

REALTEK

Debug Analyzer User Guide

V3.7

2024/04/10



REALTEK

Realtek Semiconductor Corp.

No.2, Innovation Road II, Hsinchu Science Park, Hsinchu 300, Taiwan

Tel.: +886-3-578-0211. Fax: +886-3-577-6047

www.realtek.com

COPYRIGHT

©2024 Realtek Semiconductor Corp. All rights reserved. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form or by any means without the written permission of Realtek Semiconductor Corp.

DISCLAIMER

Realtek provides this document ‘as is’, without warranty of any kind. Realtek may make improvements and/or changes in this document or in the product described in this document at any time. This document could include technical inaccuracies or typographical errors.

TRADEMARKS

Realtek is a trademark of Realtek Semiconductor Corporation. Other names mentioned in this document are trademarks/registered trademarks of their respective owners.

USING THIS DOCUMENT

This document is intended for the software engineer’s reference and provides detailed programming information. Though every effort has been made to ensure that this document is current and accurate, more information may have become available subsequent to the production of this guide.

ELECTROSTATIC DISCHARGE (ESD) WARNING

This product can be damaged by Electrostatic Discharge (ESD). When handling, care must be taken. Damage due to inappropriate handling is not covered by warranty.

Do not open the protective conductive packaging until you have read the following, and are at an approved anti-static workstation.

- Use an approved anti-static mat to cover your work surface
- Use a conductive wrist strap attached to a good earth ground
- Always discharge yourself by touching a grounded bare metal surface or approved anti-static mat before picking up an ESD-sensitive electronic component
- If working on a prototyping board, use a soldering iron or station that is marked as ESD-safe
- Always disconnect the microcontroller from the prototyping board when it is being worked on

Revision History

Date	Version	Comments	Author	Reviewer
2015/05/04	V1.0	First Version		
2017/03/02	V1.5	Add more Feature	Max	
2017/05/22	V1.7	Detail Amendment	Calvin	
2017/10/26	V2.1	Revision	Cordelia	
2022/08/17	V3.1	Revision	Cordelia	
2022/09/07	V3.2	Revision	Mengfei_Gao	
2023/04/18	V3.3	Revise format	Mengfei_Gao	
2023/05/12	V3.4	Add Disclaim	Mengfei_Gao	
2023/07/06	V3.5	Add Section 8.1	Mengfei_Gao	
2024/03/01	V3.6	Revision	Mengfei_Gao	
2024/04/10	V3.7	Add Show Timestamp	Mengfei_Gao	

Contents

1 Introduction	1
2 Overview of DebugAnalyzer Tool	2
3 Preparing.....	4
3.1 Real-time Log Capture and Decode.....	4
3.2 Local .bin Decode.....	4
4 Settings of DebugAnalyzer Tool	5
4.1 Settings of Main Dialog.....	5
4.1.1 Selections of Decoding Type	5
4.1.2 Settings of Decoder Output.....	5
4.1.3 Log Settings	8
4.2 Settings of Detail Settings Dialog.....	8
4.2.1 Settings of Serial Port	8
4.2.2 Settings of Decoder Output.....	9
4.2.3 Settings of Trace File Path.....	10
4.2.4 Swift Load .trace Files	10
4.2.5 Settings of Load/Save and Confirm/Cancel.....	11
5 Start and Stop Capturing and Decoding	12
6 Log Operation and Display on UI	13
6.1 Log Operation on UI.....	13
6.2 Log Display on UI	14
7 How to use Ellisys view package of BTSnoop.....	17
7.1 Real-time display of BTSnoop Log	17
7.2 Notes for Using Ellisys to view BTSnoop.....	19
8 Extended Usage	20
8.1 Core Dump Parser.....	20
8.2 Rename title of DebugAnalyzer Tool.....	23
8.3 View version of DebugAnalyzer Tool.....	23

Table List

Table 1-1 Supported ICs.....	1
Table 4-1 Decoder Input Selections.....	5
Table 4-2 Decoder Output Selections.....	6
Table 4-3 Log Settings.....	8
Table 4-4 Serial Port Settings.....	9
Table 4-5 Decoder Output Settings	9
Table 4-6 Trace Path Settings.....	10
Table 4-7 Decoder Output Settings	10
Table 4-8 Load/Save and Confirm/Cancel Settings.....	11
Table 8-1 Core Dump Parse	21
Table 8-2 Hard Fault Keywords	21

Figure List

Figure 2-1 Overview of Main Dialog	2
Figure 2-2 Overview of Detail Settings Dialog	3
Figure 3-1 Output Files.....	4
Figure 3-2 Local .bin Decode Output File.....	4
Figure 4-1 Log Decoding Type	5
Figure 4-2 Setting-Decoder Output	6
Figure 4-3 Decoder Output Settings.....	6
Figure 4-4 Decoder Output Settings From	6
Figure 4-5 Log Settings	8
Figure 4-6 Serial Port Settings.....	8
Figure 4-7 Decoder Output Settings.....	9
Figure 4-8 Trace File Path Settings	10
Figure 4-9 Swift Load .trace.....	10
Figure 4-10 Load/Save and Confirm/Cancel Settings	11
Figure 5-1 Select Bin File.....	12
Figure 6-1 Copy Logs.....	13
Figure 6-2 Show Log Details.....	13
Figure 6-3 Search and Filter Logs	14
Figure 6-4 Save Button.....	14
Figure 6-5 Show ARM and DSP Logs Left and Right.....	15
Figure 6-6 Show ARM and DSP Logs Up and Down.....	15
Figure 6-7 Show ARM Logs Only (DSP Logs Folded)	15
Figure 6-8 Show DSP Logs Only (ARM Logs folded)	16
Figure 7-1 Open Ellisys	17
Figure 7-2 Ellisys Configure	17
Figure 7-3 Start Capture	18
Figure 7-4 Open HCI Overview (Injection).....	18
Figure 7-5 HCI Overview (Injection).....	19
Figure 8-1 Core Dump Parse	20
Figure 8-2 RTL87x3E Symbol Information	22
Figure 8-3 TL87x3D Symbol Information	22
Figure 8-4 Rename Title.....	23
Figure 8-5 About Analyzer.....	23
Figure 8-6 View Version of Tool	23

1 Introduction

DebugAnalyzer is a tool used for capturing and decoding logs of Realtek Bluetooth SOC chip. The following ICs are supported. This document provides detailed guidance on how to set and use it.

Table 1-1 Supported ICs

No.	Supported ICs
1	RTL8762A
2	RTL8762C
3	RTL8762D
4	RTL8762E
5	RTL87x2G
6	RTL8752H
7	RTL8772F
8	RTL8763B
9	RTL8773B
10	RTL8763C
11	RTL8773C
12	RTL8763E
13	RTL8773E
14	RTL8753B

The document comprises the following chapters.

- **Chapter 1:** Introduction.
- **Chapter 2:** Provides overview of DebugAnalyzer.
- **Chapter 3:** Describes how to configure the tool and decode logs.
- **Chapter 4:** Describes how to operate decoded output (logs) on UI.
- **Chapter 5:** Describes how to view BT HCI log with Ellisys in real time.
- **Chapter 6:** Describes the extended functions included in DebugAnalyzer.

2 Overview of DebugAnalyzer Tool

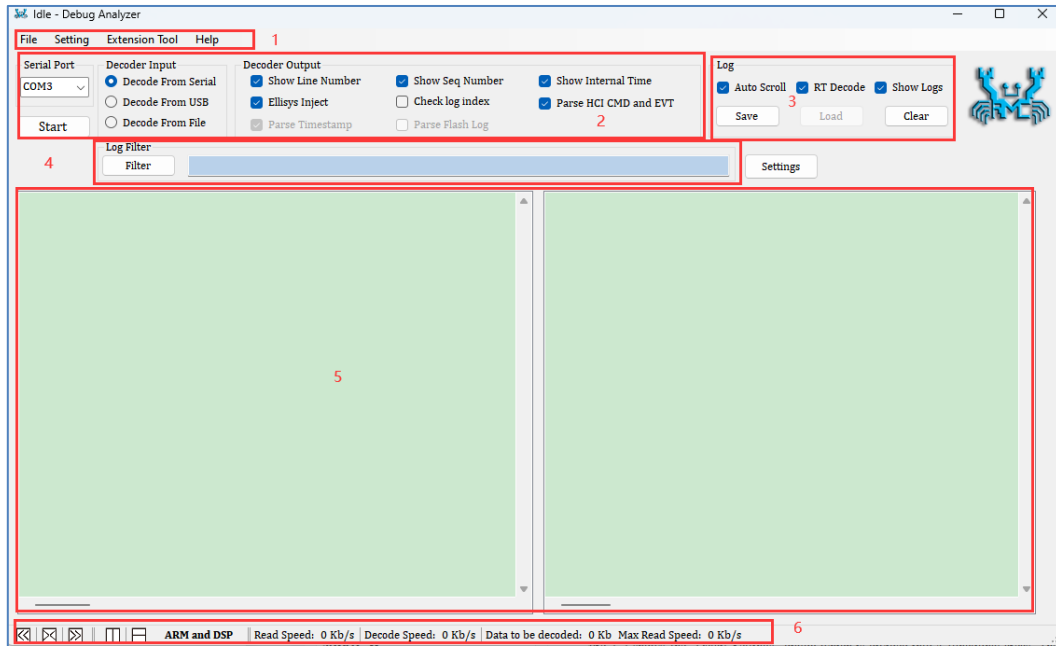


Figure 2-1 Overview of Main Dialog

Figure 2-1 is the main dialog of DebugAnalyzer, which is divided into 6 functional areas. The simple functions are described as follows.

- **Area 1:** Menu bar, contains extended settings and operations of DebugAnalyzer Tool.
- **Area 2:** Contains the optional parsing methods and parsing Log output settings.
- **Area 3:** Modify log display.
- **Area 4:** Filter logs by keywords.
- **Area 5:** Log display area, mainly includes the ARM area and the DSP area, which are used to show the ARM Log and DSP Log respectively.
- **Area 6:** Set the log display mode, and display the log reading and parsing speed.

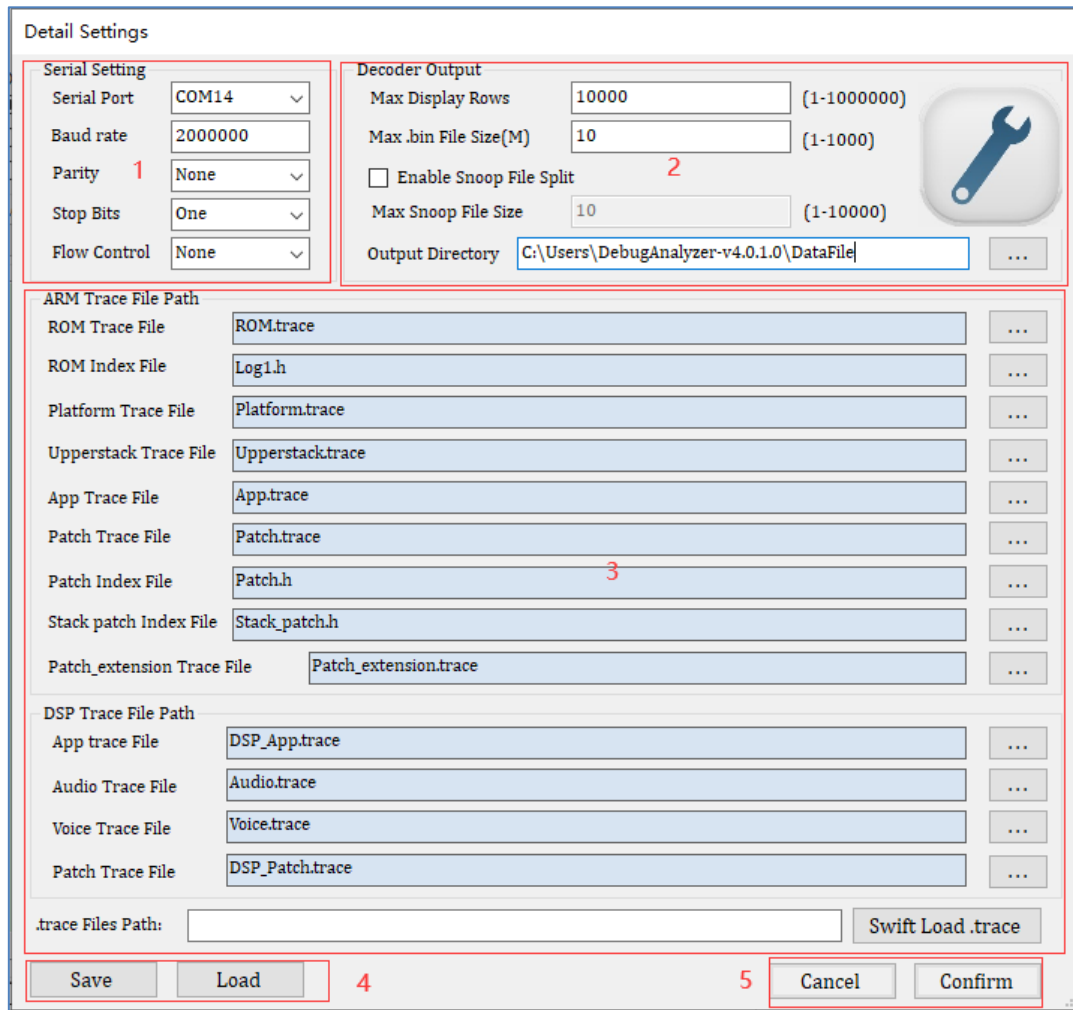


Figure 2-2 Overview of Detail Settings Dialog

The “Detail Settings” dialog is divided into 5 functional areas. The simple functions are described as follows.

- **Area 1:** Contains the changeable serial port settings.
- **Area 2:** The supplement of decoder output display and file save settings.
- **Area 3:** Set .trace file paths.
- **Area 4:** Save or load detail settings by .ini file.
- **Area 5:** Cancel or confirm current settings.

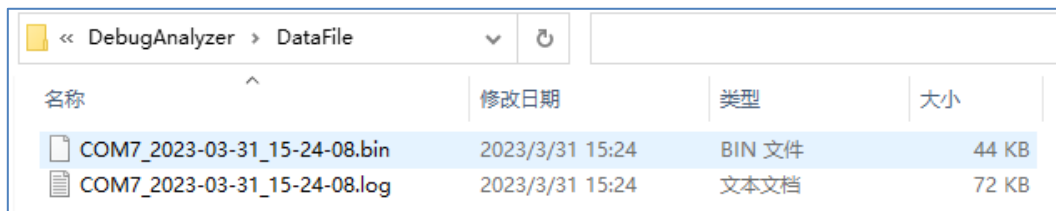
3 Preparing

3.1 Real-time Log Capture and Decode

If real time log capture is needed, connect PC and SoC chip via serial port. The raw data captured from SoC can be saved in a .bin file. It can be decoded to plaintext logs according to the .trace file set by the user, and these logs can then be saved in the .log.

Note: Please make sure that the .trace file matches the current SoC running code.

Figure 3-1 shows the **raw data file (.bin)**, **output log file (.log)**. The default save location is the DataFile folder in the same directory as DebugAnalyzer.exe.

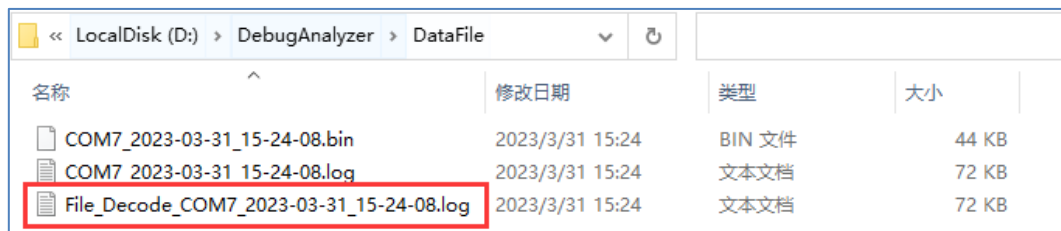


名称	修改日期	类型	大小
COM7_2023-03-31_15-24-08.bin	2023/3/31 15:24	BIN 文件	44 KB
COM7_2023-03-31_15-24-08.log	2023/3/31 15:24	文本文档	72 KB

Figure 3-1 Output Files

3.2 Local .bin Decode

- For local decoding, there is no need to connect PC and SoC chip, but .bin files which are being decoded need to be prepared.
- For example, use COM7_2023-03-31_15-24-08.bin in Figure 3-1, select Decode from File in Decode Input, refer to Figure 4-1 and Table 4-1.
- Wait for the parsing to complete, and users can find the result in the .log file with “File Decode_” prefix, as shown in Figure 3-2.



名称	修改日期	类型	大小
COM7_2023-03-31_15-24-08.bin	2023/3/31 15:24	BIN 文件	44 KB
COM7_2023-03-31_15-24-08.log	2023/3/31 15:24	文本文档	72 KB
File_Decode_COM7_2023-03-31_15-24-08.log	2023/3/31 15:24	文本文档	72 KB

Figure 3-2 Local .bin Decode Output File

4 Settings of DebugAnalyzer Tool

4.1 Settings of Main Dialog

4.1.1 Selections of Decoding Type

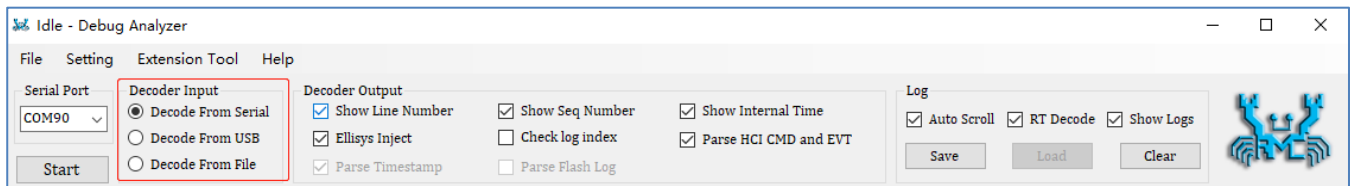


Figure 4-1 Log Decoding Type

The following table describes the specific meaning of each option in the “Decoder Input” part.

Table 4-1 Decoder Input Selections

Option	Selection	Description
Decoder Input	Decode From Serial	If selected, the raw data is captured from serial port. It is Real-time decode (RT Decode).
	Decode From USB	If selected, the raw data is captured from USB. It is Real-time decode (RT Decode).
	Decode From File	If selected, the raw data is retrieved from the .bin file on local disk. It is called local decoding. For example, if it is found that the wrong trace file is used, the user only needs to use the .bin file to reverse the solution, and there is no need to reproduce the problem again. Especially when the problem is difficult to reproduce, preserving this method of parsing is highly important. Description: Multiple .bin files can be selected at once.

4.1.2 Settings of Decoder Output

1. The “Decoder Output” area of the main page contains the commonly used parsing output parameter configurations (Figure 4-3).
2. All parameters can be configured on the “Decoder Output Settings” page (Figure 4-4).
3. The method of using the "Decoder Output Settings" page: Select "Setting-Decoder Output" from the menu bar (Figure 4-4).

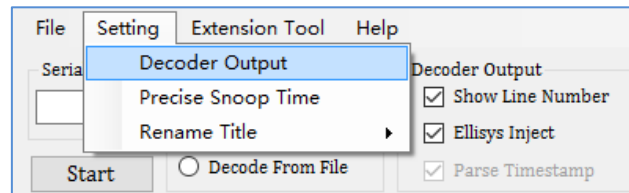


Figure 4-2 Setting-Decoder Output

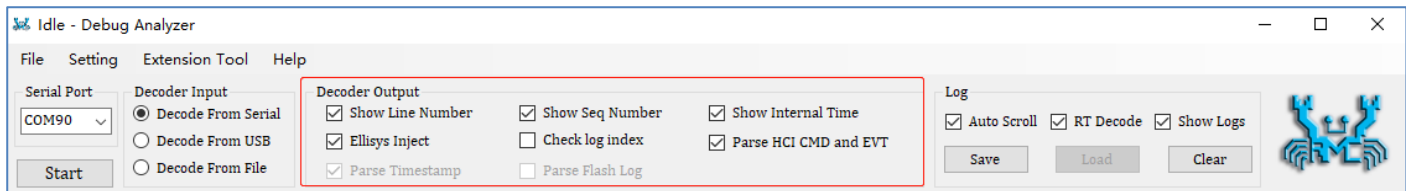


Figure 4-3 Decoder Output Settings

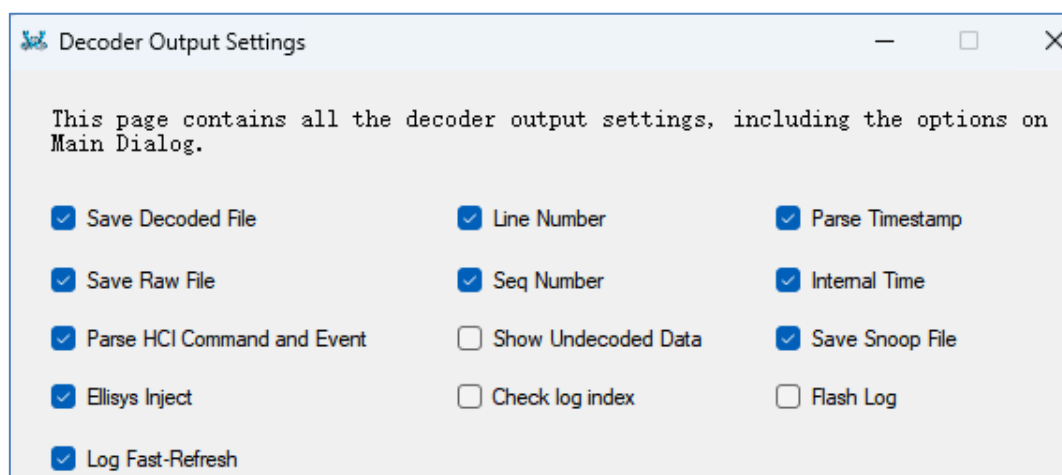


Figure 4-4 Decoder Output Settings From

The following table describes the specific meaning of each option on the Decoder Output Settings page.

Table 4-2 Decoder Output Selections

Option	Selection	Description
Decoder Output	Save Decoded File	If checked, the output logs are saved in a .log file. If not checked, the output logs are not saved in a .log file.
	Save Raw File	If checked, the raw data captured from bus (serial port/USB) is saved in .bin file.
	Parse HCI Command and Event	If checked, the tool parses HCI command and event and displays logs on the UI. If not checked, the tool does not parse the HCI command and event.
	Line Number	If checked, the "line number" is included in the output logs. e.g. <pre>0000011 08-18#10:28:55.970 013 00007 [OS] 0000012 08-18#10:28:55.970 014 00008 [OS] 0000013 08-18#10:28:55.970 015 00012 File</pre>
	Seq Number	If not checked, the "line number" is not included in the output logs. If checked, the "sequence number" is included in the output logs. e.g. <pre>0000011 08-18#10:28:55.970 013 00007 [OS] 0000012 08-18#10:28:55.970 014 00008 [OS] 0000013 08-18#10:28:55.970 015 00012 File</pre>

Option	Selection	Description
		<p>If not checked, the “sequence number” is not included in the output logs. The range of sequence number is 0~255. If one sets all required .trace and .h files in Detail Settings, the number should be consecutive.</p>
	Parse Timestamp	<p>When decoding from serial/USB, this item is always checked.</p> <p>When decoding from file, this item is changeable.</p> <p>If checked, the timestamp of the logs will be parsed.</p> <p>If not checked, the timestamp of the logs will not be parsed.</p> <ol style="list-style-type: none"> 1. The “timestamp” is added by the DebugAnalyzer Tool. 2. If the raw data is captured by the DebugAnalyzer, this item must be checked, otherwise, the decoded logs will be missing. 3. If the raw data is captured by other tool, this item should not be checked, or decoded logs will be missing.
	Internal Time	<p>If checked, “internal time” is included in the output logs. (Unit: ms)e.g. The “Internal Time” format before RTL87X3E.</p> <pre>0000011 08-18#10:28:55.970 013 00007 [OS] 0000012 08-18#10:28:55.970 014 00008 [OS] 0000013 08-18#10:28:55.970 015 00012 File[</pre> <p>The “Internal Time” format after RTL87X3E.</p> <pre>0000034 03-02#14:50:18.909 008 0000328.291 [BT 0000035 03-02#14:50:18.909 010 0000328.442 [BT 0000036 03-02#14:50:18.909 012 0000328.570 [BT</pre> <p>If not checked, “internal time” is not included in the output logs.</p> <p>Internal time is the timestamp provided by the BT SoC chip.</p> <p>Note: “internal time” is generated by SoC.</p>
	Show Uudecoded Data	<p>If checked, the raw data is appended by the decoded log. e.g.</p> <pre>0000002 08-18#10:28:55.345 000 35127 >>> Boot Patch (Raw Data: 7E-16-00-68-37-89-20-00-3E-3E-3E-20-42-6F-6F-74-20-50-61-74-63-68)</pre> <p>If not checked, the raw data is not appended to the decoded log.</p>
	Save Snoop File	<p>If the SoC chip provides HCI information to application, a .cfa file is generated when this button is checked. More details are provided in section 7.1. After version v3.0.0.10, this item is invalid.</p>
	Ellisys Inject	<p>If checked, HCI data will be sent to the application “Ellisys”.</p> <p>More details are provided in Chapter 7.</p>
	Parse Flash Log	<p>If checked, the log bin file from flash dump will be parsed.</p> <p>Only Decode From File mode can be checked.</p>
	Log Fast-Refresh	<p>If checked, the page refresh rate is 0.1s.</p> <p>If not checked, the page refresh rate is 1s (default).</p>
	Show Timestamp	<p>If checked, the timestamp of the logs will be showed on UI. e.g.</p> <pre>0000011 08-18#10:28:55.970 013 00007 [OS] 0000012 08-18#10:28:55.970 014 00008 [OS] 0000013 08-18#10:28:55.970 015 00012 File</pre> <p>If not checked, the timestamp of the logs will not be showed on UI.</p>

4.1.3 Log Settings

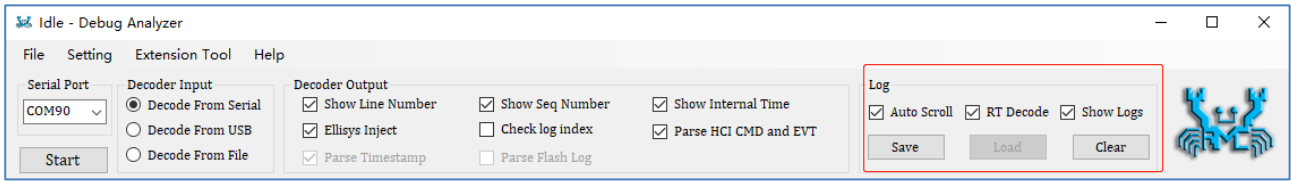


Figure 4-5 Log Settings

The following table describes the specific meaning of each option in the “Log” part.

Table 4-3 Log Settings

Option	Selection	Description
Log	Auto Scroll	If checked, the logs shown on the UI will scroll automatically during decoding. If not checked, the logs shown on the UI will not scroll automatically.
	RT Decode	If checked, the tool will capture and save the raw data, and then decode it. If not checked, the tool will capture and save the raw data but will not decode it.
	Show Logs	If checked, the tool will capture raw data, decode it, and display the logs on the UI. If not checked, the tool will capture and save the raw data but will not display the logs on the UI.

4.2 Settings of Detail Settings Dialog

4.2.1 Settings of Serial Port

If “Decode from Serial” is selected, make sure to configure “serial port (Figure 4-6)”.

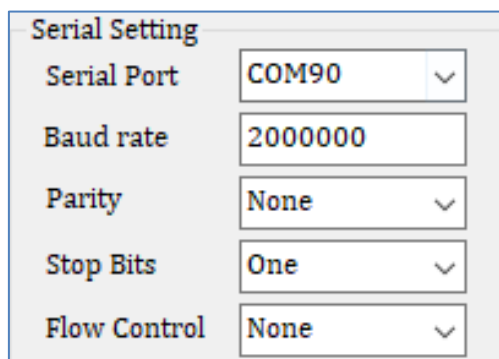


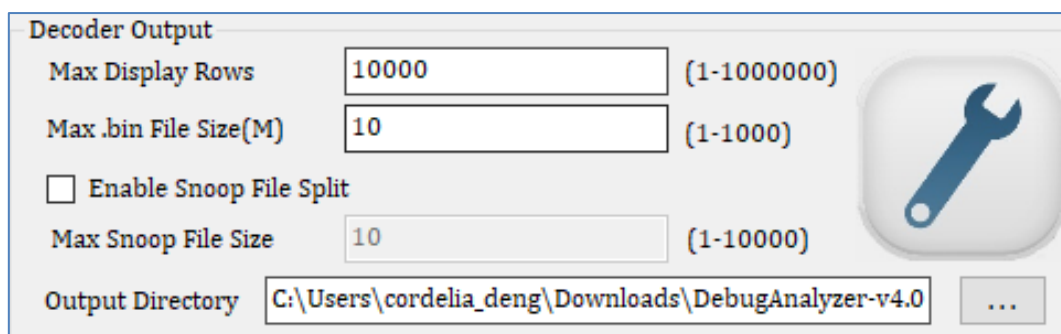
Figure 4-6 Serial Port Settings

The following table describes the specific meaning of each option in the “Serial Setting” part.

Table 4-4 Serial Port Settings

Area	Setting	Description
Serial Setting	Serial Port	Set serial port number.
	Baud rate	Set the baud rate of data transfer.
	Parity	Set the mode of parity.
	Stop Bits	Set the stop bits.
	Flow Control	Set the mode of flow control, generally, 'One' is chosen.

4.2.2 Settings of Decoder Output


Figure 4-7 Decoder Output Settings

The following table describes the specific meaning of each option in the “Decoder Output” part.

Table 4-5 Decoder Output Settings

Area	Setting	Description
Decoder Output	Max Display Rows	Set the maximum number of log rows to be displayed on the UI. If the logs shown on the UI exceed the maximum row number, the display area will be emptied.
	Max .bin File Size	Set the maximum size of log file. If the log file size exceed the maximum size, a new log file will be generated.
	Enable Snoop File Split	If checked, setting the “Max Snoop File Size” is required. If the file reaches the maximum snoop file size, the tool creates a new .cfa file. If not checked, only a .cfa file will be created during decoding. Note: It is recommended to select only when the log volume is large, to avoid the .cfa file being too large.
	Max Snoop File Size	Set the maximum size of .cfa file. This item can only be set if 'Enable Snoop File' is checked.
	Output Directory	Specify the directory of the folder that contains the .bin and .log files.

4.2.3 Settings of Trace File Path

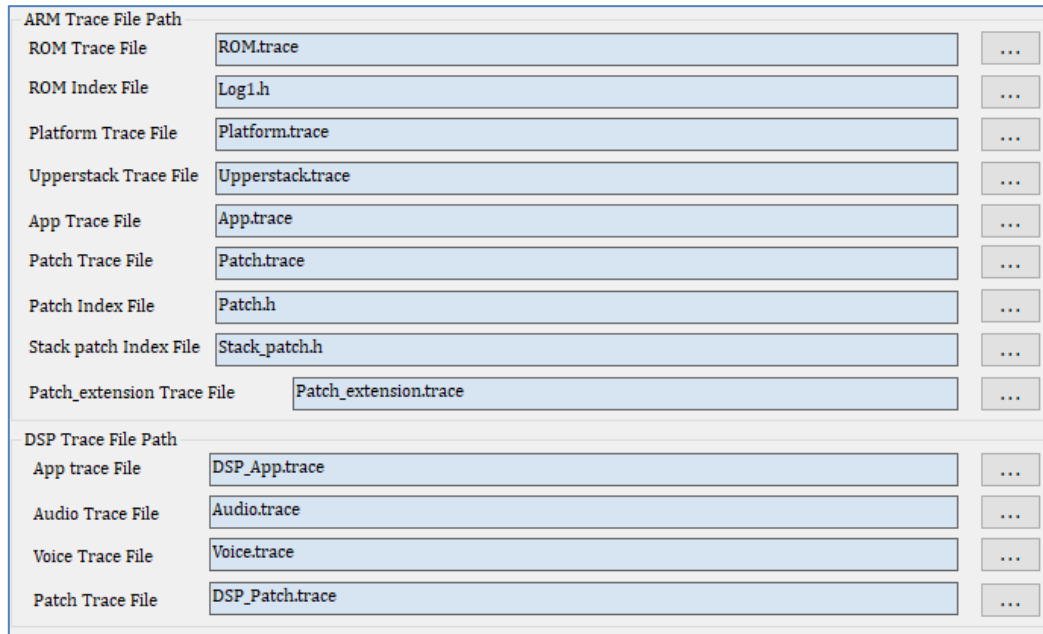


Figure 4-8 Trace File Path Settings

The following table describes the specific meaning of each option in the “Trace File Path” part.

Table 4-6 Trace Path Settings

Area	Setting	Description
Trace File Path	ARM Trace File Path	Set the .trace and .h file paths for decoding ARM log. Note: The .trace and .h files contain all the format strings, users should ensure to use the matching .trace and .h files.
	DSP Trace File Path	Set the .trace and .h file paths for decoding the DSP log. Note: The .trace and .h files contain all the format strings, users should ensure to use the matching .trace and .h files.

4.2.4 Swift Load .trace Files



Figure 4-9 Swift Load .trace

Table 4-7 Decoder Output Settings

Area	Setting	Description
Swift Load .trace	.trace File Path	By placing all the dependent .trace files in the same folder, the corresponding fields can automatically be filled with the .trace file path.

4.2.5 Settings of Load/Save and Confirm/Cancel

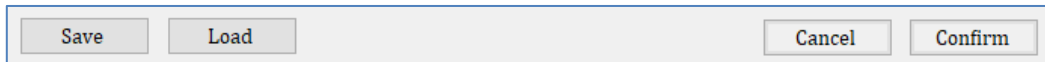


Figure 4-10 Load/Save and Confirm/Cancel Settings

The following table describes the specific meaning of each button in the “Load/Save and Confirm/Cancel Settings” part.

Table 4-8 Load/Save and Confirm/Cancel Settings

Area	Setting	Description
Load and Save Settings	Save	Save all configurations of the current page to the .ini file.
	Load	Load configuration items from the .ini files.
Confirm and Cancel Settings	Confirm	Apply all settings for the current page.
	Cancel	Cancel the settings of the current page and restore the previous settings.

5 Start and Stop Capturing and Decoding

1. Decode From Serial

- 1) Before clicking the "Start" button, please select the correct serial port, and set the correct serial port parameters and .trace file (refer to [4.2](#)). Then click the "Start" button to start receiving and parsing data.
- 2) To stop parsing, click the "Stop" button.

2. Decode From File

- 1) Before clicking the "Start" button, please set the correct .trace file. Then click the "Start" button, select the .bin file that needs to be reversed, and start parsing. Description: Multiple .bins can be selected at one time.
- 2) After parsing all .bin files, it will stop automatically.

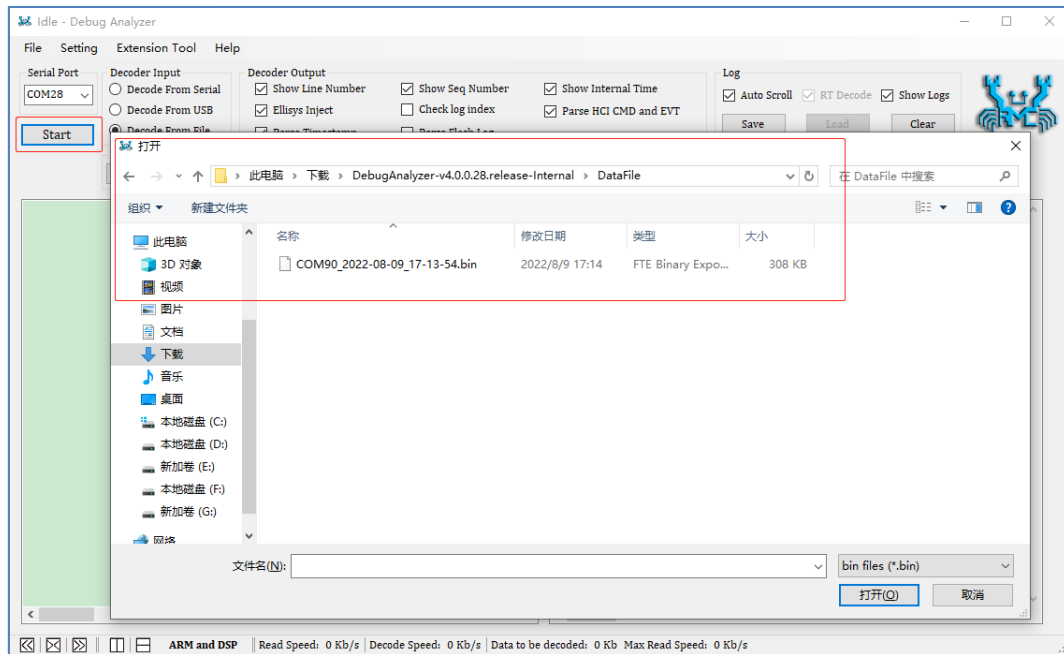


Figure 5-1 Select Bin File

6 Log Operation and Display on UI

6.1 Log Operation on UI

Users can copy, search and filter the logs shown on UI.

1. Copy logs

As shown in Figure 6-1, select the logs to be copied. Then press “Ctrl + C” to copy them.

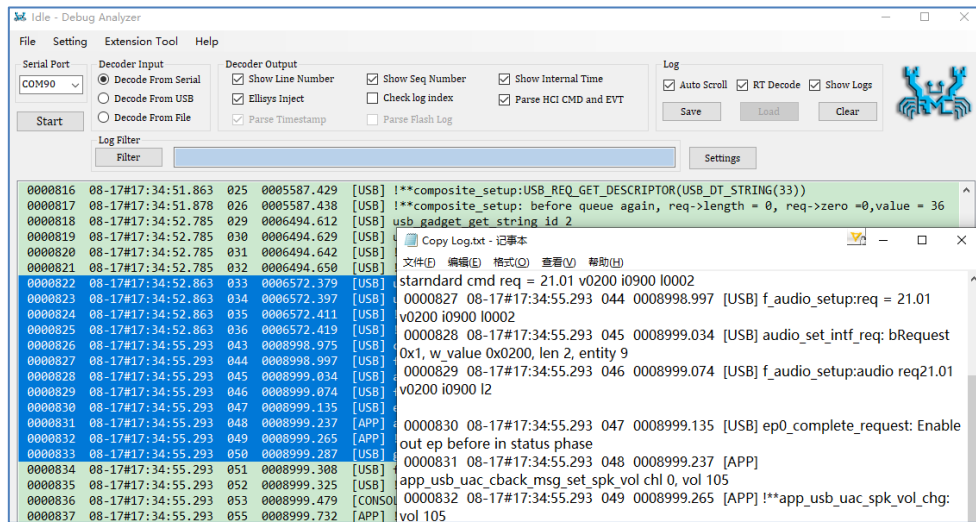


Figure 6-1 Copy Logs

2. Show details of one log

Double click on a log, a dialog box will pop up to show the details of the log, as shown in Figure 6-2.

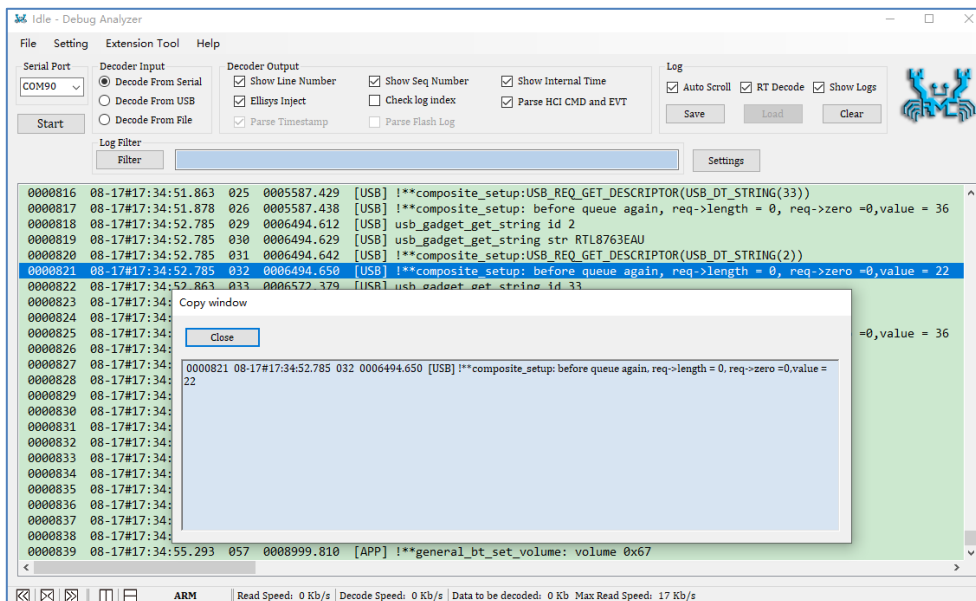


Figure 6-2 Show Log Details

3. Search and filter logs

- 1) Enter the keyword in the edit column of the filter.
- 2) Click the “Filter” button or press “Enter” key on the keyboard, the filter result will be shown as in Figure 6-3.
- 3) Double click on one log from the Log Filter dialog, which will locate the log in the main dialog.

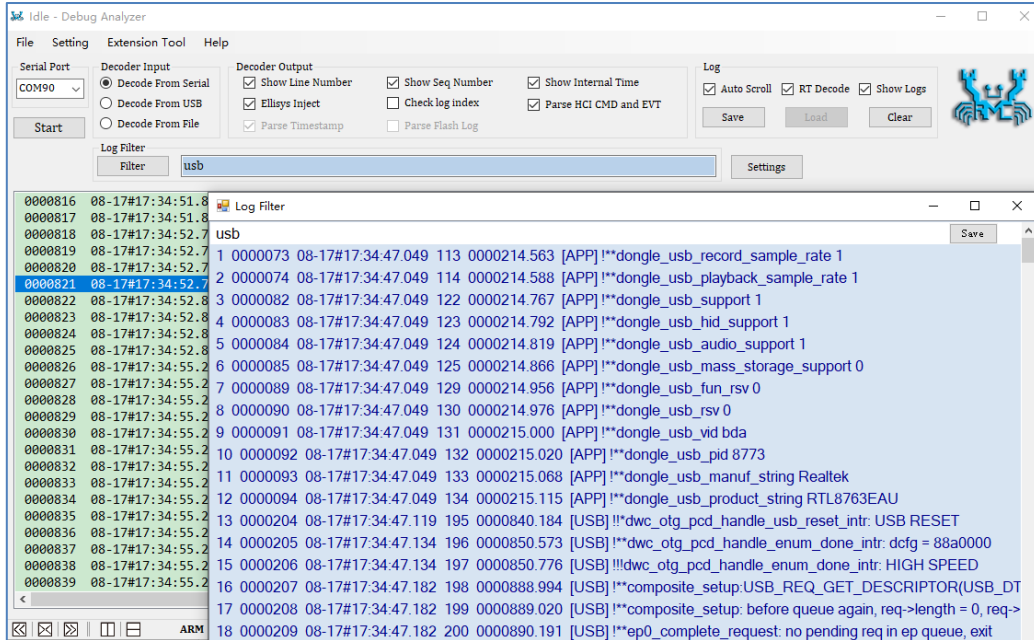


Figure 6-3 Search and Filter Logs

4. Click the button “save” to save logs

Click the "Save" button (Figure 6-4). Please note, this operation only saves all Logs displayed in the main window in the .log file. It excludes Logs that exceed the maximum number of displayed lines (which have been emptied).

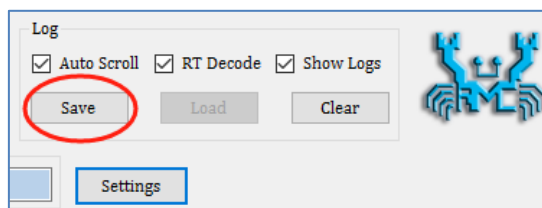


Figure 6-4 Save Button

6.2 Log Display on UI

1. ARM and DSP Logs can be displayed in left and right/upper and lower columns, as shown in Figure 6-5 and Figure 6-6.

0000816	08-17#17:34:51.863	025	0005587.429	[USB]	!!**composite	0000443	08-17#17:34:47.883	005	0001605.398	[DSP]	!!!Warning
0000817	08-17#17:34:51.878	026	0005587.438	[USB]	!!**composite	0000444	08-17#17:34:47.883	006	0001605.419	[DSP]	SdkSysInit
0000818	08-17#17:34:52.785	029	0006494.612	[USB]	usb_gadget_g	0000445	08-17#17:34:47.883	007	0001605.463	[DSP]	SdkRxInit
0000819	08-17#17:34:52.785	030	0006494.629	[USB]	usb_gadget_g	0000446	08-17#17:34:47.883	008	0001605.512	[DSP]	-----
0000820	08-17#17:34:52.785	031	0006494.642	[USB]	!!**composite	0000447	08-17#17:34:47.883	009	0001605.524	[DSP]	Recv Cmd
0000821	08-17#17:34:52.785	032	0006494.650	[USB]	!!**composite	0000448	08-17#17:34:47.883	010	0001605.538	[DSP]	Recv Cmd
0000822	08-17#17:34:52.863	033	0006572.379	[USB]	usb_gadget_g	0000449	08-17#17:34:47.883	011	0001605.550	[DSP]	g_SigProc
0000823	08-17#17:34:52.863	034	0006572.397	[USB]	usb_gadget_g	0000450	08-17#17:34:47.883	012	0001605.558	[DSP]	Rx Stereo
0000824	08-17#17:34:52.863	035	0006572.411	[USB]	!!**composite	0000451	08-17#17:34:47.883	013	0001605.566	[DSP]	Algo Enc
0000825	08-17#17:34:52.863	036	0006572.419	[USB]	!!**composite	0000452	08-17#17:34:47.883	014	0001605.626	[DSP]	Recv Cmd
0000826	08-17#17:34:55.293	043	0008998.975	[USB]	composite_se	0000453	08-17#17:34:47.883	015	0001605.691	[DSP]	Recv Cmd
0000827	08-17#17:34:55.293	044	0008998.997	[USB]	f_audio_setu	0000454	08-17#17:34:47.883	016	0001605.702	[DSP]	RecvCmdGa
0000828	08-17#17:34:55.293	045	0008999.034	[USB]	audio_set_in	0000455	08-17#17:34:47.883	017	0001605.739	[DSP]	SysRecvDa
0000829	08-17#17:34:55.293	046	0008999.074	[USB]	f_audio_setu	0000456	08-17#17:34:47.883	018	0001605.751	[DSP]	Recv Cmd
0000830	08-17#17:34:55.293	047	0008999.135	[USB]	ep0_comple	0000457	08-17#17:34:47.883	019	0001605.761	[DSP]	AUDIO_AC
0000831	08-17#17:34:55.293	048	0008999.237	[APP]	app_usb_uac	0000458	08-17#17:34:47.883	020	0001605.772	[DSP]	OnEvent 3
0000832	08-17#17:34:55.293	049	0008999.265	[APP]	!!**app_usb_u	0000459	08-17#17:34:47.883	021	0001605.842	[DSP]	epk:160
0000833	08-17#17:34:55.293	050	0008999.287	[USB]	generic_set	0000460	08-17#17:34:47.883	022	0001605.872	[DSP]	Rx Samp
0000834	08-17#17:34:55.293	051	0008999.308	[USB]	f_audio_comp	0000461	08-17#17:34:47.883	023	0001605.884	[DSP]	FifoInit
0000835	08-17#17:34:55.293	052	0008999.325	[USB]	!!**ep0_out_s	0000462	08-17#17:34:47.883	024	0001605.895	[DSP]	!AudioRoI
0000836	08-17#17:34:55.293	053	0008999.479	[CONSOLE]	!!ERROR!!	0000463	08-17#17:34:47.883	025	0001605.910	[DSP]	RmcNearEn
0000837	08-17#17:34:55.293	055	0008999.732	[APP]	!!**app_dongl	0000464	08-17#17:34:47.915	026	0001605.923	[DSP]	audio SDK
0000838	08-17#17:34:55.293	056	0008999.768	[APP]	!!**host_chck	0000465	08-17#17:34:47.915	027	0001605.939	[DSP]	CLK req
0000839	08-17#17:34:55.293	057	0008999.810	[APP]	!!**general_b	0000466	08-17#17:34:47.915	028	0001605.951	[DSP]	SdkRxInit

Figure 6-5 Show ARM and DSP Logs Left and Right

0000816	08-17#17:34:51.863	025	0005587.429	[USB]	!!**composite_setup:USB_REQ_GET_DESCRIPTOR(USB_DT_STRING(33))	0000443	08-17#17:34:47.883	005	0001605.398	[DSP]	!!!Warning!!! Get SDK Init COMMAND from MCU
0000817	08-17#17:34:51.878	026	0005587.438	[USB]	!!**composite_setup: before queue again, req->length = 0, req->zero =0,value = 36	0000444	08-17#17:34:47.883	006	0001605.419	[DSP]	SdkSysInit 1018fe30
0000818	08-17#17:34:52.785	029	0006494.612	[USB]	usb_gadget_get_string id 2	0000445	08-17#17:34:47.883	007	0001605.463	[DSP]	SdkRxInit_Free, 1
0000819	08-17#17:34:52.785	030	0006494.629	[USB]	usb_gadget_get_string str RTL8763EAU	0000446	08-17#17:34:47.883	008	0001605.512	[DSP]	----- OnSdkInit -----
0000820	08-17#17:34:52.785	031	0006494.642	[USB]	!!**composite_setup:USB_REQ_GET_DESCRIPTOR(USB_DT_STRING(2))	0000447	08-17#17:34:47.883	009	0001605.524	[DSP]	Recv Cmd Id=0xf, Len=0x8, Para=0x125fd
0000821	08-17#17:34:52.785	032	0006494.650	[USB]	!!**composite_setup: before queue again, req->length = 0, req->zero =0,value = 22	0000448	08-17#17:34:47.883	010	0001605.538	[DSP]	Recv Cmd Id=0x30, Len=0x8, Para=0x12072200
0000822	08-17#17:34:52.863	033	0006572.379	[USB]	usb_gadget_get_string id 33	0000449	08-17#17:34:47.883	011	0001605.550	[DSP]	g_SigProc.Txfs 0 g_SigProc.Rxfs 0
0000823	08-17#17:34:52.863	034	0006572.397	[USB]	usb_gadget_get_string str UCQ00011000101000	0000450	08-17#17:34:47.883	012	0001605.558	[DSP]	Rx Stereo 1 Passthrough 0
0000824	08-17#17:34:52.863	035	0006572.411	[USB]	!!**composite_setup:USB_REQ_GET_DESCRIPTOR(USB_DT_STRING(33))	0000451	08-17#17:34:47.883	013	0001605.566	[DSP]	Algo Enc:7, SamplesPerCh:512, BytePerSample=0, ChNum=1, SampleRateIdx=4
0000825	08-17#17:34:52.863	036	0006572.419	[USB]	!!**composite_setup: before queue again, req->length = 0, req->zero =0,value = 36	0000452	08-17#17:34:47.883	014	0001605.626	[DSP]	Recv Cmd Id=0xf3, Len=0x8, Para=0x1
0000826	08-17#17:34:55.293	043	0008998.975	[USB]	composite_setup: non-standard cmd req = 21.01 v0200 i0900 10002						

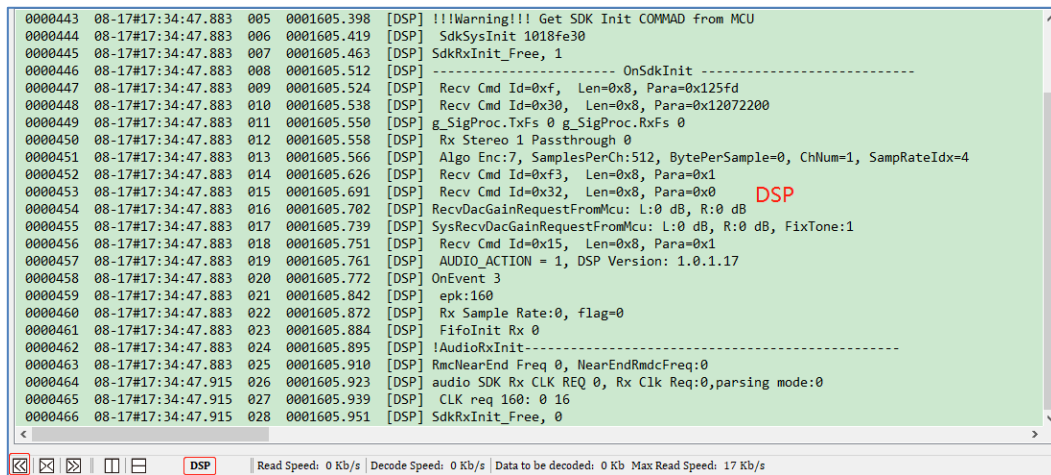
Figure 6-6 Show ARM and DSP Logs Up and Down

- If only focus on the ARM Logs, you can fold the DSP Logs display area, as shown in Figure 6-7.

0000816	08-17#17:34:51.863	025	0005587.429	[USB]	!!**composite_setup:USB_REQ_GET_DESCRIPTOR(USB_DT_STRING(33))						
0000817	08-17#17:34:51.878	026	0005587.438	[USB]	!!**composite_setup: before queue again, req->length = 0, req->zero =0,value = 36						
0000818	08-17#17:34:52.785	029	0006494.612	[USB]	usb_gadget_get_string id 2						
0000819	08-17#17:34:52.785	030	0006494.629	[USB]	usb_gadget_get_string str RTL8763EAU						
0000820	08-17#17:34:52.785	031	0006494.642	[USB]	!!**composite_setup:USB_REQ_GET_DESCRIPTOR(USB_DT_STRING(2))						
0000821	08-17#17:34:52.785	032	0006494.650	[USB]	!!**composite_setup: before queue again, req->length = 0, req->zero =0,value = 22						
0000822	08-17#17:34:52.863	033	0006572.379	[USB]	usb_gadget_get_string id 33						
0000823	08-17#17:34:52.863	034	0006572.397	[USB]	usb_gadget_get_string str UCQ00011000101000						
0000824	08-17#17:34:52.863	035	0006572.411	[USB]	!!**composite_setup:USB_REQ_GET_DESCRIPTOR(USB_DT_STRING(33))						
0000825	08-17#17:34:52.863	036	0006572.419	[USB]	!!**composite_setup: before queue again, req->length = 0, req->zero =0,value = 36						
0000826	08-17#17:34:55.293	043	0008998.975	[USB]	composite_setup: non-standard cmd req = 21.01 v0200 i0900 10002						
0000827	08-17#17:34:55.293	044	0008998.997	[USB]	f_audio_setu: req = 21.01 v0200 i0900 10002						
0000828	08-17#17:34:55.293	045	0008999.034	[USB]	audio_set_intf: req: bRequest 0x1, w_value 0x0200, len 2, entity 9						
0000829	08-17#17:34:55.293	046	0008999.074	[USB]	f_audio_setu: aud 21.01 v0200 i0900 12						
0000830	08-17#17:34:55.293	047	0008999.135	[USB]	ep0_complete_request: Enable out ep before in status phase						
0000831	08-17#17:34:55.293	048	0008999.237	[APP]	app_usb_uac_cbakg_msg_set_spk_vol ch1 0, vol 105						
0000832	08-17#17:34:55.293	049	0008999.265	[APP]	!!**app_usb_uac_spk_vol_chg: vol 105						
0000833	08-17#17:34:55.293	050	0008999.287	[USB]	generic_set_cmd_spk: cmd = 1, value = 69						
0000834	08-17#17:34:55.293	051	0008999.308	[USB]	f_audio_complete: case: (audio->set_con), END						
0000835	08-17#17:34:55.293	052	0008999.325	[USB]	!!**ep0_out_start: return for ep0 already Enable						
0000836	08-17#17:34:55.293	053	0008999.479	[CONSOLE]	!!ERROR!!! subType: Illegal subType input: 112						
0000837	08-17#17:34:55.293	055	0008999.732	[APP]	!!**app_dongle_handle_u2a_set_vol state 1 vol 69						
0000838	08-17#17:34:55.293	056	0008999.768	[APP]	!!**host_checking_handle_u2a_set_vol vol 69						
0000839	08-17#17:34:55.293	057	0008999.810	[APP]	!!**general_bt_set_volume: volume 0x67						

Figure 6-7 Show ARM Logs Only (DSP Logs Folded)

3. If only focus on DSP Logs, you can fold the ARM Log display area, as shown in Figure 6-8.



```

0000443 08-17#17:34:47.883 005 0001605.398 [DSP] !!!Warning!!! Get SDK Init COMPAD from MCU
0000444 08-17#17:34:47.883 006 0001605.419 [DSP] SdkSysInit 1018fe30
0000445 08-17#17:34:47.883 007 0001605.463 [DSP] SdkRxInit_Free, 1
0000446 08-17#17:34:47.883 008 0001605.512 [DSP] ----- OnSdkInit -----
0000447 08-17#17:34:47.883 009 0001605.524 [DSP] Recv Cmd Id=0xf, Len=0x8, Para=0x125fd
0000448 08-17#17:34:47.883 010 0001605.538 [DSP] Recv Cmd Id=0x30, Len=0x8, Para=0x12072200
0000449 08-17#17:34:47.883 011 0001605.550 [DSP] g_SigProc.TxFs 0 g_SigProc.RxFs 0
0000450 08-17#17:34:47.883 012 0001605.558 [DSP] Rx Stereo 1 Passthrough 0
0000451 08-17#17:34:47.883 013 0001605.566 [DSP] Algo Enc:7, SamplesPerCh:512, BytePerSample=0, ChNum=1, SampRateIdx=4
0000452 08-17#17:34:47.883 014 0001605.626 [DSP] Recv Cmd Id=0xf3, Len=0x8, Para=0x1
0000453 08-17#17:34:47.883 015 0001605.691 [DSP] Recv Cmd Id=0x32, Len=0x8, Para=0x0
0000454 08-17#17:34:47.883 016 0001605.702 [DSP] RecvDacGainRequestFromMcu: L:0 dB, R:0 dB
0000455 08-17#17:34:47.883 017 0001605.739 [DSP] SysRecvDacGainRequestFromMcu: L:0 dB, R:0 dB, FixTone:1
0000456 08-17#17:34:47.883 018 0001605.751 [DSP] Recv Cmd Id=0x15, Len=0x8, Para=0x1
0000457 08-17#17:34:47.883 019 0001605.761 [DSP] AUDIO_ACTION = 1, DSP Version: 1.0.1.17
0000458 08-17#17:34:47.883 020 0001605.772 [DSP] OnEvent 3
0000459 08-17#17:34:47.883 021 0001605.842 [DSP] epk:160
0000460 08-17#17:34:47.883 022 0001605.872 [DSP] Rx Sample Rate:0, flag=0
0000461 08-17#17:34:47.883 023 0001605.884 [DSP] FifoInit Rx 0
0000462 08-17#17:34:47.883 024 0001605.895 [DSP] !AudioRxInit-----
0000463 08-17#17:34:47.883 025 0001605.910 [DSP] RmcNearEnd Freq 0, NearEndRmdcFreq:0
0000464 08-17#17:34:47.915 026 0001605.923 [DSP] audio SDK Rx CLK REQ 0, Rx Clk Req:0,parsing mode:0
0000465 08-17#17:34:47.915 027 0001605.939 [DSP] CLK req 160: 0 16
0000466 08-17#17:34:47.915 028 0001605.951 [DSP] SdkRxInit_Free, 0
  
```

Figure 6-8 Show DSP Logs Only (ARM Logs folded)

7 How to use Ellisys view package of BTSnoop

7.1 Real-time display of BTSnoop Log

DebugAnalyzer supports sending BTSnoop Log to Ellisys to display HCI layer command/event interaction in real time. To use this function, users must check “Save Snoop File” and “Ellisys Injection”. The use steps are as follows.

1. Open Ellisys, as shown in Figure 7-1.

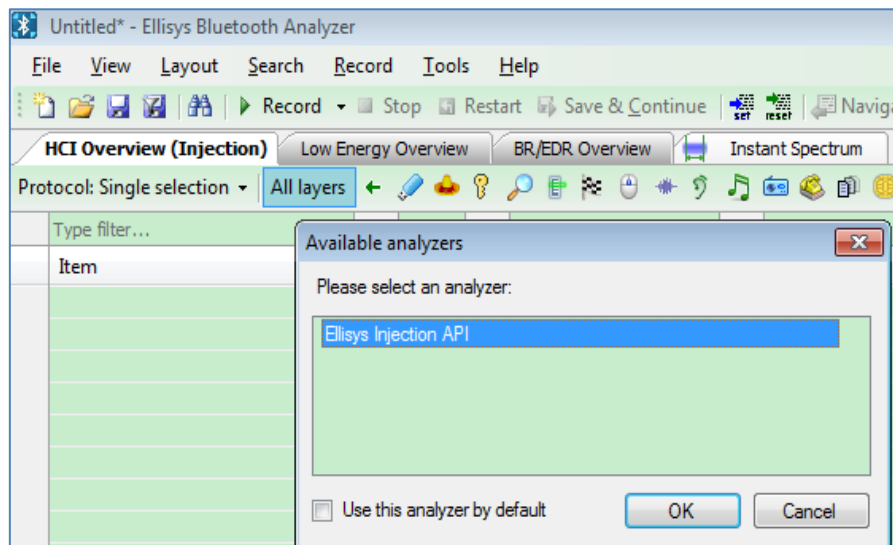


Figure 7-1 Open Ellisys

2. As shown in Figure 7-2, click “Configure ...” to confirm that the “Classic Bluetooth” and “Bluetooth Low Energy” options are selected for the first time.

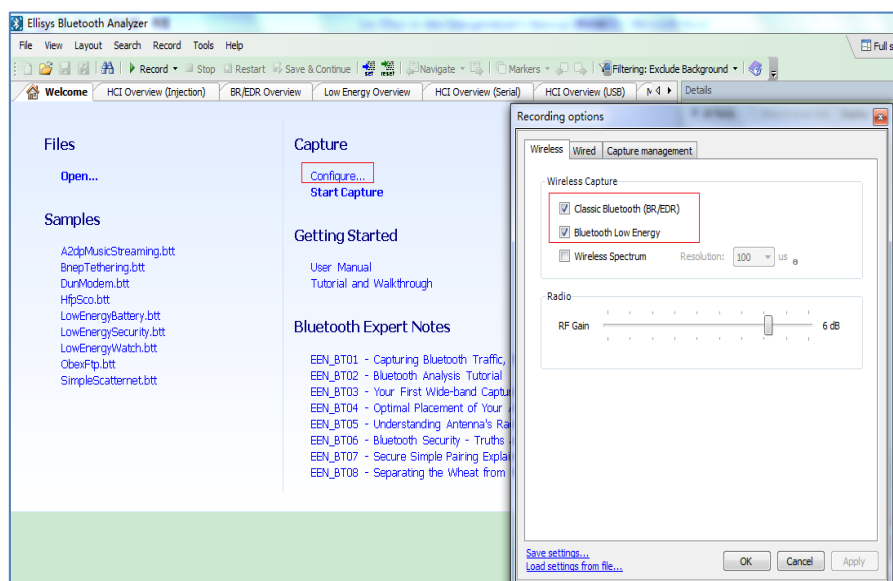


Figure 7-2 Ellisys Configure

3. Click “Start Capture” button to start capturing packets.

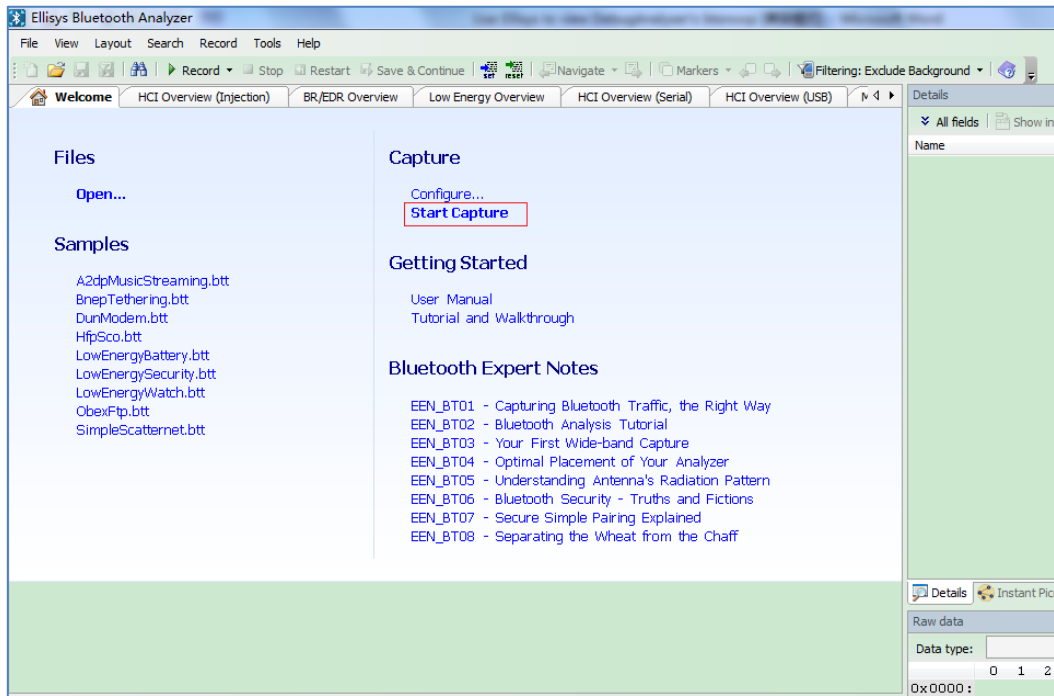


Figure 7-3 Start Capture

4. Open HCI injection

Check View→Overviews→HCI Overview (Injection), as shown in Figure 7-4.

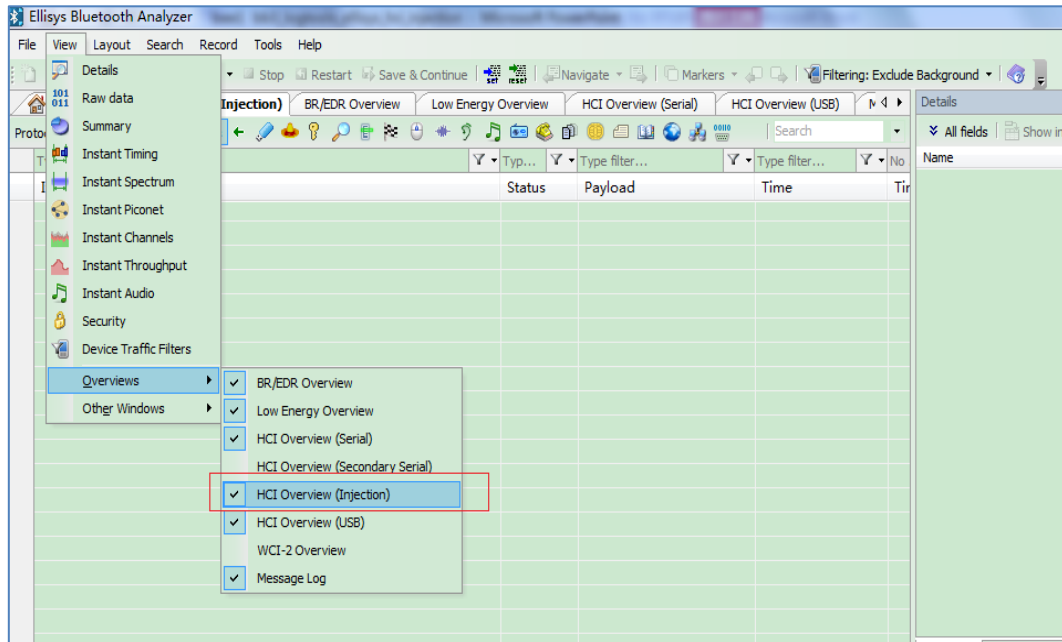
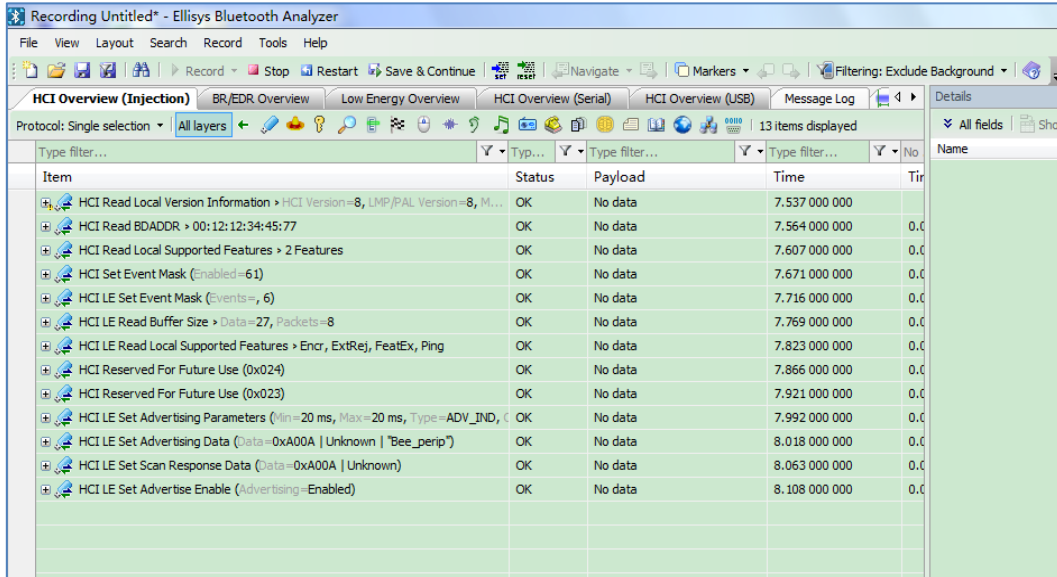


Figure 7-4 Open HCI Overview (Injection)

- The packages of BTSnoop caught by the DebugAnalyzer Tool are displayed in real time on the HCI Overview (Injection) dialog. Figure 7-5 shows the contents after SoC has been reset.



Item	Status	Payload	Time	Tir
HCI Read Local Version Information > HCI Version=8, LMP/PAL Version=8, M...	OK	No data	7.537 000 000	
HCI Read BDADDR > 00:12:12:34:45:77	OK	No data	7.564 000 000	0.0
HCI Read Local Supported Features > 2 Features	OK	No data	7.607 000 000	0.0
HCI Set Event Mask (Enabled=61)	OK	No data	7.671 000 000	0.0
HCI LE Set Event Mask (Events=, 6)	OK	No data	7.716 000 000	0.0
HCI LE Read Buffer Size > Data=27, Packets=8	OK	No data	7.769 000 000	0.0
HCI LE Read Local Supported Features > Encr, ExtRej, FeatEx, Ping	OK	No data	7.823 000 000	0.0
HCI Reserved For Future Use (0x024)	OK	No data	7.866 000 000	0.0
HCI Reserved For Future Use (0x023)	OK	No data	7.921 000 000	0.0
HCI LE Set Advertising Parameters (Min=20 ms, Max=20 ms, Type=ADV_IND, C	OK	No data	7.992 000 000	0.0
HCI LE Set Advertising Data (Data=0xA00A Unknown "Bee_perip")	OK	No data	8.018 000 000	0.0
HCI LE Set Scan Response Data (Data=0xA00A Unknown)	OK	No data	8.063 000 000	0.0
HCI LE Set Advertise Enable (Advertising=Enabled)	OK	No data	8.108 000 000	0.0

Figure 7-5 HCI Overview (Injection)

7.2 Notes for Using Ellisys to view BTSnoop

- Ellisys does not affect the decoding state of DebugAnalyzer. For example, DebugAnalyzer remains in the decoding state even if Ellisys is turned off or opened.
- The Ellisys air sniffer and HCI injection can be used simultaneously.
- Running multiple instances of the DebugAnalyzer tool on one PC is not recommended, as HCI injection data may get mixed together.
- The Ellisys function is only used to view HCI layer packages.

8 Extended Usage

8.1 Core Dump Parser

DebugAnalyzer supports parsing Core Dump.

To use "Core Dump Parser" page, select "Extension Tool – Core Dump Parser" from the menu bar.

Note:

1. Core Dump Parsing is only supported for IC Types RTL87x3E and RTL87x3D.
2. To use the Core Dump Parser, make sure that the ROM Trace File item on the [Trace File Path Settings](#) is correctly configured before recording logs via DebugAnalyzer.

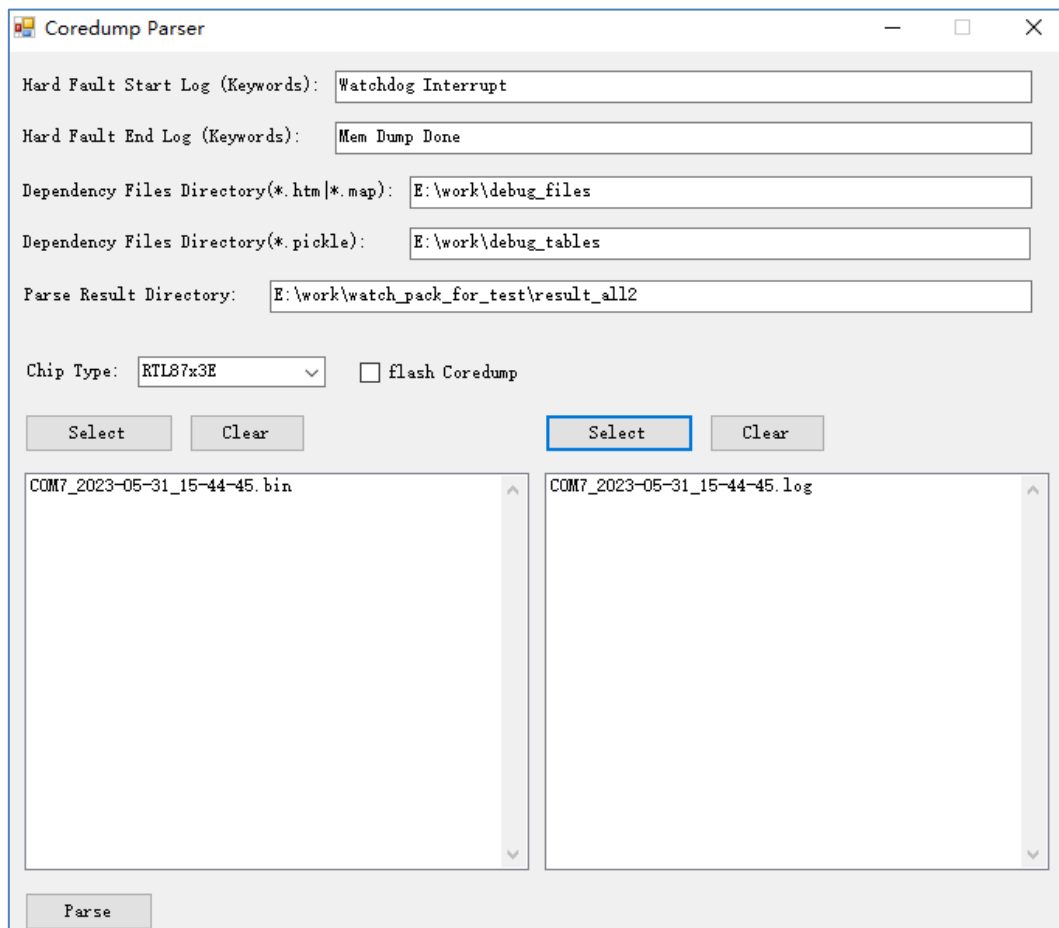


Figure 8-1 Core Dump Parse

The following table describes the specific meaning of each option in the “Core dump Parse” part.

Table 8-1 Core Dump Parse

Option	Selection	Description
Core Dump	Hard Fault Start Log (Keywords)	The keyword for the hard fault information start log in the .log file. Specific keywords can be found in Table 8-2: Hard Fault Keywords.
	Hard Fault End Log (Keywords)	The keyword for the hard fault information end log in the .log file. Specific keywords can be found in Table 8-2: Hard Fault Keywords.
	Dependency File Directory(*.htm *.map)	The path to the dependency files (.htm, .map).
	Dependency File Directory(*.pickle)	The path to the dependency files (.pickle).
	Parse Result Directory	The path to save the parsing result.
	Chip Type	RTL87x3E/ RTL87x3D chip type selection.
	Flash Core dump	Used to parse the core dump file read back from flash. After checking, you don't need to fill in Hard Fault Start Log, Hard Fault End Log Keywords, you don't need to select .log file, you need to select flash core dump bin file.
	Select .bin files	Select .bin files that contains the hard fault information.
	Select .log files	Select .log files that contains the hard fault information.
	Parse	Click the button to start parsing, the button will be grayed out after parsing starts, and will be restored after parsing is completed.

Table 8-2 Hard Fault Keywords

Option	Type	Start Log	End Log
Keyword	Hard Fault	Hard Fault Error	RTL87x3E: Mem Dump Done RTL87x3D: Memory Dump Done
	WDT	RTL87x3E: Watchdog Interrupt	RTL87x3E: Mem Dump Done
		RTL87x3D: WDT Interrupt	RTL87x3D: Memory Dump Done
	Reset	wdt reset	FSBL
	reboot	PC: 0x#####	PSR: 0x#####

Note:

1. If there are multiple exceptions in a log file, you can copy the start log and end log as a whole log line.
2. Reset: If there is no FSBL, take the first log after WDT reset.
3. Reboot: 0x##### is replaced by the specific PC and PSR values at the time of reboot, and this tool will help resolve the PC and LR corresponding to function name and PSR corresponding to interrupt type.

Dependency File Directory (*.htm|*.map) and Dependency File Directory(*.pickle) are used to set the path of the symbol information dependency file, which corresponds to the actual running ROM and flash code symbol information of the system. These two dependency file paths can be filled in only one or both, but the dependency file type cannot be duplicated.

➤ **Scenario 1: Only the Dependency File Directory (*.pickle) path needs to be set.**

When both the ROM and flash image dependency files are selected as *.pickle format, the path only needs to be set by Dependency File Directory (*.pickle).

➤ **Scenario 2: Only the Dependency File Directory (*.htm|*.map) path needs to be set.**

The debugging app image dependency file is *.htm/*.map format, and the path is set by Dependency File Directory (*.htm|*.map).

➤ **Scenario 3: Need to set both paths.**

For example, if you select *.pickle format for some dependency files (non-app image type) in ROM and flash image, and select *.htm/*.map format for debugging app image dependency files, you need to set both Dependency File Directory(*.pickle) and Dependency File Directory(*.htm|*.map) path information.

When searching for the dependency file, please pay attention to the symbol file corresponding to the actual running code of the system, only the one corresponding to the actual running of the same type of symbol file can be selected, and the same type of file of bank0 and bank1 cannot be selected at the same time.

The RTL87x3E symbol information file includes ROM symbols, common image symbols and bank flash code image symbol files, taking the pickle file as an example, and the following symbol files can be collected on bank0:











 sys_patch_bank0_variable_table.pickle	Patch
 sys_patch_bank0_function_table.pickle	
 stack_patch_4m_bank0_variable_table.pickle	
 stack_patch_4m_bank0_function_table.pickle	
 rtl87x3e_rom_variable_table.pickle	Rom
 rtl87x3e_rom_function_table.pickle	
 rtl87x3e_common_variable_table.pickle	Common
 rtl87x3e_common_function_table.pickle	
 app_variable_table.pickle	App
 app_function_table.pickle	

Figure 8-2 RTL87x3E Symbol Information

RTL87x3D symbol information file includes ROM symbol and bank flash code image symbol file, taking pickle file as an example, the following symbol files can be collected for bank0:









 app_bank0_variable_table.pickle	App
 app_bank0_function_table.pickle	
 stack_patch_bank0_function_table.pickle	Patch
 stack_patch_bank0_variable_table.pickle	
 Patch_bank0_function_table.pickle	
 Patch_bank0_variable_table.pickle	
 rtl87x3d_rom_function_table.pickle	Rom
 rtl87x3d_rom_variable_table.pickle	

Figure 8-3 TL87x3D Symbol Information

8.2 Rename title of DebugAnalyzer Tool

In many cases, such as capturing log of RWS Headsets, users would start two DebugAnalyzer Tools. Then rename the two to different titles (e.g. Left and Right) is distinguishable and helpful.

Instructions:

Click “Setting - Rename Title” (Figure 8-4), there are 5 options, and please select the new title.

1. Debug Analyzer
2. Left-Debug Analyzer
3. Right-Debug Analyzer
4. Primary-Debug Analyzer
5. Secondary-Debug Analyzer

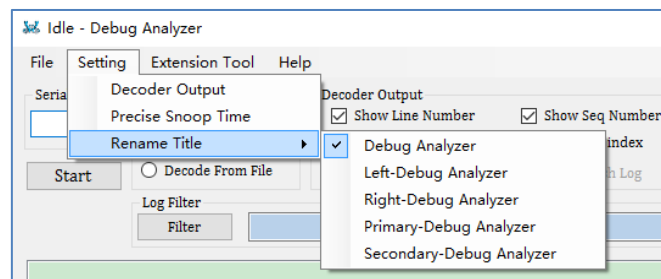


Figure 8-4 Rename Title

8.3 View version of DebugAnalyzer Tool

View method:

Click “Help” on Menu Bar and select “About Analyzer”.

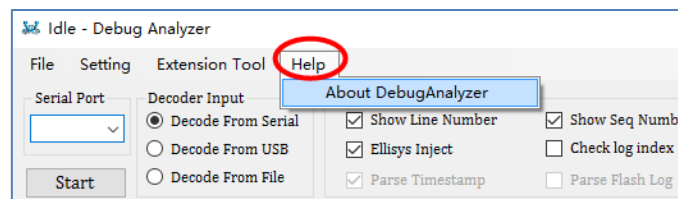


Figure 8-5 About Analyzer



Figure 8-6 View Version of Tool