



EE Dump Analyzer

User Guide

v1.03



Document Conventions

	Note: Provides related information or information of special importance.
 Caution	Caution: Indicates potential damage to hardware or software, or loss of data.
	Warning: Indicates a risk of personal injury.

Document Status

Doc Status: User Document Template	Technical Publication: v1.03
------------------------------------	------------------------------

For more information, visit our website at: www.asrmicro.com

Disclaimer

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of Asrmicro. ASR retains the right to make changes to this document at any time, without notice. ASR makes no warranty of any kind, expressed or implied, with regard to any information contained in this document, including, but not limited to, the implied warranties of merchantability or fitness for any particular purpose. Further, ASR does not warrant the accuracy or completeness of the information, text, graphics, or other items contained within this document.

ASR products are not designed for use in life-support equipment or applications that would cause a life-threatening situation if any such products failed. Do not use ASR products in these types of equipment or applications.

Copyright © 2020. ASR International Ltd. All rights reserved..

Table of Contents

1	PREFACE.....	5
1.1	FUNCTION.....	5
1.2	ABBS AND TERMS.....	5
2	INSTALLATION & USAGE.....	6
2.1	SYSTEM REQUIREMENTS.....	6
2.1.1	<i>HardWare.....</i>	<i>6</i>
2.1.2	<i>Operation System.....</i>	<i>6</i>
2.2	INSTALLATION.....	6
2.3	HOW TO USE.....	6
3	APPENDIX.....	11
3.1	REPORT ISSUES.....	11
3.2	UPDATE THIS DOCUMENT.....	11
3.3	REVISION HISTORY.....	11

1 Preface

1.1 Function

Accept the dumped files (`com_EE_Hbuf.bin`, `com_DDR_RW.bin` etc.) as input, automatically generate script files (`com_EE_Hbuf.cmm`, `PXA9xx_t32_simulator.cmm`) which contain register info and will be used by Lauterbach Trace32, also could generate text log which contains info about registers value, call stacks etc.

1.2 Abbs and terms

Abbs and Terms	Discription
EEDump	EE Dump Analyzer
Trace32	Lauterbach Trace32 tool
AP	Application Processor
CATStudio	Cellular Analysis & Test Studio
CP	Communication Processor
UE	Mobile phone, ASR DKB/EVB
ICL	ASR UE Log that saved by CATStduio

2 Installation & Usage

2.1 System Requirements

2.1.1 HardWare

Processor: Pentium IV >= 1GHz.

Memory: 512MB.

Hard Disk: 128MB.

2.1.2 Operation System

OS: MS Windows XP, Windows 7;

2.2 Installation

EE Dump Analyzer is a dll shipped with CATStudio.

Name	Description
EEDump.dll	Main function file of EE Dump Analyzer
Temp/EEDump/cmm/	Folder that contains the generated scripts for Trace32
Docs/	Folder that contains the user guide

Table 1

2.3 How to Use

Normally the using process could be as follows:

1. You should prepare **com_EE_Hbuf.bin** , **com_DDR_RW.bin** and **COM_DTCM.BIN** (When CP triggers ASSERT or Watchdog, AP will dump CP debug bins include **com_EE_Hbuf.bin** and **com_DDR_RW.bin**. **com_EE_Hbuf.bin** record task name, stack and registers values. **com_DDR_RW.bin** record all CP RW section data.)
 - a) **com_DDR_RW.bin** can be pulled used adb command or ACAT FlashExplorer at /data/com_EE_Hbuf.bin.

- b) **com_EE_Hbuf.bin** can be pulled used adb command or ACAT FlashExplorer at /NVM/com_EE_Hbuf.bin.
 - c) **COM_DTCM.BIN** can be pulled used adb command or ACAT FlashExplorer at /NVM/COM_DTCM.BIN.
2. If you want to see specific source code related with the core dump, you should also prepare source code and debug info file which is usually called .axf file (such as **Arbel_MICASE_REL.axf**). **You should virtualize the source code folder to a virtual hard disk.** For example, if source code is under **C:\R7_CP_1.50.000\mcu**, you should press “**Start->Run**” and type in “**cmd**”, then in “**cmd.exe**” window you should type in “**subst Q: C:\R7_CP_1.50.000\mcu**”, this operation would virtualize your source code folder to Hard disk **Q:**. Your .axf file path (such as **Arbel_MICASE_REL.axf**) should be **Q:\tavor\Arbel\bin\ Arbel_MICASE_REL.axf**. Notice that .axf file should match with source code, otherwise it will not produce correct result.
3. You should also prepare **Lauterbach Trace32 tool** to run the generated scripts.
4. When the above files are ready, click the “**Utility->AnalyzeCPDump**” menu in CATStudio. It will pop out the main GUI (Figure 1)

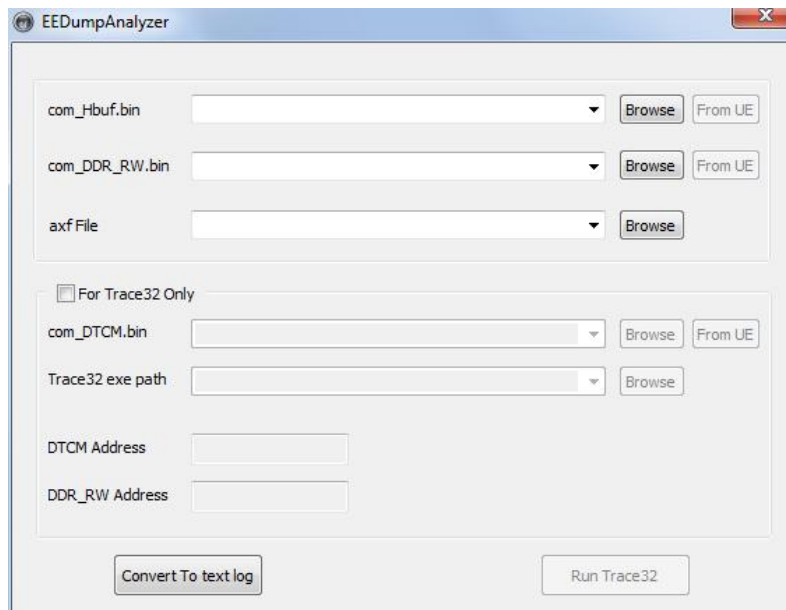


Figure 1

5. You could either click “**Convert To text log**” to generate text log or click “**Run Trace32**” to analyze the dump file by checking the “**For Trace32 only**” option or not. If “**For Trace32 only**” is not checked, only **com_Hbuf.bin**, **com_DDR_RW.bin** and **axf file** are mandatory (Figure 2) and the content of text log file looks like this (Figure 3). Otherwise, you should also provide **com_DTCM.bin**, Trace32 executable and fill in **DTCM** and **DDR_RW** address to analyze the CP dump files using Trace32. (Figure 4)

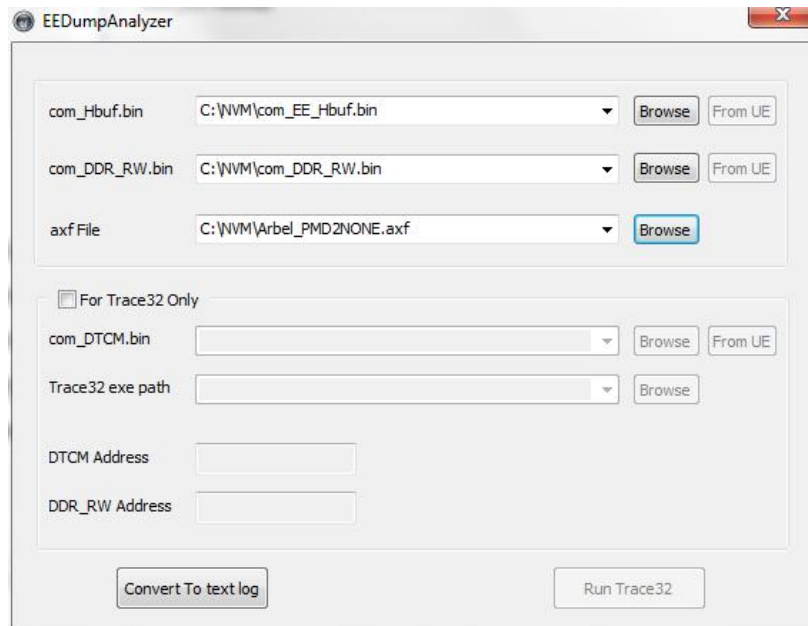


Figure 2

```

1 #SECTION_START -> Init
2 AXF_file = E:\testdata\CP_DUMP\test4cpdump\N22_LTG_SL.axf
3 DDR_RW_file = E:\testdata\CP_DUMP\test4cpdump\com_DDR_RW.bin
4 REG_file = E:\testdata\CP_DUMP\test4cpdump\com_EE_Hbuf.bin
5 DUMP_folder = E:\testdata\CP_DUMP\test4cpdump\
6 #SECTION_END -> Init
7
8 #SECTION_START -> call trace
9 0x06617EC0 ReadRegisterContext () [ utils_s : line=650 ] <<scope
10 0x06070A32 eeHandlerSaveDescFileLineAssert(unsigned char*,unsigned char*,unsigned short) (desc=0x06070E24 = ".
11 .....
12 0x063C862E utilsAssertFail(unsigned char*,unsigned char*,short,unsigned char) (cond=0x06070E24 = ".....
13 .....
14 0x0610E5E2 diagRxTask(void*) (argv=inactive void * (bits:32)) [ diag_rx_c : line=2728 ]
15 0x00004EE0 TCC_Task_Shell(void) () [ tcc_c : line=2279 ]
16 #SECTION_END -> call trace
17
18 #SECTION_START -> global variables
19 __CurrentExceptionLog [ at address 0x07859000 ] : struct { [::__CurrentExceptionLog]
20 logSignature: 22498
21 logChecksum: 61830
22 log: AT 0x07859004 [type: []struct ]
23 DBversionID: "UUUUUUUUUUUUUUUUUU"
24 }
25 dummyExt [ at address 0x00000000 ] : 3065427638 [::__dummyExt]
26 Saved_Registers [ at address 0x06890C28 ] : struct { [::__Saved_Registers]
27 r0: 109644840
28 r1: 101125648
29 r2: 1771
30 r3: 1
31 r4: 101125648
32 r5: 1771
33 r6: 101125668
34 r7: 109103820
35 r8: 109052008
36 r9: 976932268
37 r10: 109652880

```

Figure 3

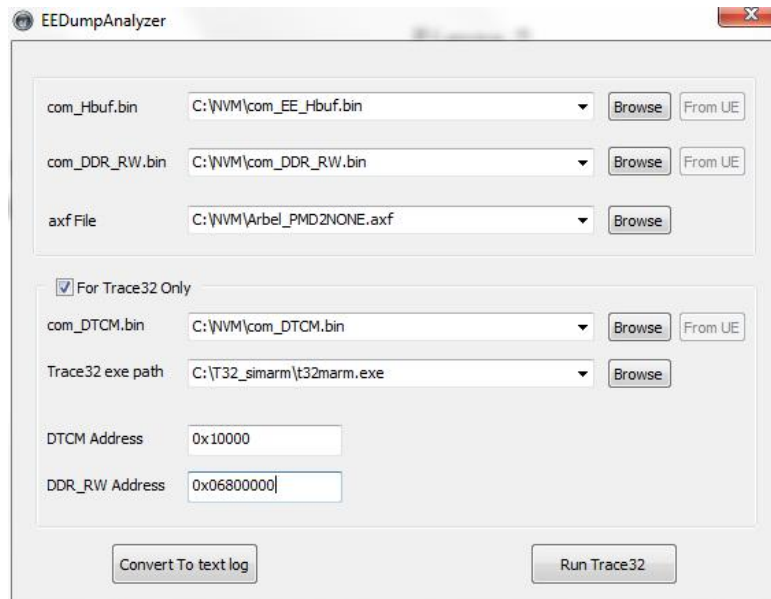


Figure 4

6. Fill in the correct address for loading **COM_DTCM.bin** and **com_DDR_RW.bin** (Figure 5). You can find the values of these two address in **map file** which you should find within the same folder as **axf file**. Usually you can get DTCM address at this line (Figure 6), and get **DDR_RW** address at this line (Figure 7).

EEDumpAnalyzer

com_Hbuf.bin C:\VVM\com_EE_Hbuf.bin Browse From UE

com_DDR_RW.bin C:\VVM\com_DDR_RW.bin Browse From UE

axf File C:\VVM\Arbel_PMD2NONE.axf Browse

☒ For Trace32 Only

com_DTCM.bin C:\VVM\com_DTCM.bin Browse From UE

Trace32 exe path C:\T32_simarm\t32marm.exe Browse

DTCM Address 0x10000

DDR_RW Address 0x06800000

Convert To text log Run Trace32

Figure 5

HLITG_SL_DKB_DSOS.map x

102640	0x0000403c	0x00000008	Ven	RO	261890	Veneer\$\$Code	anon\$\$obj.o
102641	0x00004044	0x00000008	Ven	RO	261891	Veneer\$\$Code	anon\$\$obj.o
102642	0x0000404c	0x00000080c	Code	RO	37834	.text	kiostim.o(common.lib)
102643	0x00004058	0x0000008ea	Code	RO	58378	.text	tcc.o(os-nu_xscale.lib)
102644	0x00005122	0x00000002	FAD	RO			
102645	0x00005144	0x000000160	Code	RO	61630	.text	i2c_tcm.o(hal-HAL.lib)
102646	0x000052a4	0x000000854	Code	RO	78940	.text	l1frint.o(gplc-GPLC.lib)
102647	0x00005af8	0x000000560	Code	RO	94119	.text	kiostq.o(common.lib)
102648	0x00006058	0x000000888	Code	RO	94269	.text	kiostmem.o(common.lib)
102649	0x000068e0	0x000000314	Code	RO	94667	.text	kiostlow.o(common.lib)
102650	0x00006bf4	0x000000200	Code	RO	94736	.text	kiostask.o(common.lib)
102651	0x00006d44	0x0000008dc	Code	RO	95170	.text	kmdynmem.o(common.lib)
102652	0x000076d0	0x000000138	Code	RO	6762	COMMONFMI	utils_memRetain.o(hal-HAL.lib)
102653	0x00007808	0x00000007c	Code	RO	8202	ISSUE_WFI	commpm.o(hal-HAL.lib)
102654	0x00007884	0x000000028	Code	RO	60441	IrqInt	irq.o(hal-HAL.lib)
102655	0x000078ac	0x000000154	Code	RO	58124	TMT_Code	tmt.o(os-nu_xscale.lib)
102656							
102657							
102658	Execution Region DDR_DTCM (Base: 0x00010000, Size: 0x00005208, Max: 0x00008000, ABSOLUTE)						
102659							
102660	Base Addr	Size	Type	Attr	Idx	E Section Name	Object
102661							
102662	0x00010000	0x0000003c	Data	RO	7210	.constdata	timer.o(hal-HAL.lib)
102663	0x0001003c	0x00000008e	Data	RO	28992	.constdata	plkmpelm.o(gplc-GPLC.lib)
102664	0x000100ca	0x000000002	FAD	RO			
102665	0x000100cc	0x00000001c	Data	RO	78942	.constdata	l1frint.o(gplc-GPLC.lib)

Figure 6

HLITG_SL_DKB_DSOS.map x

102694	0x00017ffc	0x00000004	Zero	RW	1	DDR_DTCM_ENDMARKER.bss	anon\$\$obj.o
102695							
102696							
102697	Execution Region DDR_RELIABLE_RO_AREA (Base: 0x068e0000, Size: 0x0001ffff, Max: 0x0001ffff, ABSOLUTE, UNINIT)						
102698							
102699							
102700	Base Addr	Size	Type	Attr	Idx	E Section Name	Object
102701							
102702	0x068e0000	0x0001ffff	Zero	RW	2	DDR_RELIABLE_RO_AREA.bss	anon\$\$obj.o
102703							
102704							
102705	Execution Region DDR_RELIABLE_RO_AREA_ENDMARK (Base: 0x068ffffc, Size: 0x00000004, Max: 0x00000004, ABSOLUTE, UNINIT)						
102706							
102707							
102708	Base Addr	Size	Type	Attr	Idx	E Section Name	Object
102709							
102710	0x068ffffc	0x00000004	Zero	RW	3	DDR_RELIABLE_RO_AREA_ENDMARK.bss	anon\$\$obj.o
102711							
102712	Execution Region DDR_RW_DATA (Base: 0x06900000, Size: 0x0006a7a8, Max: 0xffffffff, ABSOLUTE)						
102713							
102714							
102715	Base Addr	Size	Type	Attr	Idx	E Section Name	Object
102716	0x06900000	0x0000000c	Data	RW	38	.data	ps_init.o
102717	0x0690000c	0x00000008	Data	RW	524	.data	ps_vmm.o
102718	0x06900014	0x00000008	Data	RW	684	.data	SKILLick.o

Figure 7

7. Now click “Run Trace32”. The Trace32 application should start and you could continue on your analyze work (Figure 8).

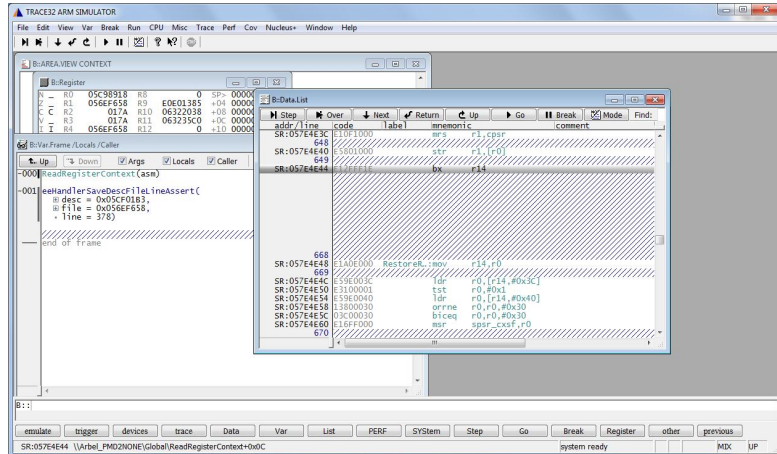


Figure 8

3 Appendix

3.1 Report issues

Please visit webs: <http://support.asrmicro.com>.

Or contact ASR FAE/AE

Or please send email to lianghuxu@asrmicro.com with “PC Tools User Doc Issue/Update: ...” title

Or submit an issue at CQ web: <http://sh2-cq01.asrmicro.com/cqweb/login>.

3.2 Update this document

Please send updated document to lianghuxu@asrmicro.com with “PC Tools User Doc Issue/Update: ...” title for merging, and you will receive merged latest version.

3.3 Revision History

Rev	Description	Date
1.00	Initial version	2013-07-26

1.01	Add the file location of com_Hbuf.bin / com_DDR_RW.bin / COM_DTCM.bin	2013-07-29
1.02	Add how to mount source code folder and set correct .axf file path	2013-07-30
1.03	1. Update the user guide according to the UI changes. 2. Add description about how to find DTCM and DDR_RW address. 3. Add description about the function "Convert to text log".	2014-08-29