

ADuC8XX UART Emulation

Revision Number: V1.0

Last Revision Date: 2008-01-17

Analog Devices Inc. has the full intelligent property (IP) of this document and the things described in this document.

Analog Devices Inc. has the right to change any of the descriptions in the document without notifying the readers.

If the readers need any technical help, contact China Applications Support Team (CAST) via <u>china.support@analog.com</u> or the toll-free number 800 810 1742.

Author: David Guo (david.guo@analog.com)

Department: China Applications Support Team (CAST)

Revision History

Date	Revision Description	Who
2008-01-17	Draft	David

Background:

Many customers have puzzles about the application of the ADuC8XX UART Emulation. This document is a simplified document about the correct operation. One thing needed to be noticed is that: debugging by UART is a simple Method of evaluation, its function is very limited. For example, the UART which is used to communicate with PC and Timer 1 can't be used in the code. Also there are some other limitations, you will find more in the below descriptions.

So we suggest you use the JTAG Emulator to debug the code.

Conditions:

Hardware: EVAL-ADuC841

Software: uVison3 V3.53

(This is a Evaluation Software, can be downloaded at <u>https://www.keil.com/c51/demo/eval/c51.htm</u>. The version limits the code size 2K

)

Step:

1. Connect the PCB and PC by the serial cable, power on the hardware board; open the Keil Software and set up a project, and add your c file into the project.

2. Configure relative parameters.

1). Open the "Options for Target", as Figure 1 shown, select the type of MCU in your PCB;



Figure 1

2). Configure the debug parameter. You should operate according to Figure 2. Pay attention to the places denoted by red lines.

Options for Target 'Target 1'	
Device Target Output Listing User C51 A51 C Use Simulator Settings Limit Speed to Real-Time	BL51 Locate BL51 Misc Debug Utilities
Load Application at Startup Run to main() Initialization File: Edit	Load Application at Startup Initialization File:
Restore Debug Session Settings Image: Breakpoints Image: Breakpoints	Restore Debug Session Settings
CPU DLL: Parameter: S8051.DLL	Driver DLL: Parameter: S8051.DLL
Dialog DLL: Parameter: DP51.DLL pAD841	Dialog DLL: Parameter: TP51.DLL pAD841
ОК Са	ncel Defaults Help

Figure 2

3. Translate and Build the target as Figure 3 shown.



Figure 3

4. Now we can download the program and emulate the code. Before downloading, you need to press the Reset Key in your PCB. After the hardware reset, you can put down the "Debug" button as Figure 4 displays. Then you can enter into the debug mode.



Figure 4

5. As Figure 5 displays, we can only debug the code in the Disassembly window. The yellow arrow is the beginning of program.

👿 test_841 - 費 ision3 - [Dis	sassembly]			
Eile Edit View Project Debu	ug Fl <u>a</u> sh Pe <u>r</u> ipherals	Tools SVCS W	indow <u>H</u> elp	
🎦 😂 🖬 🎒 🕺 🖻 🕄	오오 建建	16 % % %	9	
👫 🖹 🔕 🔁 🔂 🖓 🕬	♦ ½± 0±	💭 🖤 🔰 💷	E	a >
Project Workspace 🔺 🗙	16: vo:	id main (voi	id)	
🖃 🔁 Target 1	17: {			
🗄 📇 Source Group 1	18:			
START_AD.A51	19:	PLLCON	$1 = 0 \times 00$);
🗄 🔝 test_841.c	C:0x0800	E4 0	CLR	A
	C:0x0801	F5D7 M	10V	PLLCON(0xD7),A
	20:	IT0 =	1;	
	C:0x0803	D288 5	SETB	IT0(0x88.0)
	21:	EX0 = 1;		
	22:			
	C:0x0805	D2A8 3	SETB	EX0(0xA8.0)
	23:	P3 = 0	OXFF;	
	C:0x0807	75BOFF N	10V	P3(0xB0),#0xFF
	24:	i = 7;		
	25: //	while	(1)	
	C:0x080A	F508 N	10V	1 (0x08), A
	C:0x080C	750907 1	100	0x09,#0x07
	26:	IOT(1=	=0;1<600 4077	(0:08) N
	C:0x080F	F500 P	107	1 (0X08),A
	C:0x0811	1509 1	4017	DC 5 (0+09)
	C:0x0815	AEUS P	1017	D7 0v00
	C:0x0817	FF N	1017	A P6
	C:0x0818	33 1	and a	A, NO
	C:0x0819	95E0 S	SUBB	A. ACC (0xE0)
	C:0x081B	FD N	10V	R5.A
	<			,
		D	a -	
Symbols • ×	test_841.c	START_AD	💦 Disasser	nbly



1). You can set the breakpoint at places where you hope the program stops. You can press the right key of the mouse at the place like Figure 6, and click the "Insert/Remove Breakpoint", then there will be a red point at the head of the current row.

2). Also, you can observe the variables you are interested to. In the "View" Menu, click the "Watch & Call Stack Window", then the Window like Figure 7 appears. You can input the names of the variables in the window.





Figure 6

Figure 7

3). Now we can run the program step by step. Click the button as Figure 8 shown.





6. In the Figure 9, the yellow arrow denotes the current place of program. You can observe the change of the variable's value. Here the IT0 is 1, the variable "i" is 7.

17: {			
18:			
19:	PLL	CON = 0x(10;
C:0x0800	E4	CLR	A
C:0x0801	F5D7	MOV	PLLCON(0xD7), A
20:	ITO	= 1;	
C:0x0803	D288	SETB	IT0(0x88.0)
21:	EX0 = 1	;	
22:			
C:0x0805	D2A8	SETB	EX0(0xA8.0)
23:	P3	= OXFF;	
C:0x0807	75B0FF	MOV	P3(0xB0),#0xFF
24:	i =	7;	
25: //	whi	.1e(1)	
C:0x080A	F508	MOV	1 (0x08), A
C:0x080C	750907	MOV	0x09,#0x07
26:	for	(1=0;1<60	(1000) 1++)
C:0x080F	1508	MOV	1 (UXUS), A
C:0x0811	1509	MON	
C:0x0813	ALUS	MON	R0,1 (UXU0)
C:0x0815	ALO3	MOV	
C:0x0817	22	PLC	A, KO 7
C:0x0810	0580	CIIDD	A 7 NCC (0++F0)
C:0x0813	FD	MON	
C:0x0816	FC	MOV	
<	10		×.,,*
🖹 test_841.c	START_AD	🙉 Disass	embly
ead record\\A	ADUC\\ADu	C8xx\\Te	stCode\\ADuC841\\test_841"
			ТТО 1
			P3 0xFF
			ז 0x0007
			dype F2 to edit>

Figure 9

7. For UART Emulation, you can't set breakpoints in a cycle, for example, the "for" cycle and the "while" cycle. And you can't stop the program when the program is running, if you try to stop the program, then a warning happens as Figure 10.

ADuC8xx/ADE7xxx Devices cannot be stopped in the Debugger when connected to a serial interface. This is only possible with a 1-pin POD!				
Continue Return to running application.				
Stop Debugging Stops debugging session				
Note:				
As long as the running application does not reach a breakpoint it cannot be stopped. Now you can wait until a previously set breakpoint is reached or you need to stop the current debug session.				