Contents

Note: This guide is an introduction to the QuickStart+ Development System including the basics of the **aspire** development system. A comprehensive User Guide for **aspire** is available in the Help Topics section of the **aspire** program. There is an option available there to print this User Guide.

For further technical support and the latest updates contact us through our website at **www.accutron.com** or call us at **+353 61 331055** or fax us at **+353 61 331965**.

Our Address - Accutron Limited, National Technological Park
Limerick, Ireland. Note: No zip code required

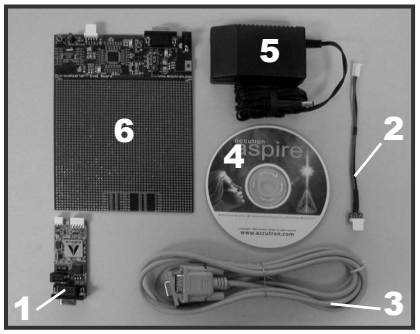
accutron supplying innovative solutions
QuickStart+
Get Started Guide

1. Unpacking, Installation & Assembly

1.1.1 Unpacking QuickStart+

The QuickStart+ Development system contains the following items:

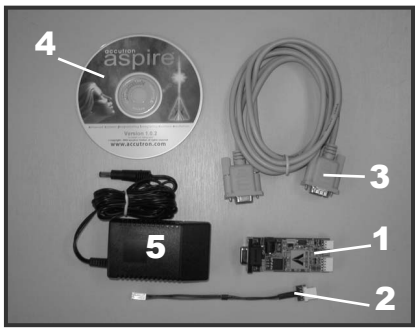
1. Emulator POD.
2. Target Cable (for connecting between the Emulator and your target system).
3. Serial port cable for connecting between Emulator and your PC.
4. aspire software on CD.
5. Emulator Power Supply Unit.
6. Prototyping Board



1.1.2 Unpacking aspire

The **aspire** system contains the following items:

1. Emulator POD.
2. Target Cable (for connecting between the Emulator and your target system).
3. Serial port cable for connecting between emulator and your PC.
4. aspire software on CD.
5. Emulator Power Supply Unit.



Important Note: All example projects provided on the **aspire** software CD including those explained in this guide will run on the **QuickStart+ Development System**. Users of the **aspire** system please note that this system will **only** operate with a target board containing the Analog Devices ADuC family of MicroConverters®. The **aspire** system may be upgraded to a **QuickStart+ Development System** by purchasing an **accutron Prototyping Board.** See our website for details.

1.2 Software Installation

1.2.1 Install the aspire software.. Run the "setup.exe" program located on your **aspire** CD and follow the on-screen instructions.

1.3 Assembly (QuickStart+ Development System)

1.3.1 Connect the Serial Cable to both the Emulator POD and the PC. **Note:** It is recommended practice that you turn off your PC before connecting any device to the serial port on your PC.

1.3.2 Connect the **Emulator POD** to the **Prototyping Board** by means of the 5-Way Connector. (See Fig. 1.)

1.3.3 Connect the **Power Supply** to the Emulator POD. (See Fig. 1.) **Note:** The Prototyping Board derives its power through the 5-Way connector on the Emulator POD. **Do not supply power to both the Emulator POD and Prototyping Board at the same time.**

1.3.4 Place the Prototyping Board in Download/Debug mode. Hold down the **Debug Button** and press and release the **Reset Button.**

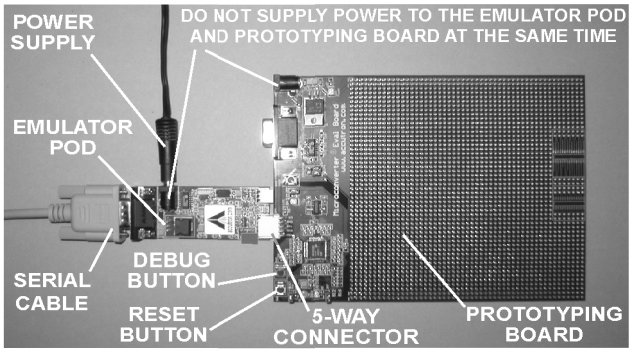
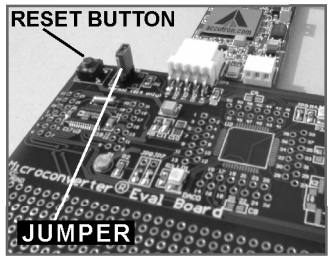


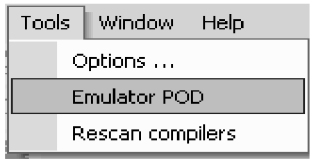
Fig.1. Emulator Pod & Evaluation Board

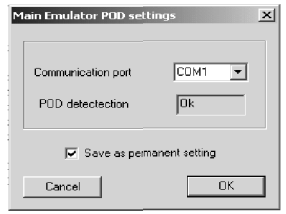
1.3.5 In the Case of the Development Kit containing an ADuC 814,it is only necessary to press the **Reset Button**, ensuring the **Jumper (JP14)** is in the **Debug** position.



1.3.6 From the START menu choose **Programs->Aspire->Aspire**. This loads the ASPIRE IDE, or double click on the desktop icon.

1.3.7 Under the tools menu select Emulator POD as shown here. This will open the main Emulator POD settings dialog as shown.





Ensure that the communications port selected from the drop-down menu is the one to which the Emulator POD is connected. This is indicated in the POD detection field as shown above.

1.3.8 **Assembly (aspire system)**

In cases where you are using a **Target Board** other than the **Prototyping Board**, connect the Target Cable to the Emulator POD and the **Target Board**. (See Fig.2)

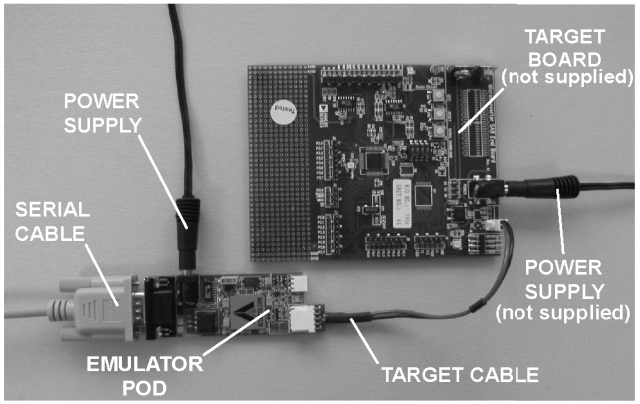


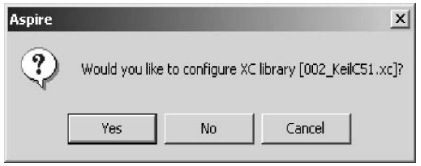
Fig. 2. Emulator Pod & Separate Target Board

2. Working With Aspire

2.1 Starting Aspire

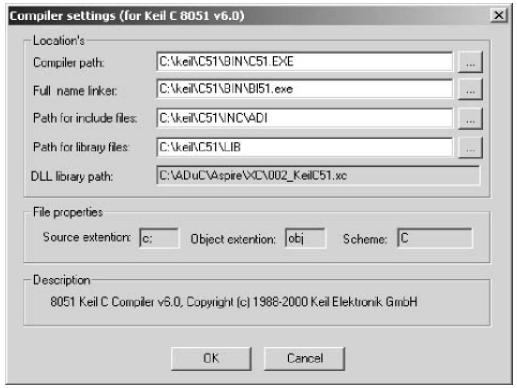
2.1.1 From the START menu choose **Programs->Aspire->Aspire**. This loads the **aspire** IDE. The **aspire** executable is located at C:\ADuC\Aspire\bin\Aspire.exe, or double click on the desktop icon.

2.1.2 The ASPIRE when first opened will ask you to configure the assemblers and compilers that you may have installed on your system.



*Click **Yes** if you wish to configure.*
*Click **No** if you do not wish to configure.*
*Click **Cancel** if you do not wish to configure compiler/linker at this time but would like to be asked again next time aspire is started.*

aspire allows the c compiler from Keil Software to be targeted by the IDE. To configure this compiler enter the correct path for the compiler executable. The paths for the linker, include files etc. will automatically be filled in based on the compiler default directory structure. For the case of the Keil Software compiler the configuration window would look like the dialogue below, based on the default directory structure C:\Keil .



An evaluation copy of Keil is available at \misc\Keilc folder in the install directory on your PC. For the latest version of Keil go to **www.keil.com**.

2.1.3 By default the Metalink freeware assembler, the accutron 8051 'C' compiler and assembler are configured for use with aspire.

2.2 Loading an Example Project

Two sample projects can be found in the **aspire** install directory in the folder "\Examples", one is a "Keil" Project and the other is an "AccuC" Project, each is in it own subfolder. Either can be opened in the following manner:

2.2.1 Select "Project" from the ToolBar and from the dropdown menu select "Open Project".

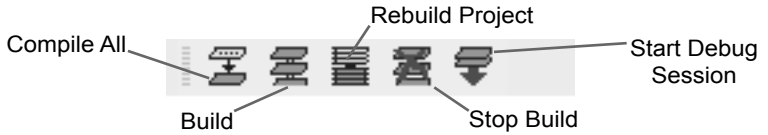
2.2.2 A standard Windows "Open" Dialog is then presented where you can browse for an Aspire project file (*.pfi). The User should Navigate to the directory "\Examples\Keil\Blink" and select the file "Blink.pfi", which is located in the install directory on your PC.

2.2.3 The Blink Workspace will then be displayed which contains two source files "blink.c" and "Delay.a51" which are compiled and built with the Keil Compiler and Assembler.

Note: The Examples directory also contains another folder '\AccuC' which contains an Accutron 8051 'C' Compiler project (AccuC).

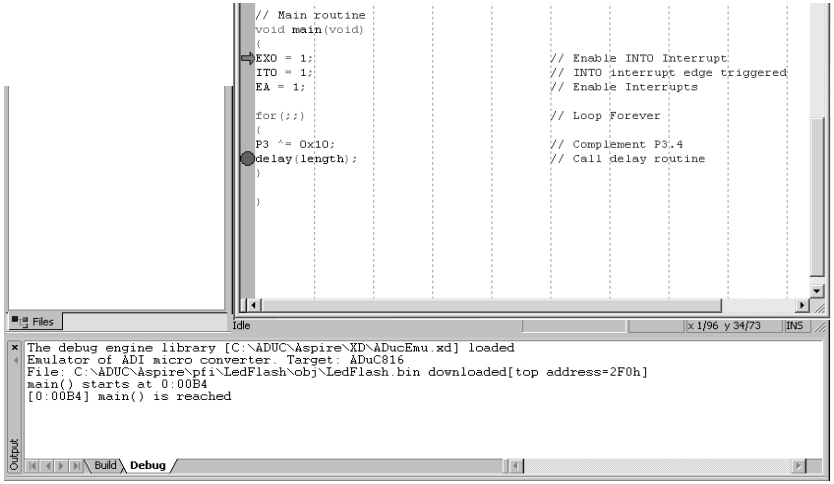
2.3 Loading the Program onto the Target

Note: Before continuing, open the **Builder bar** by selecting the **Bars** option from the **View** menu and clicking on **Build bar**.

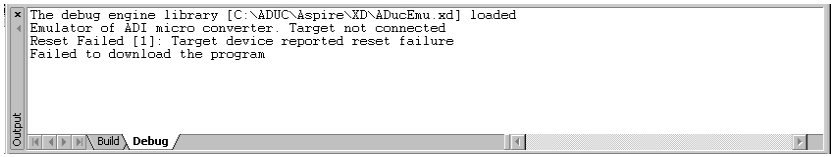


2.3.1 To place the ADuC device into download/debug mode hold down the **Debug Button** and press and release the **Reset Button** on the **Prototyping Board** (see 1.3.4). (For Prototyping Boards using the ADuC 814 see 1.3.5).

2.3.2 To download code onto the target device select 'Load Program' from the run menu, press the icon or F12.

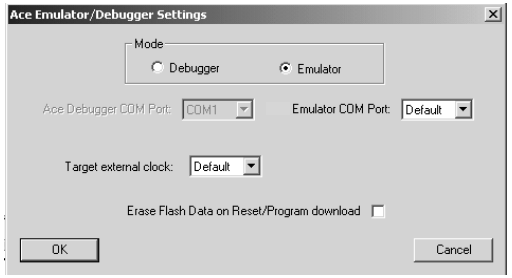


The last message in the status window should state that 'main() is reached'. A Blue Arrow will be located at the first line of main() in the "Blink.c" editor window at line **EX0=1**. If the status window indicates that the device was not programmed (as shown below)



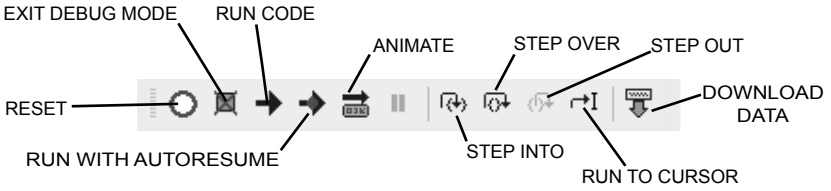
1. Try placing the device in Download\Debug mode.
2. Try checking the **'ADuC Hardware Emulator'** setup under the **Run** menu.

This should be set up to use the Emulator and the correct serial port of your system.



2.4 Runing the Program on the Target

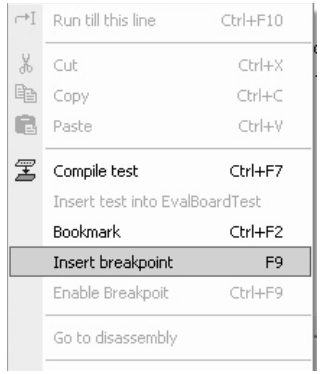
Note: Debug Control options are available in the **Debug bar**. The Debug Bar can be opened by selecting the **Bars** option from the **View** menu and clicking on **Debug bar**. This bar automatically opens on entering **Debug** mode.



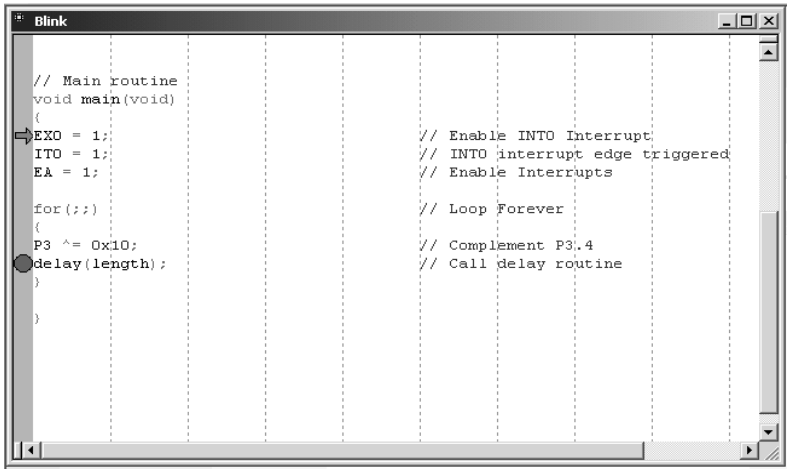
2.4.1 Once the Program has been downloaded, the program can be executed by either pressing the "F5" function key or selecting "Run" from the "Run" toolbar menu or click on the → icon in the **Debug bar**. On selecting **Run**, the Display LD2 (LED) on the Evaluation Board should be seen flashing. The Rate at which LD2 flashes can be altered be pressing the INT0 switch (SW3).

2.4.2 To halt the program execution select the break option from the **Run** menu (Ctrl + F5) or click on the ■ icon in the **Debug bar**

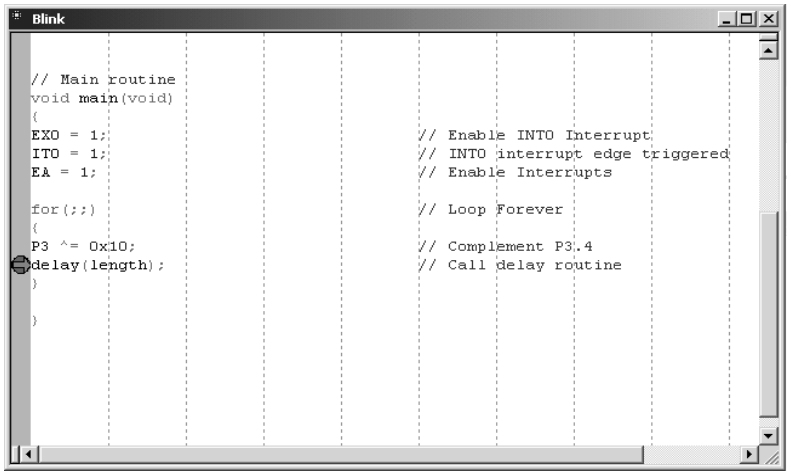
2.4.3 The projects found in the folder '\Examples' are mixed projects ie. contain both assembly and 'C' Modules. Set a breakpoint on the **delay (length)** function call in main() in the source file Blink.c . This is done by right clicking on the line of code in the editor window and then selecting **"Insert breakpoint"** as shown here.



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



Press the Run icon and the code executes and halts at the breakpoint indicated by the blue arrow at the breakpoint.



To step-Into the **delay()** function press.

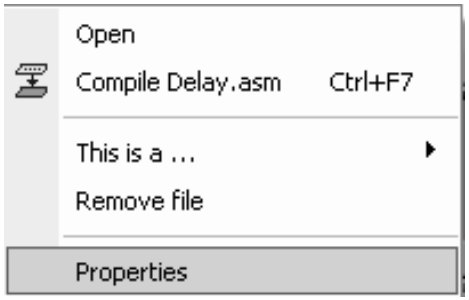
To step-Out of the **delay()** function press

To step-Over the **delay()** function press

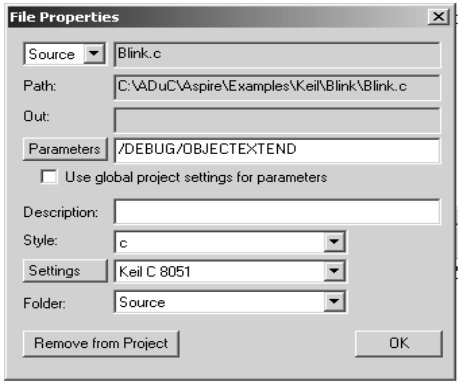
Run To Cursor by selecting a source line in the editor window and pressing

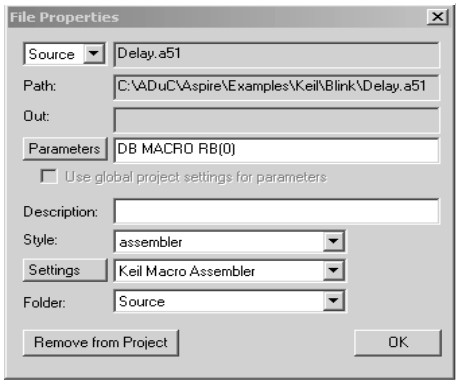
2.5 File Properties of The Example Project

2.5.1 The projects found in the folder '\Examples' are mixed projects ie. contain both assembly and 'C' Modules. The Properties of the project files can be viewed and edited in the following manner. Right click on the source file in the workspace window and select properties in the context window as shown below.



Shown below are the file properties set for each of the modules in the Example project.





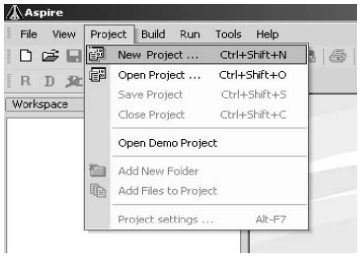
2.5.2 To save your project click on the save button on the toolbar or go to the Project menu and select save project.

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

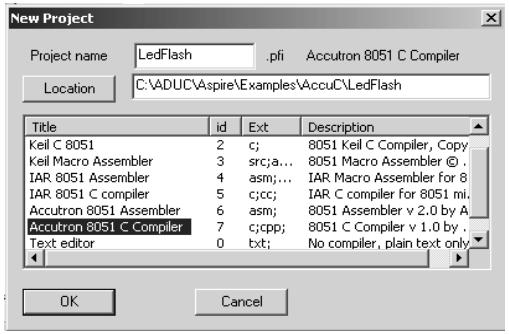
accutron supplying innovative solutions
QuickStart™
Get Started Guide

2.6 Starting a Project

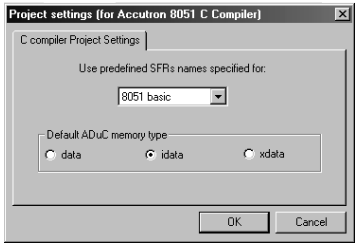
2.6.1 From the **Project** menu select **New Project** as shown below.



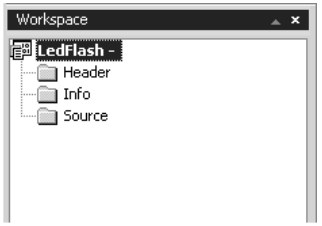
2.6.2 Enter the name of the project as **LedFlash** and chose the '**Accutron 8051 C Compiler'** (AccuC). Specify the directory for the project files as shown below.



On selecting OK the user will be asked to configure the project settings as shown below.

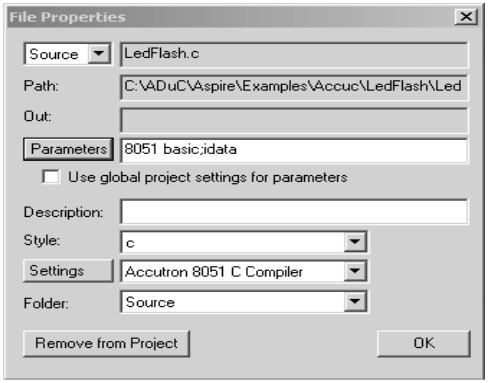


2.6.3 The new project will take the following structure. The Header folder is where any include files are placed. The Info folder is used to allow the user keep track of project status and for general note taking. The Source folder is where the source files for the project are placed.



2.6.4 Right Click on the source folder and select "**Add Files to Folder**". You can select the file \Examples\Accuc\LedFlash\Ledflash.c ,or enter the name of a new file you wish to create.

2.6.5 If the source file does not open automatically double click on the file in the workspace window. This file should automatically be configured as a 'C' file with the **Accutron 8051 C Compiler** as its compiler. This can be seen by right clicking on the ledflash.c file in the source folder and selecting properties.

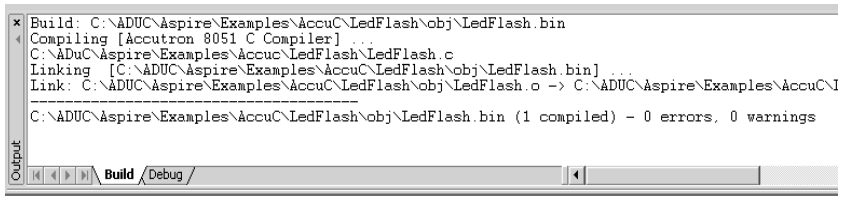


Note: You must ensure any C or assembly files you add to your projects have correctly configured properties. Failure to ensure this will result in failure to build your projects

accutron supplying innovative solutions
QuickStart+
Get Started Guide

2.7 Compiling Code

2.7.1 To compile select '**Compile ledflash.c**' from the **Run** menu.



2.7.2 If there are errors in your source code this 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.

Note: For further information on the accutron 'C' Compiler AccuC refer to the HTML help file **'AccuC Guide'** which is located on the default install directory. '**\ADUC\Aspire'**

2.8 Building the Code

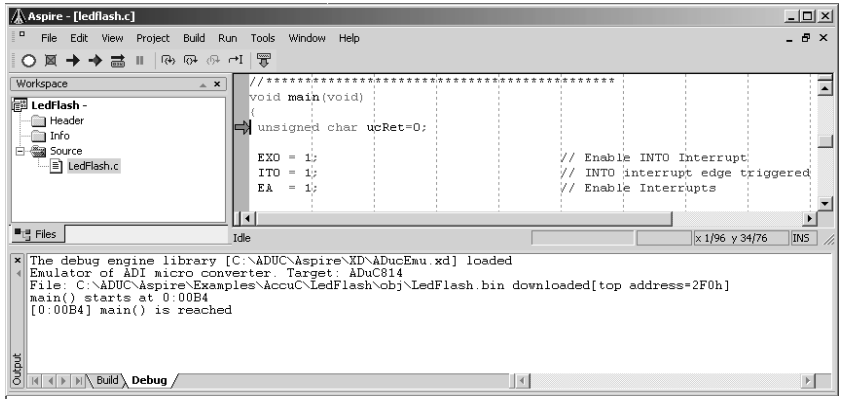
2.8.1 Before the code can be downloaded to the ADuC target the entire project must be built. This is done by clicking on the icon on the toolbar or by selecting '**Build**' from the '**Build**' menu.

2.9 Downloading/Debugging Code

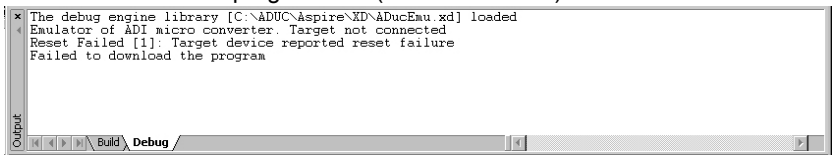
Note: Debug Control options are available in the **Debug bar**. The Debug Bar can be opened by selecting the **Bars** option from the **View** menu and clicking on **Debug bar**.

2.9.1 To place the ADuC device into download/debug mode the reset button must be pressed while the debug button is held down on the evaluation board. (see 1.3.4 & 1.3.5)

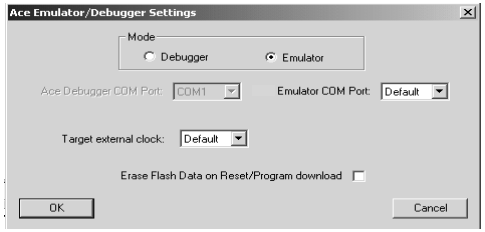
2.9.2 To download code onto the target device select '**Load Program**' from the **Run** menu, press the icon in the **Debug bar** or press F12.



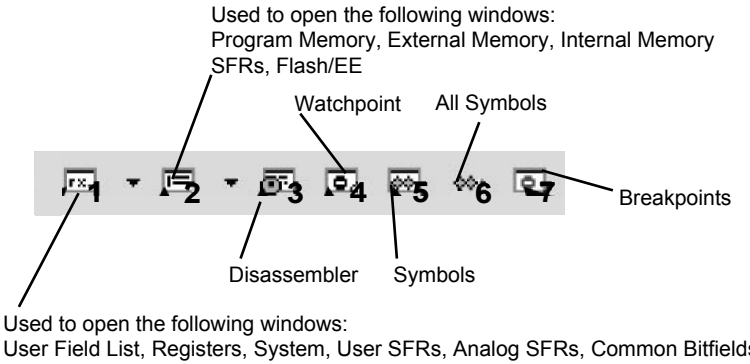
The last message in the status window should state that 'main() is reached' and a blue arrow at the first line of main() in this case on the line '**unsigned char ucRet=0;**'. If the status window indicates that the device was not programmed (as shown below)



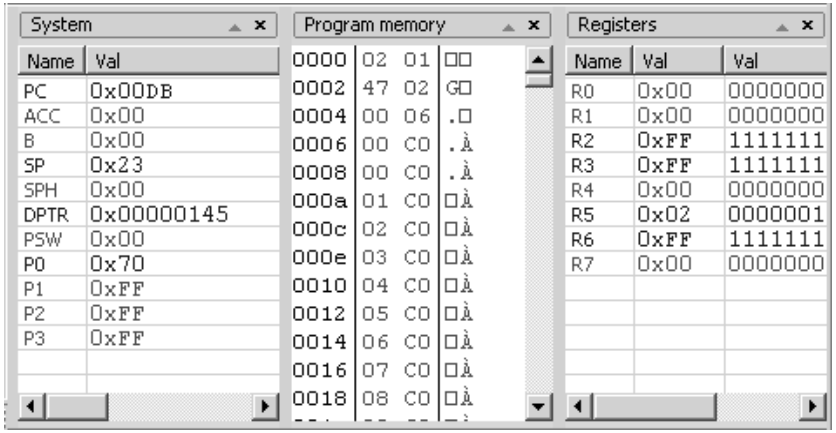
1. Try placing the device in Download\Debug mode (2.9.1)
2. Try checking the '**ADuC Hardware Emulator setup**' under the **Run** menu. This should be set up to use the Emulator and the serial port of your system to which the Emulator POD is connected.



Note: Open the **DebugView bar** by selecting the **Bars** option in the **View** menu and clicking on **DebugView bar**.



2.9.4 Open the following windows, Program Memory, System and Registers. These dockable windows by default appear on the right hand side of the code. These windows can be dragged around the workspace, to return them to their original location double click on the window name. Arrange these windows as shown.



Note: A register or memory location that appears in green has been updated and hence the values accurately represent the values currently on the chip. Any registers or memory that appears in red has not been uploaded. The configuration of what data is uploaded from the chip is determined in the "**Select uploadable areas / SFRs**" window under the **Run** menu. If after a breakpoint or a step, any of the registers or memory locations that are uploaded change from their previous value they will appear in blue.

2.9.5 Set a breakpoint on the '**P3_4^=0x1;**' instruction. This is done by right clicking on the line of code and then selecting "**Insert breakpoint**". Note that the breakpoint is indicated by a large red dot to the left of the line.

2.9.6 Press the run code icon . The code is executed and halts at the breakpoint. Press the step over icon . Notice that the value of the P3 register changes in the system window.

2.9.7 When code is halted the values of the SFRs and memory locations can be changed. To change the value of the P3 SFR right click on the value of P3. This will allow you to type in a new value. Insert the value 0xFF, and update the value by deselecting the field (left click anywhere outside this field). Download this data to the target device by clicking on the Download Data icon . This updates all SFRs changed by the user. The LD2 on the evaluation board should be turned off.

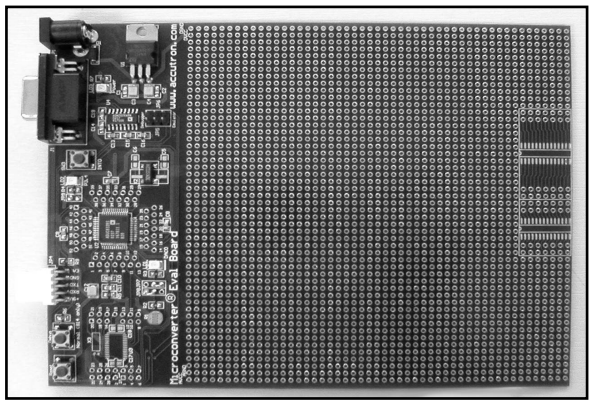
2.9.8 To resume code execution, press the run code icon again. The code will resume execution and will halt when it reaches the breakpoint. To clear the breakpoint right click on the breakpoint and select '**Remove breakpoint**'.

2.10 Saving/Closing a Project

2.10.1 To save your project click on the save button on the toolbar or go to the Project menu and select save project.

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

3. Evaluation Board Reference Guide



3.1 Prototyping Board Overview

The MicroConverter Prototyping Board has the following features:

- 2 layer PCB.
- On board regulated 5V power supply (see section 3.2.1)
- RS232 interface to MicroConverter via 9-way D-type connector (see section 3.2.2)
- All MicroConverter Ports are brought out to holes to ease access (see section 3.4)
- Large prototyping area
- Push button switches for reset, debug and interrupt (see section 3.2).
- Power indicator and general purpose LEDs. (see section 3.2)
- Light Dependant resistor demonstration circuit (see section 3.5)

3.2 Prototyping Board Features

3.2.1 Power Supply

The 9V supply is fed to the board via the 2.1mm input power socket (JP1). The Power Supply is not required if the Emulator POD is used (see 1.3). The Prototyping Board derives its power through the 5-Way connector on the Emulator POD. **Do not supply power to both the Emulator POD and Prototyping Board at the same time.** The input connector is configured as 'CENTRE NEGATIVE' i.e. GND on the centre pin and +9V on the outer shield. The 9V supply is regulated via a linear voltage regulator (U1), the 5V output being used to drive the digital side of the board directly. The 5V supply is filtered and then used to supply the analog side of the board. When on, the LED (LD3) indicates that a valid 5V supply is being generated by the regulator. All analog supplies are decoupled with 10uF and 0.1uF at device supply pins. Digital supplies are decoupled with 0.1uF at the digital supply pins.

3.2.2 RS232 interface

The TXD and RXD (pins 17 and 16 respectively) lines are connected to the centre pins of JP5 and JP6 respectively. When the jumpers are in Debug position these lines connect through an RS232 transceiver (U2) to the external 9-way D-Type connector (J1). The transceiver generates the required level shifting to allow direct connection to a PC serial port. A standard serial port cable is included as part of the QuickStart+ Development System to connect the PC directly to the evaluation board. When the jumpers are in the emulation position, the TXD and RXD are connected directly to JP4.

3.2.3 Connections

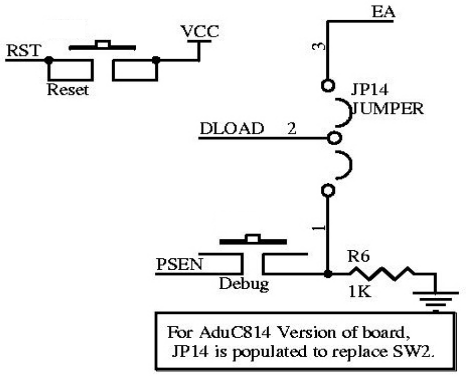
All of the MicroConverter pins are brought to plated through holes for easy access.

3.2.4 Reset/Debug/INT0 Push Buttons

A Reset push button switch is provided to allow the MicroConverter to be manually reset. This switch is tied directly to the reset pin of the MicroConverter. A Debug push button switch is tied directly to the PSEN pin of the MicroConverter. This applies to all MicroConverters except the ADuC 814. The Debug switch in conjunction with reset switch allows the MicroConvertor to be placed into serial download, serial debug or emulation mode. To place the MicroConverter in Serial Download/Serial Debug/Emulation mode simply press and hold the Debug switch, then press and release the Reset switch and then release the debug switch. To apply a normal reset to the MicroConverter simply press and release the reset switch. When the board is populated with an ADuC 814 device, the Debug switch is replaced with a jumper JP 14.

When the jumper is placed in Debug position it enables Debug / Serial Download mode on power-up or RESET by being pulled high through a resistor. When the jumper is placed in Normal position the user code is executed by this pin being pulled low on power-up or RESET.

A push button switch (SW3) is tied between the MicroConverter interrupt O pin (P3.2) and ground. This allows an external interrupt to be generated easily



Reset and Debug push buttons

3.2.5 Power Indicator/General Purpose LEDs

A red power LED (LD3) lights when the board receives the proper supply voltage
A LED (LD2) is connected to P3.4 of the MicroConverter through link JP9, remove link to disconnect
A LED (LD1) is connected to DAC0 to the MicroConverter through link JP7, Remove link to disconnect.

3.3 Link Options

JP5 & JP6 connects RXD and TXD of the MicroConverter to RS232 interface chip in debugger position and directly to JP4 in Emulator position.

Emulator Debugger Header

JP7 connects LED (LD1) to DACO of the MicroConverter
JP8 connects LDR circuit to ADC of the MicroConverter
JP9 connects LED (LD2) to P3.4 of the MicroConverter

3.4 External Connections

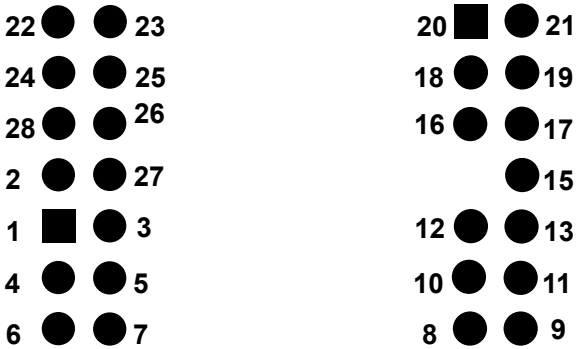
3.4.1 J1 Serial Interface Connector

J1 provides a simple connection of the Prototyping Board to the PC via a PC serial port cable.
(provided with the MicroConverter QuickStart+ Development System).

3.4.2 JP1 Power Supply Connector

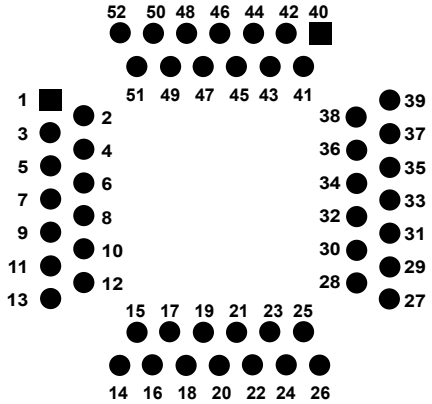
JP1 allows for the connection between the Prototyping Board and the 9V power supply provided in the MicroConverter QuickStart+ Development System. Note: When the Emulation POD is used the Prototyping Board derives its power from the POD

3.4.3 ADuC 814 TSSOP Connections



Pin Connections to ADuC 814 TSSOP package.
Note: Pin 14 is not made available because pins 14 and 15 are AGND.

3.4.4 ADuC PQFP Connections

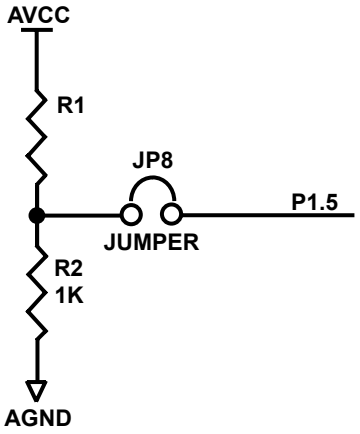


Pin Connections to all ADuC PQFP packages

3.5 Demonstration Circuits

3.5.1 LDR Circuitry

As can be seen from examining the schematic, an example LDR circuit is connected in a potential divider configuration as shown here.
Remove link to disconnect from MicroConverter

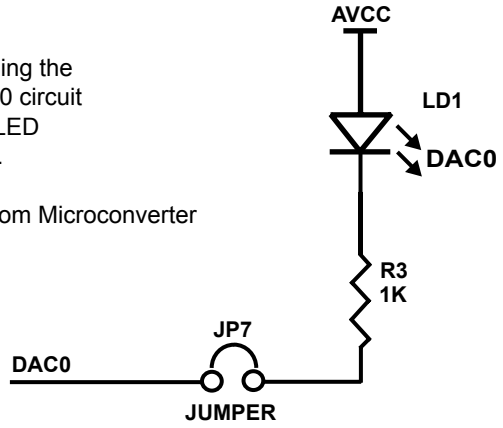


Circuit diagram of the LDR Circuit

3.5.2 DAC Circuitry

As can be seen from examining the schematic, an example DAC0 circuit is connected with a resistor LED configuration as shown here.

Remove link to disconnect from Microconverter



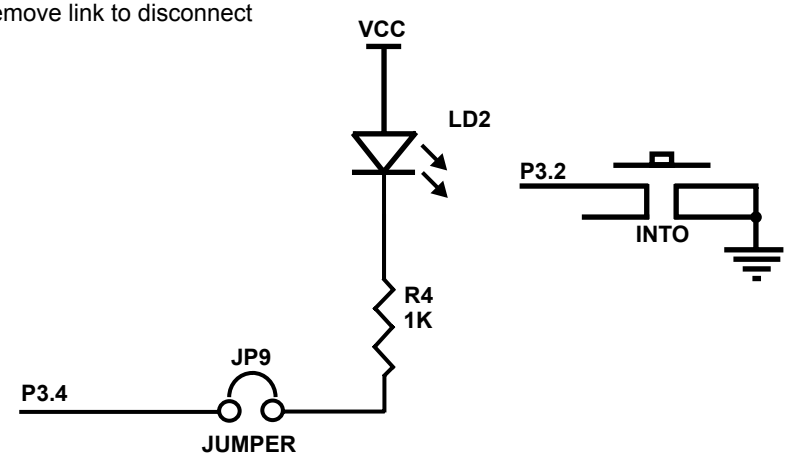
Circuit diagram of the DAC0 Circuit

accutron supplying innovative solutions
QuickStart+
Get Started Guide

3.5.3 BLINK Circuitry

P3.4 is connected to a resistor LED configuration and P3.2 (INT0) to a push button switch which are utilised by the blink program.

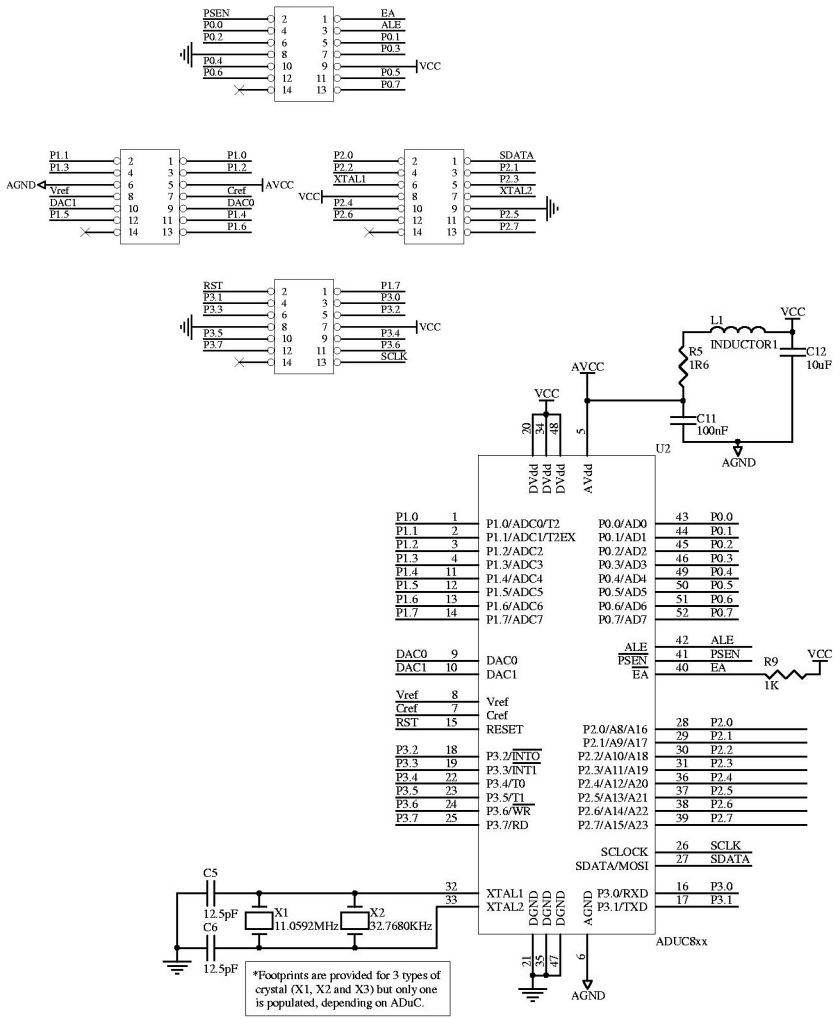
Remove link to disconnect



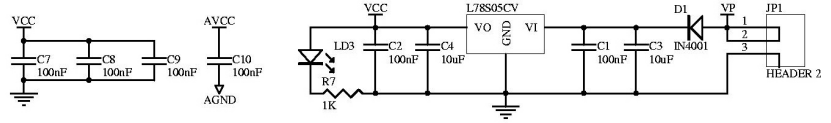
Circuit diagram of the BLINK Circuit

3.6 Prototyping Board Schematics

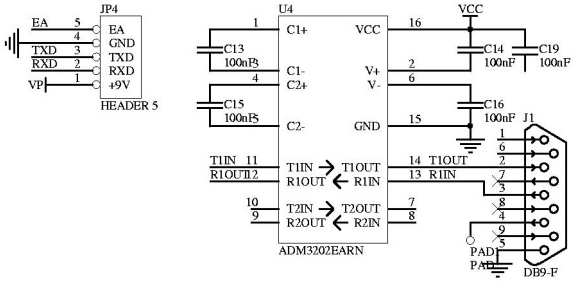
ADuC PQFP Pins



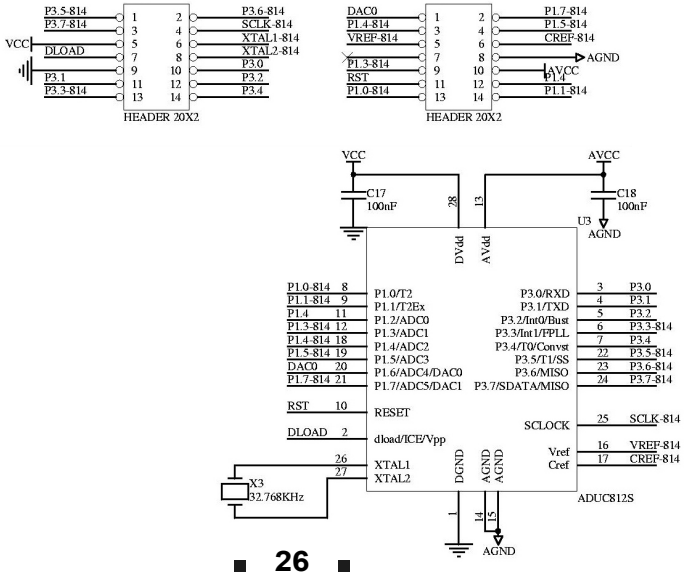
Power Supply

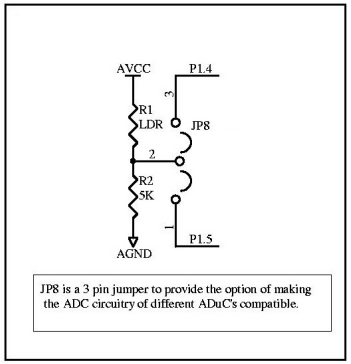


RS232

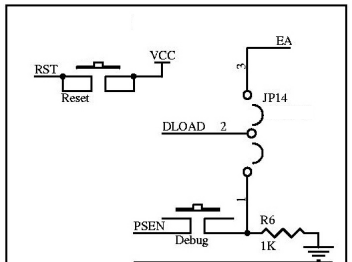


ADuC 814 TSSOP Pins

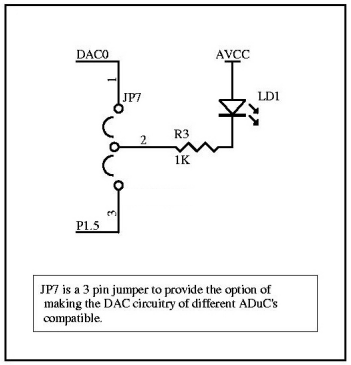




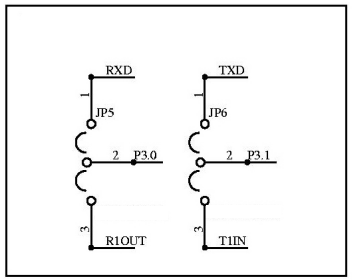
JP8 is a 3 pin jumper to provide the option of making the ADC circuitry of different ADuC's compatible.

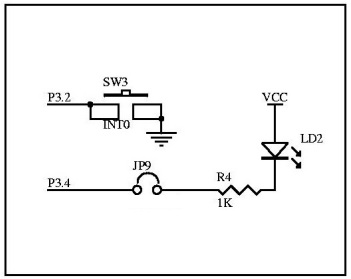


For AduC814 Version of board, JP14 is populated to replace SW2.



JP7 is a 3 pin jumper to provide the option of making the DAC circuitry of different ADuC's compatible.





Notes