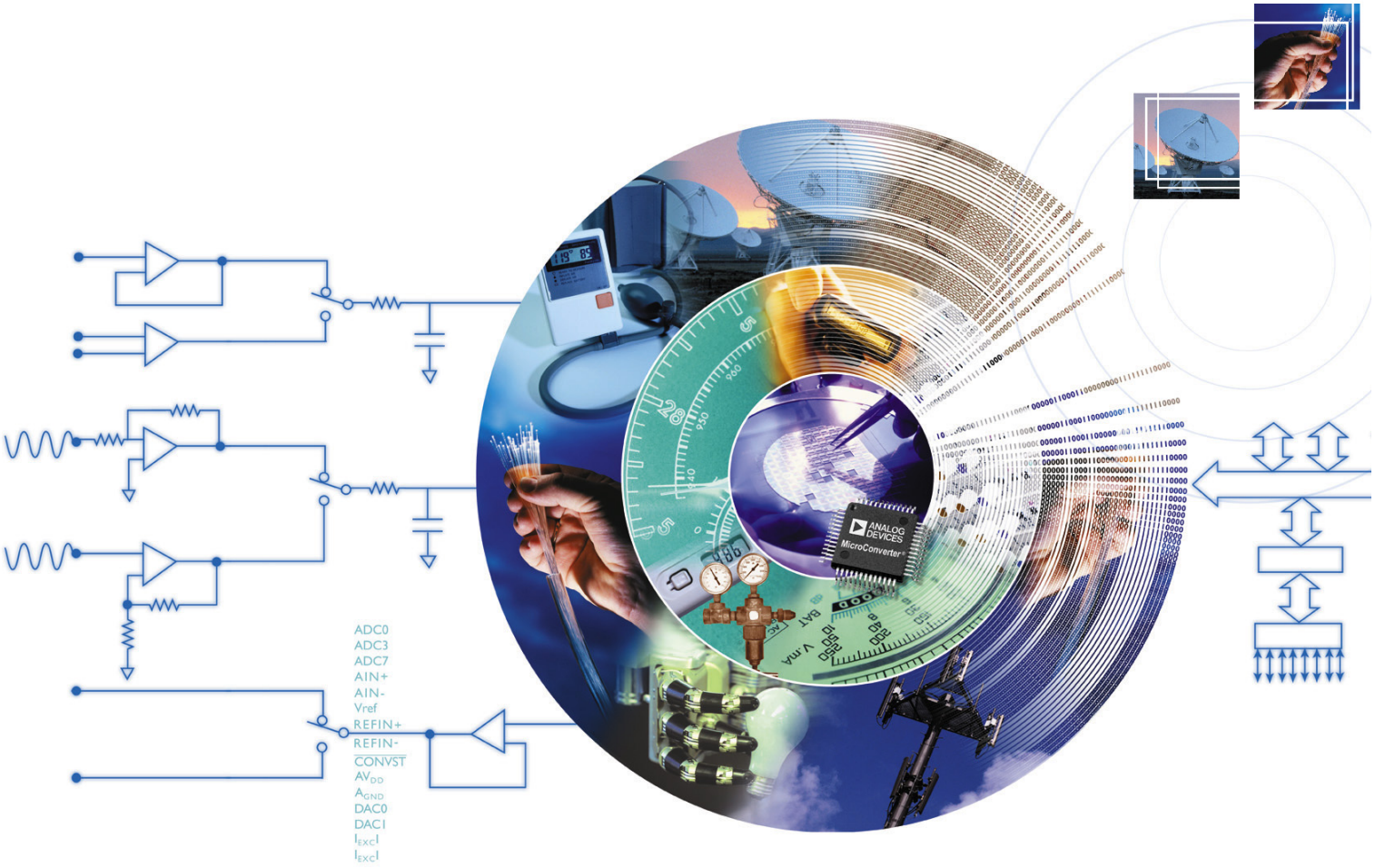
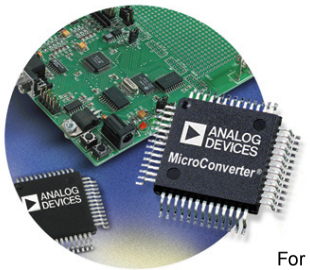
MicroConverter®





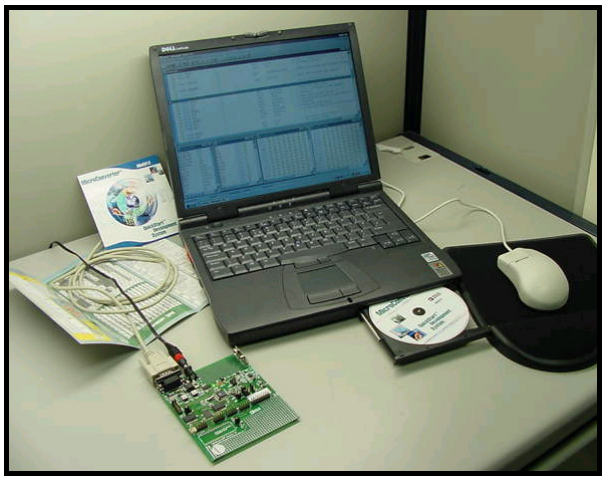
For product information, visit our web page at www.analog.com/microconverter

QuickStart™ Development System

GetStarted Guide

ANALOG DEVICES

One Technology Way, P.O. Box 9106, Norwood MA. 02062-9106, USA.
Tel. (781) 329-4700, Fax. (781) 326-8703



The ADuC8xx QuickStart Development System

Technical Support:

North America & ROW: linear.apps@analog.com
Europe: euro.linear@analog.com
China: china.support@analog.com

The following GetStarted tutorial guide will bring the user through the various tools that are part of the MicroConverter QuickStart development system. As all our tools are ADuC8XX generic, this tutorial guide should be read for a development on any of our ADuC8XX parts.

The ADuC8XX parts and the evaluation boards that are referenced in this tutorial guide are as follows:

PART	EVALUATION BOARD
ADuC812	MicroConverter SAR Eval Board Rev A3
ADuC814	Eval-ADuC814QS SAR Eval Board Rev B1
ADuC831	MicroConverter SAR Eval Board Rev A3
ADuC832	MicroConverter SAR Eval Board Rev A3
ADuC841	MicroConverter SAR Eval Board Rev A3
ADuC842	MicroConverter SAR Eval Board Rev A3
ADuC816	MicroConverter ΣΔ Eval Board Rev B
ADuC824	MicroConverter ΣΔ Eval Board Rev B
ADuC834	MicroConverter ΣΔ Eval Board Rev B
ADuC836	MicroConverter ΣΔ Eval Board Rev B
ADuC844	MicroConverter ΣΔ Eval Board Rev B
ADuC845	MicroConverter ΣΔ Eval Board Rev B
ADuC846	MicroConverter ΣΔ Eval Board Rev B

The tools discussed during this GetStarted tutorial guide are as follows:

TOOL	EXECUTABLE	FUNCTION
ASSEMBLER	Asm51.exe	The Metalink 8051 Cross Assembler takes an assembly language source file created with a text editor, saved with a .ASM extension and translates it into two files, a listing file output (.lst) and a machine language object file in standard Intel Hex format (.hex).
DOWNLOADER	WSD.exe	The Windows Serial Downloader (WSD) is a windows software program that allows a user to serially download standard Intel Hex files, as created by the ASM51 assembler, to the MicroConverter, while in circuit.
INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) - ASSEMBLY SOURCE -	ASPIRE.exe	ASPIRE is a complete IDE (Integrated Development Environment) integrating all the tools necessary to edit, assemble, simulate and debug Assembly source code via the serial port. For C-Source non-intrusive emulation, the **QuickStart Plus** System is required.
WINDOWS ANALOG SOFTWARE PROGRAM	WASP.exe	The **Windows Analog Software Program** (WASP), is an analysis tool allowing the user to easily measure the analog noise performance of the MicroConverter.

Installing from CD:

- Insert the MicroConverter® QuickStart™ Development System CD ROM into your CD ROM drive, select the CD-ROM drive and double-click on the file "setup.exe".
- Follow the on screen instructions to install the software on your PC.

Notes:

- Although you can install the software onto any hard drive and into any directory you wish, for the purposes of simplicity the rest of this document will assume that you have installed the software at the default location of C:\ADuC.
- If you already have a previous ADuC8XX QuickStart Development System tool-suite installed on your machine, this version may also be installed by default at C:\ADuC. The ADuC8XX Software Tools installation will automatically update any previous ADuC8XX tools in this directory.
- The ASPIRE IDE is only available for Windows 98, Windows Me, Windows 2000, Windows NT and Windows XP.

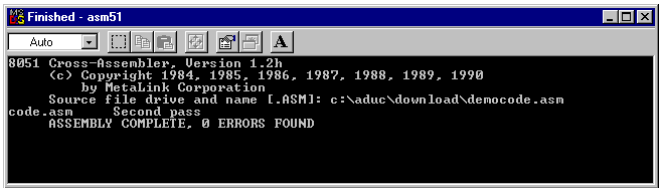
The Metalink 8051 Cross Assembler takes an assembly language source file created with a text editor, saved with a .ASM extension and creates two files, an output list file (.LST) and a machine language object file in standard Intel Hex format (.HEX).

The list file output (.LST) displays the results of the assembler operation, including any syntax or other errors present in the original source code.
The Intel Hex file (.HEX) is used to program the part using the Windows Serial Downloader (WSD) as described in section 3.0.

2.1 Using the Metalink Assembler

1. In the C:\ADuC\ASM51 directory, double-click on the *ASM51.exe* executable.

2. In the DOS window that comes up, type the path of the assembly file you wish to assemble. For example, to assemble the example file C:\ADuC\Download\DemoCode.asm, simply type "C:\ADuC\Download\DemoCode.asm" as shown below.



The assembler will display the text "ASSEMBLY COMPLETE, 0 ERRORS FOUND" indicating that it has successfully assembled the file and has created the hex and list files (i.e. DemoCode.hex & DemoCode.lst) along side the assembly input file (i.e. DemoCode.asm). If the assembler indicates assembly errors, you should view the list file (i.e. DemoCode.lst) to examine the errors. To view the list file, open it with notepad or any standard text editor.

Note: ***If the assembler returns an error message indicating a failure to read drive A, or a fatal error opening a file on the A drive, then it is most likely failing to find the MOD52 or MOD8XX file referenced by the assembly file. Make sure all MOD files (plus any other "include" files referenced in your assembly code) are located in the C:\ADuC\ASM51 directory (where you clicked the ASM51.exe executable).***
The ASM51.exe program can be copied/moved to another directory to prevent typing in the long path name each time. Make sure that the relevant MOD files are also moved with the ASM51.exe program.

For additional details on the use of the Metalink ASM51 assembler, please refer to the ASM51 user manual at C:\ADuC\ASM51\ASM51.pdf.

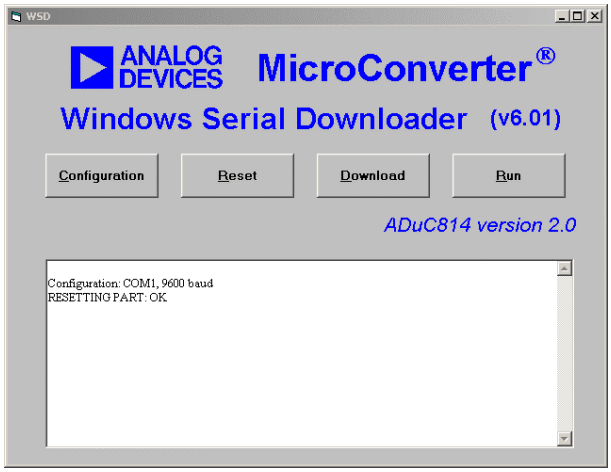
The Windows Serial Downloader (WSD) is a windows software program that allows a user to serially download standard Intel Hex files as created by the ASM51 assembler to the MicroConverter via the serial port. The standard Intel hex file is downloaded into the on-chip FLASH/EE program memory via a selected PC serial port (COM1 to COM4). The WSD also incorporates the protocols for downloading to FLASH/EE data memory, setting of security bits and various RUN options.

3.1 Opening the Windows Serial Downloader

1. Power up the evaluation board using the 9V power supply. Connect the evaluation board header J4 to your PC's COM1 serial port using the RS-232 dongle cable provided. The PC serial COM port may be changed from COM1 via the WSD '**configuration**' option…see section 3.4 below.

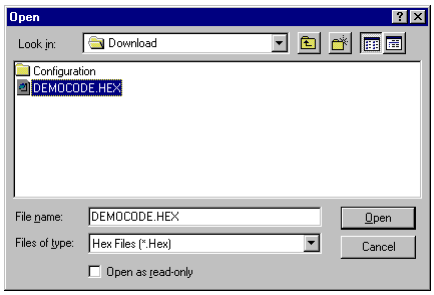
2. The user should put the MicroConverter into serial download mode.
   To enter serial download mode on the **ADuC814** the user should:
   Connect S3 into the DLOAD/DEBUG position and press the RESET button.
   To enter serial download mode on any of the other **ADuC8xx** products the user should:
   While holding down the "SERIAL DOWNLOAD" button press and release the RESET button.

3. From the START menu choose Programs → ADuC → WSD. This launches the Windows Serial Downloader application. The WSD executable is located at
   C:\ADuC\Download\WSD.exe.
   The WSD automatically sends the reset command to the MicroConverter. If the MicroConverter is in serial download mode and the comms between the PC and the evaluation board are setup correctly then the WSD should display the following text above the top right corner of the Status Box.
   'ADuC8XX version 2.Y'
   i.e. the screen shot for an ADuC814 below shows the result….



3.2 Downloading using the WSD

4. Click the Download Button. Select the file at C:\ADuC\Download\DemoCode.hex. Double click on the selected file or click on 'Open' to download the file.



While the file is downloading a progress bar will appear indicating how much of the file has been downloaded.

Once the file has been successfully downloaded the progress bar will disappear and the Status Box will be updated with the message

DOWNLOADING CODE [C:\ADuC\Download\DEMOCODE.HEX]:.....OK

3.3 Running the Downloaded File

Running using the WSD

5. Click on the Run Button. The Status Box is updated with the message.

STARTING USER CODE: ... OK

The program starts running from address 0000h, as can be seen by a flashing LED on the Evaluation Board.

To perform additional downloads; repeat step 2 and press the RESET button on the WSD.

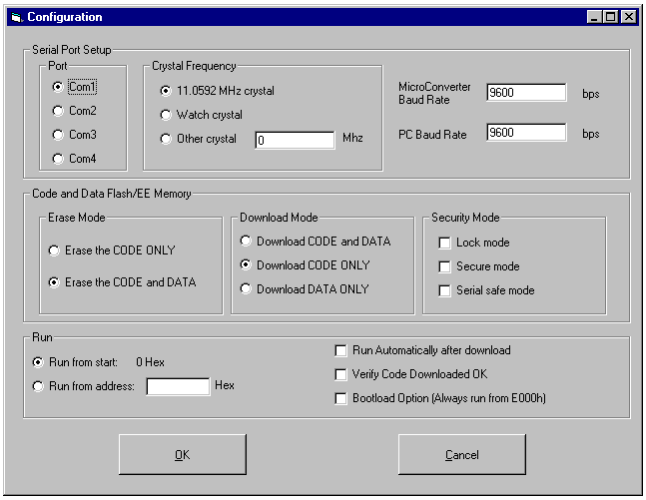
Manual Run Option.

6. Press RESET on the Evaluation Board with the SERIAL DOWNLOAD switch released (for the ADuC814 switch S3 to the NORMAL position). The program starts running automatically after reset as can be seen by the flashing LED.

Note: DemoCode.hex blinks a LED on the eval board. The rate of blinking is reduced each time the INT0 button is pressed

3.4 Additional Download/RUN Options

The MicroConverter incorporates a serial download protocol that also allows various Download/RUN options (see uC004 at C:\ADuC\Documentation\TechNotes). These options can be easily selected in the Configuration window as shown below (the Configuration button can be found on the front panel of the WSD as shown in 3.1.3 previously). As you can see various Erase, Download and RUN options exist here.



Certain options may be grayed out depending on the particular MicroConverter you may have.

Run Automatically after Download

7. Click on the Configuration button. Tick the box for 'Run automatically after download' as shown in the configuration window above. Click on OK. Enter serial download mode as in step 3.1.2. Download as in step 3.2.4. The program starts running automatically after download as can be seen by the flashing LED.

NOTE: Use of the PC COM Port:

Only one application may use the PC serial port at any one time.
The WSD only uses the PC COM serial port when

- Resetting the device
- Downloading to the device
- Sending the Run command to the device

Therefore, the WSD does not have to be closed before launching the Debugger/WASP/Hyperterminal or any other application that uses the PC COM serial port.

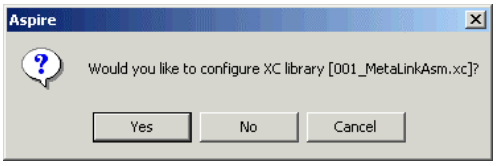
However if another application, that uses the PC serial port, is open then the WSD will not be able to communicate with the MicroConverter until the PC serial port is released by disconnecting/closing the other application.

The ASPIRE IDE integrates all the tools necessary to edit, assemble and debug code. The Quickstart development system supports assembly level debug only. The **QuickStart Plus system** supports full C-Source debug and non-intrusive emulation. Please refer to our web site (www.analog.com/microconverter) for a more detailed overview of the various development tool options available for the MicroConverter family.

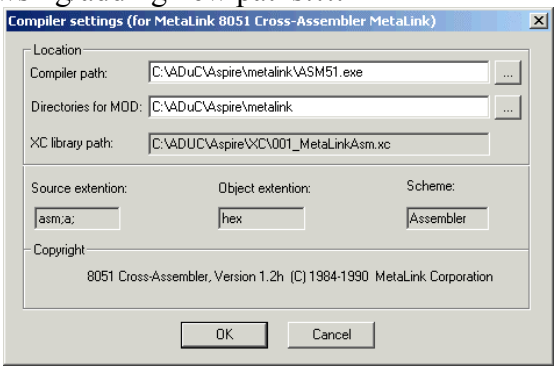
4.1 Starting Aspire

1. From the START menu choose Programs → ADuC → ASPIRE. This launches the Aspire IDE application. The Aspire executable is located at C:\ADuC\Aspire\bin\Aspire.exe.

2. Following Aspire launch, you may be prompted as to what assembler/compiler you would like to use. If not go to TOOLS → RESCAN COMPILERS and follow the instructions….



By default the assembler used is the MetaLink assembler and you should click YES to this , as shown above. Now another window will open asking you for the paths to the MetaLink assembler….delete incorrect path names before browsing/adding new paths….



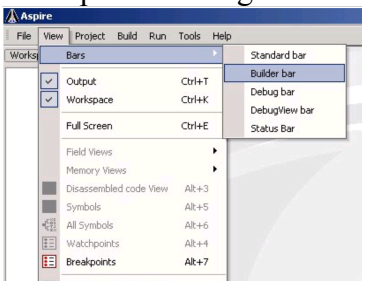
The MetaLink tools are located at C:\ADuC\Aspire\MetaLink\ASM51.EXE. Everything required is included here…..the assembler/compiler (ASM51.EXE) and all the MOD/header files for the MicroConverter family. Click OK as shown above.

ASPIRE allows other compilers/assemblers to be used instead of or as well as the MetaLink assembler. Following the above steps you will be asked if you want to configure these. Click NO to all subsequent options. Compilers/Assembler options can be retargeted at any time by going to TOOLS → RESCAN COMPILERS and selecting the options you require.

4.2 Toolbars

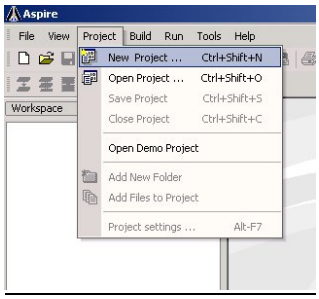
3. Under the VIEW menu, click on the Bars menu item and open the following Toolbars.
   a. Standard Toolbar
   b. Builder Toolbar

   These Toolbars will be active only when the ASPIRE IDE is in Edit/Compile mode. Any Toolbar opened in Download/DeBug mode will be hidden upon returning to Edit/Compile mode.

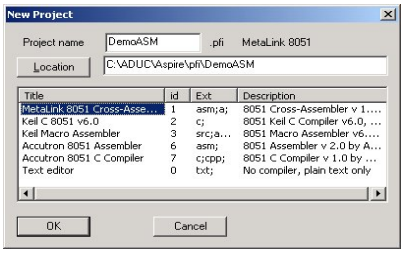


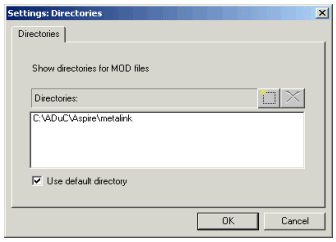
4.3 Starting a project

4. From the project menu select New Project as shown below.



5. Enter the name of the project as **DemoASM** and chose the **MetaLink 8051 Cross-Assembler** in the Title window and click OK. You will then be asked for the directory for the MOD header files (C:\ADuC\Aspire\metalink\…), this can be left as the default since this will have been configured earlier if the default assembler (C:\ADuC\ASPIRE\MetaLink\ASM51.EXE) was used.



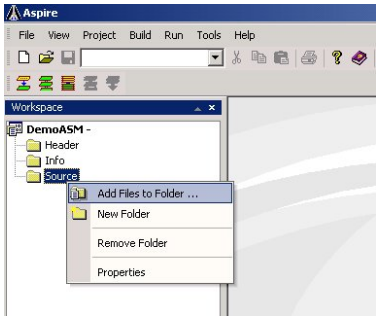


The project folder **DemoASM** is now located in C:\ADuC\Aspire\pfi\… path

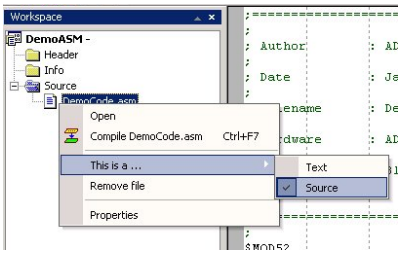
6. The new project will now take the following structure in the **Workspace** window of ASPIRE…The **Header folder** is where any include files reside. The **Info folder** is used to allow the user keep track of project status and for general note taking. The **Source folder** is where the program source files for the project are placed.

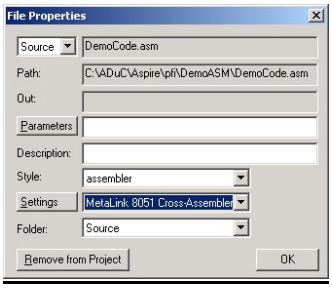
7. To use Aspire, an assembly program is required in the project folder just created (DemoASM). This assembly file can be created from new by clicking on File → New and writing the program in the **New Assembler Text** window that pops up. However, for simplicity, in this tutorial an assembly file is provided for you (DemoCode.ASM) located in the C:\ ADuC\Download\… folder. Use Windows Explorer to copy this file into the folder created earlier (C:\ADuC\ASPIRE\pfi\DemoASM).

8. This DemoCode.asm file now needs to be added to the **Source folder** in the **WorkSpace** window. To add a file to an Aspire project, right click on the Source folder and select "Add Files to Folder". Select the file C:\ADuC\ASPIRE\pfi\DemoASM\DemoCode.asm and click OPEN.

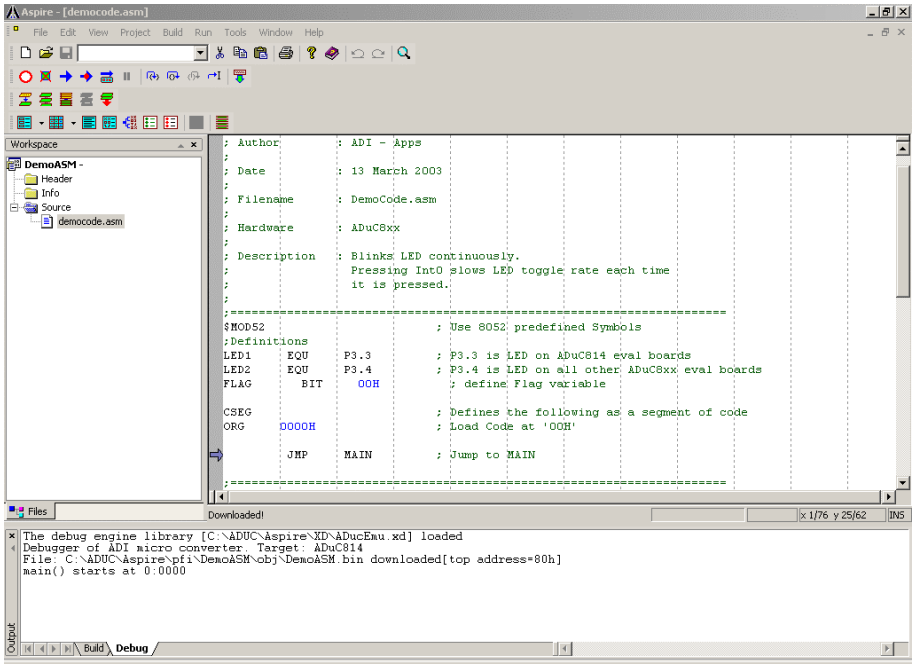


9. Select the file from the Source folder. This file should automatically be configured as an assembly source file with the Metalink assembler as its compiler. To check if the file is a source file, right click on the file, select "This is a …" and verify that the SOURCE option is ticked. If the TEXT option is selected, tick the SOURCE option instead. Files selected as TEXT, will not be assembled/compiled by the assembler/compiler.
Right click on the source file again but this time select the Properties option. Make sure that the "Metalink 8051 Cross-Assembler" appears in the settings option



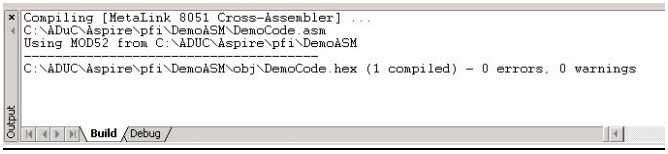


10. If the source file does not automatically open, then it can be opened by double clicking on the file name (DemoCode.asm) in the Source folder in the Workspace window. The screen should appear as follows.



4.4 Assembling/Compiling Code

11. To assemble the DemoCode.asm file, click on the compile ( ) icon in the toolbar. The file should assemble correctly and the following will be seen in the status window. If there are errors in your source code, these will appear in the status window. To identify the line of code which corresponds to the error double click on the error in the Output window and an arrow will appear highlighting the line of code in which the error appears.



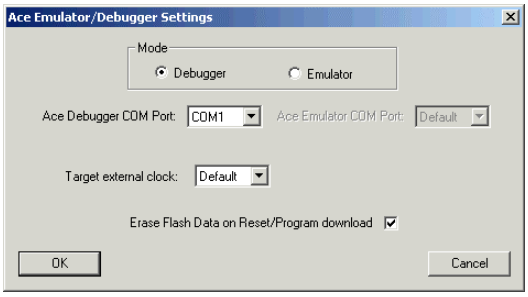
12. Before the code can be downloaded to the MicroConverter the entire project must be built. This is done by clicking on the build ( ) icon on the toolbar.

4.5 Downloading/Debugging Code

13. Power up the evaluation board using the 9V power supply. Connect your PC's COM1 serial port to the 4-way header, J4, on the evaluation board using the RS-232 dongle cable. The PC serial COM port may be changed from COM1 via the Aspire
    Run → ADuC Hardware Emulator Setup option.

14. The user should put the MicroConverter into serial download mode as described in section 3.1.2.

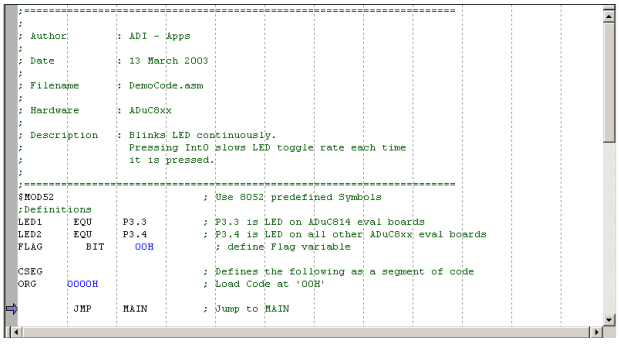
15. Under the RUN menu choose "ADuC Hardware Emulator Setup". Make sure that the **Debugger** option is ticked in the MODE section. The **Emulator** option allows Non-Intrusive emulation of the MicroConverter and is only available as part of the QuickStart Plus Development System. Click OK as shown below.



16. To enter Download/DeBug mode on Aspire, click on the Start Debug Session ( ) icon in the toolbar. This starts the debugger and downloads the code to the MicroConverter. The output window should be updated as shown below (Note: the example below targets an ADuC832 MicroConverter).

```
The debug engine library [C:\ADUC\Aspire\XD\ADucEmu.xd] loaded
Debugger of ADI micro converter. Target: ADuC832
File: C:\ADUC\Aspire\pfi\DemoASM\obj\DemoASM.bin downloaded[top address=2Bh]
main() starts at 0:0000
```

A blue arrow in the code window indicates the next line to be executed as shown below.

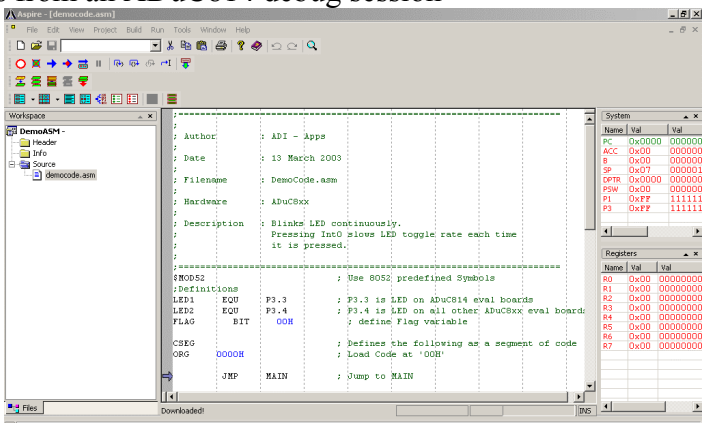


```
;=========================================================================
;
; Author        : ADI - Apps
;
; Date          : 13 March 2003
;
; Filename      : DemoCode.asm
;
; Hardware      : ADuC8xx
;
; Description   : Blinks LED continuously.
;                 Pressing Int0 slows LED toggle rate each time
;                 it is pressed.
;
;=========================================================================
$MOD52                          ; Use 8052 predefined Symbols
;Definitions
LED1    EQU     P3.3            ; P3.3 is LED on ADuC814 eval boards
LED2    EQU     P3.4            ; P3.4 is LED on all other ADuC8xx eval boards
FLAG      BIT     00H             ; define Flag variable

CSEG                            ; Defines the following as a segment of code
ORG     0000H                   ; Load Code at '00H'

        JMP     MAIN            ; Jump to MAIN
```

If any error messages are returned, try resetting the MicroConverter as described in step 4.5.14. Also check the ADuC Hardware Emulator setup under the RUN menu. This should be configured to use the Debugger as described in section 4.5.15.

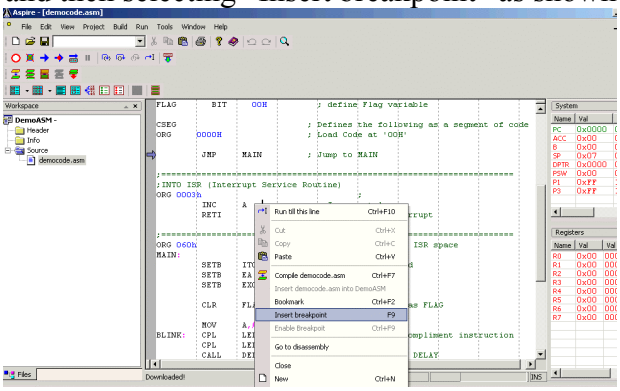
17. To view the DeBug menu icons, open the DEBUG and DEBUGVIEW toolbars under the VIEW→BARS menu. These toolbars will only appear in the Download/DeBug mode, i.e. they will disappear once you enter Edit/Compile mode.

18. Open the System and Registers windows from the **Field View** Icon (). Position the windows as shown below. These windows, by default, appear on the right hand side of the code. These windows can be moved around the workspace. To return them to their original location double click on the window name. The screen shot below is from an ADuC814 debug session



Note: Data values colored blue or green, indicate that this data was just uploaded and hence the values accurately represent those currently on the chip. If the data is blue, then this data has changed since the last upload. If the data is green, then this data has not changed since the last upload. Data values colored red, indicate that this data was not just uploaded, hence these values may have changed on the chip and may not be accurately represented in the relevant windows.

19. Set a breakpoint on the INC A instruction in the INT0 ISR part of the program. This is done by right clicking on the line of code and then selecting “Insert breakpoint” as shown below.



Notice that the breakpoint is indicated by a large red dot to the left of the line.

20. Press the Run (➔) icon. The Run icon will gray out and the program execute. The L.E.D. at P3.4 (P3.3 for the ADuC814) of the evaluation board will blink continuously. Press the INT0 button on the evaluation board, note that the blue arrow and the red dot now indicate the same line. This means that the program has stopped executing and is waiting at the breakpoint location. Also note that the Run icon has become active again waiting for the user to restart code execution. Following each breakpoint, Aspire will upload the specified registers (see section 4.5.21). Note the value of the accumulator (ACC) in the system register window on the right hand side of Aspire…it should read 0x01. Repeat the operation above and note that the LED blinks slower and also that the value in ACC increments with each INT0 button press and run sequence.

21. If you wish to change the automatic register uploads, change the settings in the RUN→Select Uploadable Areas/SFRs. By default both the Internal RAM and System registers are automatically uploaded.

22. To single step through the program, press the Step () icon. You will see the blue arrow step down through the program line by line as this icon is pressed.

23. To run the program to a particular point without having to set a breakpoint use the 'Run To' option. Left click on the RET instruction. Select the Run To () icon, or Right click on the instruction and select 'Run Till This Line' from the pop-up menu. Code execution resumes at the current instruction and runs till it hits a temporary breakpoint at the RET instruction. All specified registers are uploaded at this point.

24. To change the format of the data in any register view window, double click on the current value to change it from Hexadecimal to Binary to Decimal with each double click.

25. To change a value in any register view window, simply right click on the value and type the new value. To apply this change click anywhere else in the register view window.

26. Press the Run icon again to resume. Code execution resumes from the current instruction.

4.6 Saving/Closing a Project

27. To save your project click on the save () icon on the toolbar or go to the Project menu and select save project.

28. To close your project go to the project menu and select close project.

29. To remove a closed project from the Workspace window, right click on the project name and select Remove from the pop-up menu.

For more information on the use of the Aspire software, use the 'Help' options within the application.

DeBug Bar

ICON	FUNCTION	DESCRIPTION
	Reset	Resets the target MicroConverter
	Stop DeBug Session	Stop Download/DeBug mode and returns the Aspire to Edit/Compile mode
	Run	Runs MicroConverter Code
	Run with Auto-Resume	Runs MicroConverter code. The code will automatically resume from a breakpoint after the selected uploads have been performed.
	Animate	Performs continuous single step operations, uploading the selected uploads after every single step.
	Break	In emulation mode code execution will stop immediately. In DeBug mode code execution will stop at the next breakpoint.
	Step	Performs a single step operation
	Step Over	Steps over the next function
	Step Out	Steps out of the existing function
	Run to Cursor	Runs the code until the code is about the execute the current instruction at the cursor.
	Download Data	Downloads any SFRs/RAM/Flash values changed from the DeBugger front end to the target MicroConverter.

DeBugView Bar

	Field View	Allows the user to open special memory locations. These include Registers windows, System registers, Analog SFRs etc.
	Memory View	Allows the user to view the various 8051 Memory Windows:
	Dissassembled View	Displays the debug code as disassembled code
	List of Selected Symbols	Displays a list of any selected symbols to be monitored
	All Symbols List	Any Symbol used in the C code will be displayed here
	Watchpoint List	Displays the Watchpoint Window allowing any memory location to be monitored
	Breakpoint List	Displays a list of all the breakpoints
	Serial Interface Window	Displays RS232 data sent from the MicroConverter
	Mixed Assembly	Displays the debug code as source and disassembly code

Builder Bar

	Compile	Compiles the source code
	Build	Compiles all files in the project, links them and builds the hex file
	Re-Build	Re-builds a project. This will not re-compile any source code which has not changed
	Stop Build	Allows the user to stop a build while it is in progress
	Start DeBug Session	Downloads Code to Target and Starts DeBug Session

The Windows Analog Software Program (WASP) is a general application for all MicroConverter products that allows analysis of their analog performance. The WASP recognizes which MicroConverter the PC is communicating with, before automatically downloading the appropriate code. In this tutorial we will briefly introduce both the SAR WASP (for the SAR ADC parts … ADuC812, ADuC814, ADuC831, ADuC832, ADuC841 and the ADuC842) and the ΣΔ WASP (for the ΣΔ ADC parts … ADuC816, ADuC824, ADuC834, ADuC836, ADuC844, ADuC845 and ADuC846). The terms SAR WASP and ΣΔ WASP relate to the same WASP software. The software differentiates between the different products.
After downloading the appropriate code the WASP launches the Acquisition Window. This allows the user to configure, control and analyze the ADC noise performance with the various analog and digital peripherals enabled/disabled.

1. Power up the evaluation board using the 9V power supply. Connect the evaluation board to your PC's COM1 serial port using the RS-232 dongle cable connected to the 4-way header, J4.

2. The user should put the MicroConverter into serial download mode as described in section 3.1.2.

3. From the START menu choose Programs → ADuC → WASP. This launches the WASP application. The WASP executable file, WASP.exe, is located at C:\ADuC\WASP\WASP.exe.

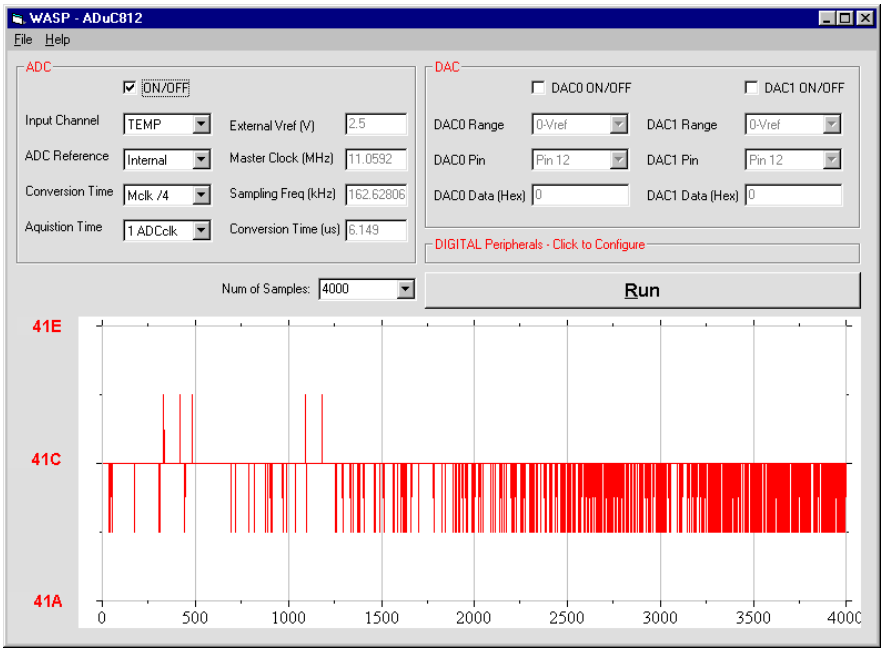
4. Click the **DOWNLOAD** Button. 'ADuC8XX' should appear and the code starts to download. A task bar indicates the download progression. A message appears to tell you when the file is downloaded. The program automatically runs after this download.

Note: The **NEXT** button bypasses the 'Download' sequence and can be used if the WASP code is already downloaded and running on the MicroConverter.To identify the MicroConverter for the WASP software the user should select the appropriate MicroConverter from the "MicroConverter Select" option box and the click on NEXT.

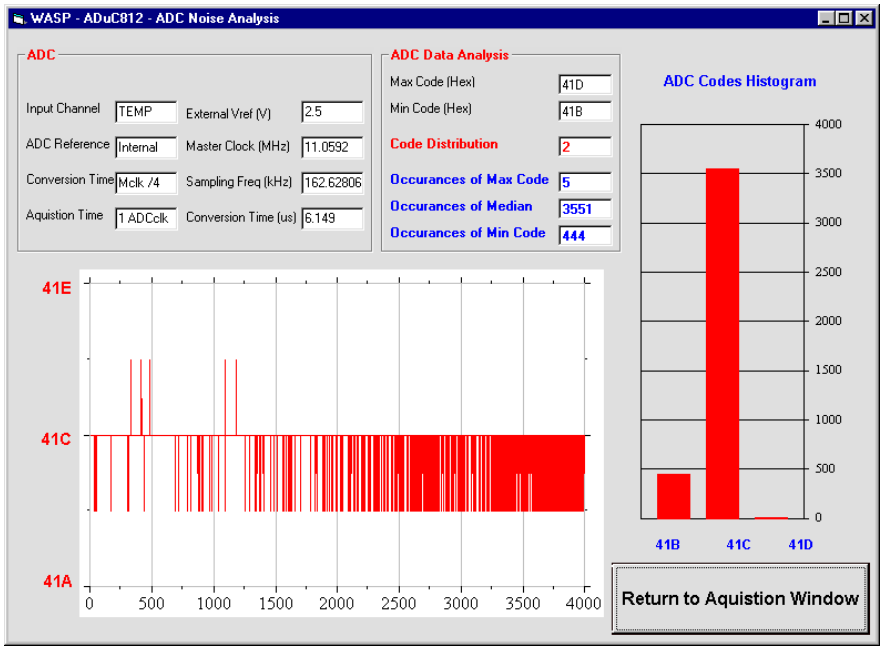
SAR WASP

5. The SAR Acquisition window (as shown below) opens for any of the SAR ADC MicroConverter products (ADuC812, ADuC814, ADuC831, ADuC832, ADuC841 or the ADuC842)
   From the Acquisition window you can…
   a. Select the channel on which you want to convert
   b. Set up the ADC Conversion time and Sampling Parameters.
   c. Select the number of samples that you want to acquire.
   d. Set up voltages on the DAC channels
   e. Select the use of the Internal Reference or an External Reference device.
   f. Enable/Disable various Analog/Digital Peripherals

6. In this example we will convert on the temperature sensor using the internal reference as shown below. With the parameters shown below selected, press the RUN button. The acquired ADC samples will appear on the chart as shown for the ADuC812 example below.



7. When all the samples are collected the WASP immediately launches the Analysis window. The histogram plot and the ADC Data Analysis fields within the analysis window gives a measure of the code distribution for the ADC input.



8. Click on the *Return to Acquisition Window* button in the Noise Analysis window to return to the acquisition panel.

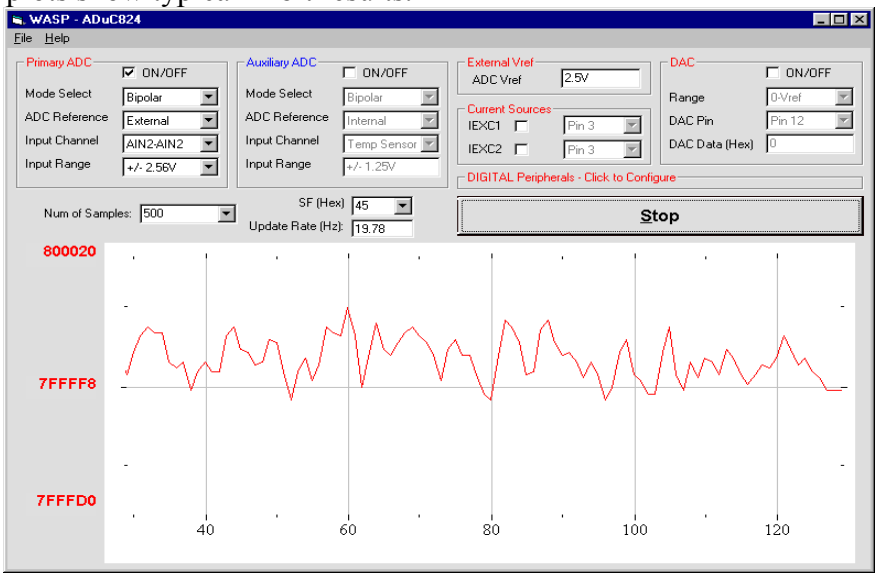
9. The functionality of the DAC(s) and general Digital Peripherals can also be exercised via the options available from the WASP front panel.

Sigma Delta WASP

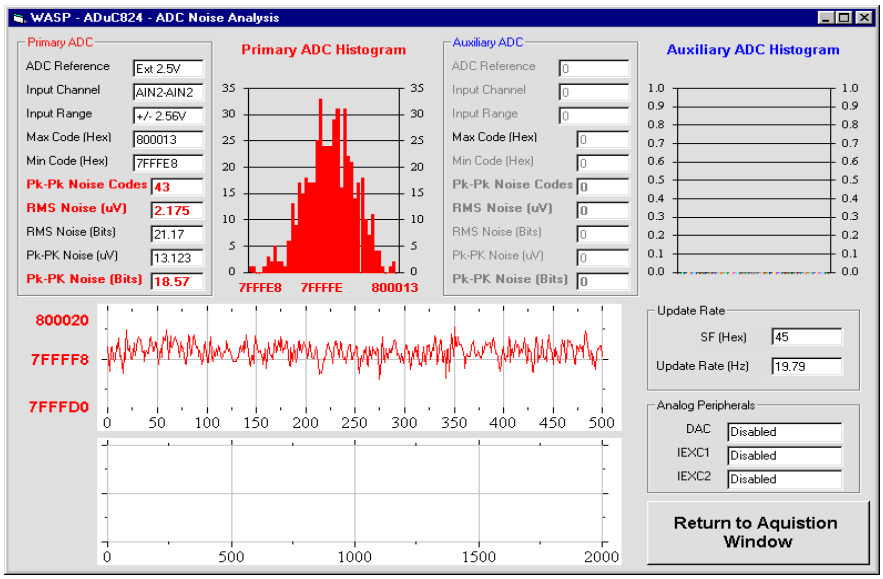
10. The Sigma Delta Acquisition window (as shown below) opens for any of the Sigma Delta ADC MicroConverter products (ADuC816, ADuC824, ADuC834, ADuC836, ADuC844, ADuC845 and ADuC846)
    From the Acquisition window you can…
    a. Select the channel on which you want to convert
    b. Set up the ADC Update Rate.
    c. Select the number of samples that you want to acquire.
    d. Set up voltages on the DAC channels
    e. Select the use of the Internal Reference or an External Reference device.
    f. Enable/Disable various Analog/Digital Peripherals

11. **Note: Switch Configuration**
    Make sure that the external reference (2.5v REF+) is connected (S1.6 ON) and that AIN2 is biased to 2.5V (S1.7 ON). Also ensure that REF- is grounded (S1.5 ON). All Other connections should be OFF.

    By default the WASP enables the Primary ADC configured as below (i.e. Primary ADC converting in bipolar mode using an external reference on the 2.56V range with internally shorted inputs Ain2 → Ain2. The Auxiliary ADC, DAC and Current sources are all disabled).
    Press the RUN button to send this default configuration to the MicroConverter device and begin the conversions. The screen changes to configure for a single Primary ADC acquisition sequence. The results of conversion are displayed in real time. Because the channel is configured for an internal short then we can expect ADC conversions close to 80000h. The WASP performs 500 ADC conversions by default and displays the conversion results.
    The following plots show typical 24bit results.



11. When all the samples are collected the WASP immediately launches the Analysis window. This window displays some mathematical analysis on the ADC conversions, including RMS noise (in µV and bits) and Peak-to-Peak Noise (Code Distribution, µV and Bits). The most important performance figures are highlighted in Red.



12. Click on the *Return to Acquisition Window* button in the Noise Analysis window to return to the acquisition panel.

13. The functionality of the DAC(s) and general Digital Peripherals can also be exercised via the options available from the WASP front panel.

Installing the MicroConverter® QuickStart™ Development System CD installs documentation for all the MicroConverter products at C:\ADuC\Documentation. Directories for each product exist in the Documentation folder, as well as QuickStartTools and TechNotes directories.
Each of the product directories follows a similar folder structure as shown below. All Technical Notes for any of the MicroConverter products appear in the \TechNotes directory. Check our website for the latest tech notes (www.analog.com/microconverter).

```
C:\ADuC\Documentation\ADuC8XX\
   DataSheets\
        ADuC8XX_Y.pdfADuC8XX   DataSheet version Y
        Errata8XX_Y.pdf        ADuC8XX Errata Sheet version Y
        8XXqrefY.pdf           ADuC8XX Quick Reference Guide version Y
   EvalDocs\
        8XXEvalGuide_Y.pdf     ADuC8XX Eval Board Reference Guide version Y
        8XXPCB_Y.pdf           ADuC8XX Eval Board Schematic version Y
        8XXgbrs\8XXgbrs_Y.zip  ADuC8XX Eval Board Gerber files version Y.
   Other\
        8XXFAQs_Y.pdfADuC8XX   Frequently Asked Questions version Y
        8XXgetstartedY.pdf     Get Started Guide version Y
        USERGuideDRAFTY.pdf    Draft User guide version Y
```

It is recommended that all documentation mentioned above be reviewed before starting the QuickStart Development System.

Installed Code Locations

Installing the MicroConverter® QuickStart™ Development System CD installs an Assembly code directory for each MicroConverter products at C:\ADuC\Code. Product directories (e.g. ADuC832 below) for each MicroConverter exist with Assembly code examples.

```
C:\ADuC\Code\832\
       ADC      - code examples for the ADC
       DAC      - code examples for the DAC
       DualDPTR - code example for using the Dual Data Pointer
       FlashEE  - code example for using the Flash/EE Data Memory
       I2C      - code examples for I2C master and slave operation
       Misc     - Miscellaneous MicroConverter code example
       PDown    - code example demonstrating powerdown mode
       PSMon    - code example for the power supply monitor
       SP       - code example for the extended stack pointer
       SPI      - code examples for SPI master and slave operation
       TIC      - code example for the Time Interval counter
       Uart     - code examples for configuring the UART
       WDTimer  - code example for watchdog timer
```

C example code is available in the C-Code directory. C-Code for the ADuC832 is in the \832 folder.

```
C:\ADuC\C-Code\832\
       ADC      - code examples for the ADC
       DAC      - code example for the DAC
       MISC     - Miscellaneous MicroConverter code example
       PDOWN    - code example demonstrating powerdown mode
       PLL      - code example for changing the core execution speed
       TIC      - code example for the Time Interval counter
```